Inventory routing with pickups and deliveries

Claudia Archetti, Marielle Christiansen,
M. Grazia Speranza

Published in

European Journal of Operational Research

Volume 268, Issue 1, 1 July 2018, Pages 314-324

https://doi.org/10.1016/j.ejor.2018.01.010

Inventory routing with pickups and deliveries

Claudia Archetti [(1)] Marielle Christiansen [(2)]

M. Grazia Speranza [(1)]

[(1)] *Department of Economics and Management*
University of Brescia, Italy
{claudia.archetti,grazia.speranza}@unibs.it
[(2)] *Department of Industrial Economics and Technology Management*
Norwegian University of Science and Technology, Trondheim, Norway
marielle.christiansen@iot.ntnu.no

March 14, 2018

Abstract

This paper introduces a class of problems which integrate pickup and delivery vehicle routing problems (PDPs) and inventory management, and we call them inventory routing problems with pickups and deliveries (IRP-PD). We consider a specific problem of this class, where a commodity is made available at several origins and demanded by several destinations. Time is discretized and transportation is performed by a single vehicle. A mathematical programming model is proposed together with several classes of valid inequalities. The models are solved with a branch-and-cut method. Computational tests are performed to show the effectiveness of the valid inequalities on instances generated from benchmark instances for the inventory routing problem. Results show that the branch-and-cut algorithm is able to solve to optimality 345 over 400 instances with up to 50 customers over 3 periods of time, and 142 over 240 instances with up to 30 customers and 6 periods. From a management perspective, results show that the average cost of a non integrated policy is more than 35% higher than the cost of an integrated policy.

Keywords: Pickup and delivery problems, Inventory routing, Valid inequalities

1 Introduction

The literature that integrates vehicle routing problems (VRPs) with other related decision problems has been constantly growing in the last decade. The study of these complex problems is made possible by the advances in optimization techniques and in technology, and is relevant because of the benefits coming from the simultaneous optimization of inter-related problems (see [15], [33], [42], [9] and [25] for quantitative studies of the value of integrated decisions). Papers have appeared which study, for example, the integration of vehicle routing problems with loading problems (see the survey by Iori and Martello [35] and, more recently, the one by Pollaris et al. [38]), with location problems (see the surveys by Prodhon and Prins [39] and by Drexl and Schneider [28]) and with production planning (see the survey by Adulyasak *et al.* [2]). These problems are often called *integrated vehicle routing problems* (see Bektaş *et al.* [12]).

One of the most studied classes of integrated VRPs is the class of problems that integrate vehicle routing and inventory management decisions. These problems are called Inventory Routing Problems (IRPs) (see Andersson *et al.* [5], Bertazzi and Speranza [14], and Coelho *et al.* [20]). A classical version of the IRP for a single vehicle is studied by Archetti *et al.* [6], who developed the first exact algorithm for the problem. A planning horizon is considered discretized in time periods, for example in days. A product must be shipped from a supplier to customers by means of capacitated vehicles. The demand of each customer is given in each time period as well as the availability at the supplier. Inventory capacity is known for the customers. The objective is the minimization of the total cost of the distribution, that includes routing cost and inventory cost at the supplier and at the customers. Several exact algorithms have been proposed for multi-vehicle IRPs (see Adulyasak *et al.* [1], Archetti *et al.* [7], Coelho and Laporte [22], [23], and Desaulniers *et al.* [27]). Matheuristics (Archetti *et al.* [8]), metaheuristics (Coelho *et al.* [19]) and decomposition-based heuristics (Cordeau *et al.* [24]) have also been developed. In Cordeau *et al.* [24] multiple products are considered as well. Recently, dynamic and stochastic versions of IRPs have been studied (see Coelho *et al.* [21] and Roldán *et al.* [40]).

In this paper we extend the class of problems that integrate vehicle routing and inventory management decisions to include problems where a product has to be picked up from multiple origins and delivered to multiple destinations, that is we consider integrated problems where pickup and delivery operations and inventory management are simultaneously optimized. Pickup and delivery problems (PDPs) have been studied for a long time (see

Berbeglia *et al.* [13], Parragh *et al.* [37] and Battarra *et al.* [11]) and can be classified according to the patterns of goods movement, the characteristics of the customers and restrictions on goods transported on vehicles. Several of the studied PDPs are motivated by a broad range of applications. Unfortunately, different names are sometimes used to indicate the same problem. We will refer here to the classification adopted in [11] where problems are classified in many-to-many (M-M) problems, where commodities have multiple origins and multiple destinations, one-to-one (1-1), where each commodity has a single origin and a single destination, and one-to-many-to-one (1-M-1) which is a mixture of the previous two classes.

Among the applications of PDPs, in this paper we are interested in distribution problems where a single commodity is made available at several origins and demanded at several destinations. These problems are usually modeled as M-M PDPs where quantities to be picked up at the origins (pickup locations) and quantities demanded at the destinations (delivery locations) are known. The focus is on the organization of the routes. We are interested in introducing the time dimension in the M-M PDP, considering a planning horizon discretized in time periods where a vehicle can perform at most one route in each time period. This is in accordance with the definition of the IRP by Archetti *et al.* [6]. Thus, the quantities made available at an origin may be stored at the origin, if the inventory capacity allows us to do that, or picked up. Similarly, the quantities demanded at a destination may be delivered in advance and stored. As a result, the problem integrates the pickup and delivery problem with the inventory management at the origins and destinations.

Several practical examples of an IRPs with pickups and deliveries can be thought of and have inspired our work. In road-based transportation, freight is often loaded onto Europe pallets for unit load or other load carriers. These pallets need to be picked up at locations where freight is delivered (for instance, food stores) and transported to other facilities where they are needed. Normally, there is a limited amount of space for such load carriers at both types of locations, and the transportation of pallets may be performed by a dedicated truck located at a warehouse (depot). A similar problem can be observed in a chain of stores where it often happens that some stores have an excess of inventory and others are running out of stock. A truck may be dedicated to pickup and deliver products to rebalance the inventory levels. We expect that IRPs with pickups and deliveries will become even more important in the future with new business models and systems for sharing and collaborative economy. In situations with several small producers and consumers where none of the producers or consumers are large enough to have their own transportation system, a separate transporter can have the responsibility of the transportation. With increased digitalization and information availability, the transporter should be able to use the inventory information from these producers/consumers. Another very different example is from maritime transportation. Arnesen *et al.* [10] studied an in-port ship routing and scheduling problem faced by chemical shipping companies. They modeled the problem as a Traveling Salesman Problem with Pickups and Deliveries, Time Windows and Draft Limits. In situations where a shipping company has the responsibility of both the transportation and the inventory management at the different locations in a port area and a particular ship is dedicated to perform the transportation, the problem has the structure of an inventory routing problem with pickups and deliveries. For simplicity, we use the terms supply at the pickup location (origins) and demand at the delivery location (destinations) independently of the relevant terms in practice.

Similar problems, integrating PDPs with inventory management, have been studied in the literature. An important class of such problems is related to closed-loop supply chain, which include the return processes besides forward flows to recover the value from the customers or end-users. This means that the locations are simultaneously pickup and delivery locatons. Closed-loop inventory routing problems for returnable transport items with simultaneous pickup and delivery are studied by Soysal [41] for transportation of soft drink bottles and Iassinovskaia *et al.* [34] for transporting all sorts of returnable items (boxes, trays, trolleys etc). These problems deviate from our work due to the characteristics of the commodities and locations. We do not allow simultaneous pickup and delivery at a customer, and our locations are classified as either a pickup or delivery location. In addition, we consider a single commodity. Another combined inventory management and pickup and delivery routing problem is studied by Van Anholt *et al.* [43], where the authors present the problem of replenishing automated teller machines (ATMs). Commodities can be brought to and from a depot, as well as being exchanged among customers to manage their inventory shortages and surpluses. This means that at a particular ATM, both pickup and delivery operations can be performed depending on the inventory level. There are several points in common with the problem studied in this paper. A major difference is that an ATM can act as both a pickup and delivery location.

PDP problems with inventory management have been extensively studied in the maritime context. We refer to Christiansen *et al.* [18] and Christiansen and Fagerholt [17] for an introduction and overview of maritime inventory routing problems (MIRPs) with pickup and delivery structure. These MIRPs deviate from the inventory routing problem studied in this paper in several ways. Normally, the fleet consists of several ships and is heterogeneous. Furthermore, there exists no depot, and the ships are sailing around the clock. The maritime routing problems are characterised by the long sailing times and time in port. All these aspects make the problem structure substantially different from the problem considered in this paper. The underlying mathematical models are either formulated with a continuous time variable (see e.g. Christiansen [16]) or the time is discretized in time periods (see e.g. Agra *et al.* [3]). For the time discrete models, a sailing leg or a stay in port may consist of several time periods. In addition, many of these works are strongly related to real applications; e.g. [31, 30, 4].

Finally, we would like to mention that there exist also contributions related to the study of IRPs where pickup operations are performed instead of the classical delivery operations. One example is given by Edirisinghe and James [29], where a problem arising in barge scheduling for oil pickup from off-shore oil-producing platforms with limited holding capacity is studied. The authors present a problem formulation which is then used to solve the problem and tested on the off-shore barge scheduling application.

The aim of this work is to introduce, and study, a basic version of the inventory routing problem with pickups and deliveries. In particular, we start by studying the problem where a single vehicle is available for the distribution. A vehicle starts and ends its tour at a depot which also plays the role of warehouse, in the sense that the commodity can be stored. Such a depot may coincide with one of the origins or destinations. We call the problem single-commodity, single-vehicle inventory routing problems with pickups and deliveries (1-1-IRP-PD).

The contribution of this work can be summarized as follows:

1. the 1-1-IRP-PD is introduced;
2. a mathematical programming formulation for the problem and several families of valid inequalities are presented;
3. a branch-and-cut algorithm is proposed;
4. benchmark instances on the basis of benchmark IRP instances introduced in [6] are generated;
5. the benefit of integrating the inventory management and routing problem with pickup and delivery structure compared to a non-integrated policy is highlighted.

Results show that the branch-and-cut algorithm is able to solve to optimality 345 over 400 instances with up to 50 customers over 3 periods of time, and 142 over 240 instances with up to 30 customers and 6 periods. A computational study shows that the average cost of a non integrated policy is more than 35% higher than the cost of an integrated policy. A similar study on the value of integration in lot sizing, inventory control and distribution is done in Darvish and Coelho [26].

The outline of the rest of the paper is as follows. In Section 2 we describe the problem and present the mathematical programming formulation. Several sets of valid inequalities are given in Section 3 together with a description of the branch-and-cut algorithm. Section 4 is devoted to the computational experiments, while conclusions are illustrated in Section 5.

2 Problem description and formulation

We consider the distribution problem over a planning horizon discretized in time periods. A single commodity is made available at several pickup locations and consumed at several delivery locations. One vehicle is available for the transportation. At each location, pickup or delivery, the commodity can be stored. An upper and a lower limit on the inventory stored at each location is known, where the lower limit may represent a safety stock against uncertainty and the upper limit storage capacity. In each time period, the quantity made available at each pickup location and the demand of each delivery location are known. These quantities may vary over time. The vehicle can perform at most one route in each time period, and the vehicle starts and ends its route at a depot where the commodity can be stored. A given amount of commodity is available at the depot at the beginning of the planning horizon. The depot acts as a warehouse where the commodity can be stored. The vehicle can visit any sequence of locations in a route. However, each location can be visited at most once in a route. At any time in the route the load of the vehicle cannot exceed the vehicle capacity. The costs include the inventory holding costs per period at pickup and delivery locations and the routing costs. The problem consists in deciding, for each time period, which locations to serve, how much to pickup or deliver at each visited location, and the route of the vehicle. If beneficial, the vehicle may stay at the depot in a time period without performing a route. As already mentioned, we call this problem the single-commodity, single-vehicle inventory routing problems with pickups and deliveries (1-1-IRP-PD). The objective is to minimize the sum of transportation cost and inventory cost at all locations. The sequence of operations performed at any pickup or delivery node in each time period is the following: first the commodity is delivered, in case of a delivery node, or picked up, in case of a pickup node, then the supply or demand is satisfied, and, finally, the inventory level is calculated. This sequence of operations is consistent with the assumption made in [7] and [20].

The 1-1-IRP-PD is defined on a graph $G = (N', A)$, where N' is the set of nodes (depot and pickup and delivery nodes) and A is the arc set. We denote by 0 the depot and by N^P and N^D the sets of pickup and delivery nodes, respectively. $N = N^D \cup N^P$ is the set of all n pickup and delivery nodes. Thus, $|N'| = n+1$. A cost c_{ij} is associated with each arc $(i,j) \in A$. The set of time periods within the planning horizon is denoted by $\mathcal{T} = \{1, ..., T\}$. The quantity made available at pickup node i or demanded at delivery node i, $i \in N$, in time period t, $t \in \mathcal{T}$, is indicated by d_{it}. The initial inventory level, the lower and upper limits on the inventory at node i, $i \in N$, are denoted as I_i^0, L_i and U_i, respectively. No upper and lower limit is set at the depot, while an initial inventory level I_0^0 is considered. No stock-out is allowed, i.e., the quantity distributed from the depot cannot exceed the amount of commodity in inventory. The unitary inventory holding cost at node i, $i \in N$, is denoted by h_i. The vehicle capacity is Q.

The mathematical programming formulation makes use of various sets of variables. Variables q_{it} identify the quantity picked up at node i, $i \in N^P$, or delivered to node i, $i \in N^D$, in time period t, $t \in \mathcal{T}$. The route of the vehicle in time period t is defined by means of the usual binary variables $\{x_{ijt}\}$, where x_{ijt} takes value 1 if arc (i,j) is traversed in time period t, and 0 otherwise. The visits to pickup and delivery nodes are identified by variables $\{y_{it}\}$, where y_{it} takes value 1 if node i, $i \in N$, is visited in time period t, and 0 otherwise. Two other sets of variables will be used. Variables $\{l_{ijt}\}$ are load variables and $\{I_{it}\}$ are inventory level variables. In particular, l_{ijt} gives the quantity on the vehicle when traversing arc (i,j), $i,j \in N'$, in time period t. Variable I_{it} gives the inventory level at node i, $i \in N'$, in time period t.

For easy reference, the notation is summarized in the following.

Indices

i, j	Nodes (depot and pickup and delivery nodes)
t	Time periods

Sets

A	Set of arcs
N^P	Set of pickup nodes
N^D	Set of delivery nodes
N	Set of pickup and delivery nodes, $\{1,....,n\}$
N'	Set of all nodes, $\{0,1,\ldots,n\}$
$\mathcal{T}$	Set of time periods, $\{1,....,T\}$
S	Subset of pickup and delivery nodes

Parameters

n	Number of pickup and delivery nodes
c_{ij}	Routing cost to travel directly from i to j, $i, j \in N'$
d_{it}	Supply at i, $i \in N^P$, or demand at i, $i \in N^D$, in time period t
h_i	Inventory holding cost at node i per time period
I_i^0	Initial inventory level at node i, $i \in N'$
L_i	Lower inventory limit at node i, $i \in N$
Q	Vehicle capacity
T	Number of time periods
U_i	Upper inventory limit at node i, $i \in N$

Decision variables

I_{it}	Inventory level at node i, $i \in N'$, in time period t
l_{ijt}	Quantity on the vehicle when traversing arc (i, j), $i, j \in N'$, in time period t
q_{it}	Quantity picked up at i, $i \in N^P$, or delivered to i, $i \in N^D$, in time period t
x_{ijt}	1 if arc (i, j), $i, j \in N'$, is traversed by the vehicle in time period t, and 0 otherwise
y_{it}	1 if node i, $i \in N$, is visited in time period t, and 0 otherwise

The 1-1-IRP-PD can be formulated as follows.

$$\min z = \sum_{(i,j)\in A} \sum_{t\in\mathcal{T}} c_{ij} x_{ijt} + \sum_{i\in N'} \sum_{t\in\mathcal{T}} h_i I_{it}, \tag{1}$$

$$\sum_{j\in N'} x_{ijt} - \sum_{j\in N'} x_{jit} = 0, \qquad i \in N', t \in \mathcal{T}, \tag{2}$$

$$\sum_{j\in N'} x_{ijt} - y_{it} = 0, \qquad i \in N', t \in \mathcal{T}, \tag{3}$$

$$\sum_{i\in S} \sum_{j\in S} x_{ijt} \leq \sum_{i\in S} y_{it} - y_{mt}, \qquad S \subseteq N, m \in S, t \in \mathcal{T}, \tag{4}$$

$$I_{it} - I_{i(t-1)} - d_{it} + q_{it} = 0, \qquad i \in N^P, t \in \mathcal{T} \tag{5}$$

$$I_{it} - I_{i(t-1)} + d_{it} - q_{it} = 0, \qquad i \in N^D, t \in \mathcal{T}, \tag{6}$$

$$I_{0t} - I_{0(t-1)} + \sum_{i \in N^D} q_{it} - \sum_{i \in N^P} q_{it} = 0, \qquad t \in \mathcal{T}, \tag{7}$$

$$I_{i0} = I_i^0, \qquad i \in N', \tag{8}$$

$$I_{i(t-1)} + q_{it} \leq U_i, \qquad i \in N^D, t \in \mathcal{T}, \tag{9}$$

$$I_{it} \geq L_i, \qquad i \in N^D, t \in \mathcal{T}, \tag{10}$$

$$I_{it} \leq U_i, \qquad i \in N^P, t \in \mathcal{T}, \tag{11}$$

$$I_{i(t-1)} - q_{it} \geq L_i, \qquad i \in N^P, t \in \mathcal{T}, \tag{12}$$

$$0 \leq q_{it} \leq min\{Q, U_i - L_i\} y_{it}, \qquad i \in N, t \in \mathcal{T}, \tag{13}$$

$$q_{it} \leq Q y_{0t}, \qquad i \in N, t \in \mathcal{T}, \tag{14}$$

$$\sum_{j \in N} l_{jit} + q_{it} - \sum_{j \in N} l_{ijt} = 0, \qquad i \in N^P, t \in \mathcal{T}, \tag{15}$$

$$\sum_{j \in N} l_{jit} - q_{it} - \sum_{j \in N} l_{ijt} = 0, \qquad i \in N^D, t \in \mathcal{T}, \tag{16}$$

$$0 \leq l_{ijt} \leq Q x_{ijt}, \qquad (i,j) \in A, t \in \mathcal{T}, \tag{17}$$

$$l_{0jt} \leq I_{0(t-1)} \qquad j \in N, t \in \mathcal{T}, \tag{18}$$

$$x_{ijt} \in \{0,1\}, \qquad (i,j) \in A, t \in \mathcal{T}, \tag{19}$$

$$y_{it} \in \{0,1\}, \qquad i \in N', t \in \mathcal{T}. \tag{20}$$

The objective function (1) minimizes the sum of routing and inventory holding costs. Constraints (2) are the flow conservation constraints, while constraints (3) link the $\{x_{ijt}\}$ and $\{y_{it}\}$ variables. The subtour elimination constraints are given in constraints (4), where set S is a subset of nodes. Furthermore, inventory balance for pickup and delivery nodes is ensured in constraints (5) and (6), respectively. Inventory levels at the depot are defined in (7). The initial conditions for the inventory levels are given by constraints (8), and the inventory limits are ensured in constraints (9)-(11). Constraints (9)-(11) are consistent with the assumption about the sequence of operations: first the node is served (the goods are picked up, in case of a pickup node, or delivered, in case of a delivery node), the demand (negative for a pickup node and positive for a delivery node) is satisfied and, finally, the inventory level is calculated. Constraints (12) establish that the quantity picked up at pickup nodes at time t should not exceed the inventory level at time $t-1$ adjusted for the lower limit L_i. The limits for the quantity picked up or delivered are provided by (13). Constraints (14) state that distribution can be performed in a given period only if a route is performed in the same period. The load balance constraints for pickup and delivery nodes are given by constraints (15) and (16), respectively. Constraints (17) ensure that the load on the vehicle does not exceed the capacity. Constraints (18) state that what is distributed from the depot at time t does not exceed the inventory level available at the end of the previous period. Finally, the requirements for the binary variables can be found in (19) and (20).

3 Solution method

The model is solved by a branch-and-cut algorithm. In Section 3.1 we present the valid inequalities developed, while Section 3.2 is devoted to a brief description of the branch-and-cut algorithm.

3.1 Valid inequalities

In the following we introduce sets of valid inequalities for the 1-1-IRP-PD problem. Inequalities

$$I_{i(t-1)} \geq d_{it}(1 - y_{it}) + L_i \qquad i \in N^D, \qquad t \in \mathcal{T} \tag{21}$$

imply that if delivery node i is not served at time t, i.e. $y_{it} = 0$, then the inventory level $I_{i(t-1)}$ at node i at time t is at least equal to the quantity d_{it} consumed by i at time t plus the minimum level L_i, while $I_{i(t-1)} \geq L_i$

otherwise. Inequalities

$$I_{i(t-k-1)} \geq \left(\sum_{j=0}^{k} d_{i(t-j)}\right)\left(1-\sum_{j=0}^{k} y_{i(t-j)}\right)+L_i \qquad i \in N^D, \quad t \in \mathcal{T}, \quad k=0,1,\ldots,t-1 \tag{22}$$

extend inequalities (21) to the case where, given k, delivery node i is not served at times $t-k, t-k+1, \ldots, t$. Therefore, if $\sum_{j=0}^{k} y_{i(t-j)} = 0$, then $I_{i(t-k-1)} \geq \sum_{j=0}^{k} d_{i(t-j)} + L_i$. Otherwise, $I_{i(t-k-1)} \geq L_i$. In

$$\sum_{j=1}^{t} y_{ij} \geq \left\lceil \frac{\sum_{j=1}^{t-1} d_{ij} - I_i^0 + L_i}{min\{Q, U_i - L_i\}} \right\rceil \qquad i \in N^D, \quad t \in \mathcal{T} \tag{23}$$

quantity $\sum_{j=1}^{t-1} d_{ij} - I_i^0 + L_i$ is the minimum quantity that has to be delivered to delivery node i up to time t. Since the maximum shipping quantity is $min\{Q, U_i - L_i\}$, then i has to be visited at least $\left\lceil \frac{\sum_{j=1}^{t-1} d_{ij} - I_i^0 + L_i}{min\{Q, U_i - L_i\}} \right\rceil$ times up to time t.

Inequalities (21)–(23) can be extended to pickup nodes. This is done in (24)–(26) in the following:

$$I_{i(t-1)} \leq U_i - d_{it}(1-y_{it}) \qquad i \in N^P, \qquad t \in \mathcal{T}, \tag{24}$$

$$I_{i(t-k-1)} \leq U_i - \left(\sum_{j=0}^{k} d_{i(t-j)}\right)\left(1-\sum_{j=0}^{k} y_{i(t-j)}\right) \qquad i \in N^P, \quad t \in \mathcal{T}, \quad k=0,1,\ldots,t-1, \tag{25}$$

$$\sum_{j=1}^{t} y_{ij} \geq \left\lceil \frac{\sum_{j=1}^{t-1} d_{ij} + I_i^0 - U_i}{min\{Q, U_i - L_i\}} \right\rceil \qquad i \in N^P, \quad t \in \mathcal{T}. \tag{26}$$

In addition, we have consistency inequalities (27) and (28). Inequalities

$$y_{it} \leq y_{0t} \qquad i \in N, \quad t \in \mathcal{T} \tag{27}$$

state that, if any node i is visited at time t, i.e. $y_{it} = 1$, then the depot has to be included in the route traveled at time t, i.e. $y_{0t} = 1$, whereas inequalities

$$x_{ijt} \leq y_{it} \qquad i \in N', \quad j \in N', \quad t \in \mathcal{T} \tag{28}$$

state that, if node j is the successor of node i in the route traveled at time t, i.e. $x_{ijt} = 1$, then i has to be visited at time t, i.e. $y_{it} = 1$.

The valid inequalitites (21)–(28) are based on the valid inequalities developed in [6] and [21], but are adjusted for the pickup and delivery structure of the problem.

The following four classes of valid inequalities have been proposed in Hernández-Pérez and Salazar-González [32] for the Pickup and Delivery Traveling Salesman Problem (PDTSP) and are valid for the 1-1-IRP-PD:

$$\sum_{i \in N^P \cup \{0\}} \sum_{j \in N^D} x_{ijt} \geq \frac{\sum_{j \in N^D} q_{jt}}{Q} \qquad t \in \mathcal{T}, \tag{29}$$

$$\sum_{i \in N^D} \sum_{j \in N^P \cup \{0\}} x_{ijt} \geq \frac{\sum_{i \in N^D} q_{it}}{Q} \qquad t \in \mathcal{T}, \tag{30}$$

$$\sum_{i \in N^D \cup \{0\}} \sum_{j \in N^P} x_{ijt} \geq \frac{\sum_{j \in N^P} q_{jt}}{Q} \qquad t \in \mathcal{T}, \tag{31}$$

$$\sum_{i \in N^P} \sum_{j \in N^D \cup \{0\}} x_{ijt} \geq \frac{\sum_{i \in N^P} q_{it}}{Q} \qquad t \in \mathcal{T}. \tag{32}$$

3.2 Branch-and-cut algorithm

We propose a branch-and-cut algorithm to solve the 1-1-IRP-PD. Formulation (1)–(20) is solved by relaxing constraints (4), which are exponentially many and are dynamically inserted only when violated. To separate subtour elimination constraints, we use the classical min-cut algorithm proposed in [36]. In particular, for each $t \in \mathcal{T}$, we build an auxiliary graph where the weights of the edges correspond to the values of variables x_{ijt}. Then, the min-cut algorithm is called which returns the value of the min-cut and the corresponding set S. If the corresponding inequality (4) is violated, it is added to the formulation.Valid inequalities (21)–(32) are all added to the formulation as they are polynomial in number. At each node of the branch-and-bound tree, we solve the relaxation of formulation (1)–(20), with the addition of all valid inequalities and the subtour elimination constraints separated in previous nodes. Once the relaxation is solved, the separation algorithm is called, the violated subtour elimination constraints are added to the formulation and the procedure is repeated. If no violated subtour elimination constraint is found, branching is performed. The branching strategy as well as the exploration strategy are performed using the default setting of the commercial solver we used in our tests, which is specified in Section 4. This setting is similar to the one used in [23] where a branch-and-cut algorithm is proposed which is currently a state-of-the-art exact solution algorithm for the IRP.

4 Computational results

In this section we describe the computational tests we performed. Section 4.1 is devoted to the description of the instances we generated, while computational results are reported in Sections 4.2 and 4.3. Section 4.2 is focused on the analysis of the performance of the formulation presented in Section 2 and of the valid inequalities. In particular, we analyze the impact of vehicle capacity on problem difficulty, effectiveness of valid inequalities and solution cost. Section 4.3 is instead devoted to the analysis of the benefits obtained when solving the 1-1-IRP-PD with respect to applying a decentralized policy. The aim of this analysis is to highlight the benefit of considering the integrated IRP and PDP instead of solving the subproblems related to the different actors of the system independently, as it happens in a decentralized policy.

The branch-and-cut algorithm presented in Section 3.2 is implemented in C++ in a Windows 10 operating system and compiled under Visual C++ 2012 Express Edition. The computational experiments are carried out on an Intel(R) Xeon(R) CPU E5- 1650 v2, 3.50 GHz machine with 64 GB of RAM. CPLEX 12.6 (64 bit version) is used to solve all MILPs on a single thread, for the ease of future comparison. CPLEX default parameters are used.

4.1 Instance generation

We generate instances by adapting the IRP instances described in [6]. In these instances, customer demand is constant over the entire planning horizon. Data related to customers are kept as such. A pickup node will have a negative value of demand, whereas a delivery node will have a positive value. In order to decide which customers are pickup nodes and which are delivery nodes we sum up the per period demand of customers with an odd index and the demand of customers with an even index. Then, the set of customers with the largest demand becomes the set of pickup locations. The instances have the following characteristics. The horizon T is equal to 3 and 6 and the number of customers is $n = 5t$, with $t = 1...6$ when $T = 6$ and $t = 1...10$ when $T = 3$. Two classes of instances are considered: with low and high inventory cost. For each instance characteristic (horizon, number of customers, inventory cost), 5 random instances were generated for a total of 160 instances.

4.2 Computational results: Formulation and valid inequalities

The experiments described in this section are aimed at testing the impact of the valid inequalities on the size of the instances solved to optimality and the optimality gaps for the unsolved instances. In addition, we will assess the impact of vehicle capacity on solution quality and problem difficulty. In fact, preliminary tests showed that, in most cases, only a small portion of the vehicle capacity inherited from the IRP instances was used in the optimal solution. Thus, we derived new instances by multiplying the original IRP vehicle capacity Q by a parameter α. We tested the following values of $\alpha : 1, 1/2, 1/4, 1/8$.

First, we performed a set of preliminary tests to evaluate the impact of the different classes of valid inequalities on the lower bound at the root node and on the final solution. In particular, we tested the impact of inequalities (21)–(32). In addition, we tested the impact of separating subtour elimination constraints on integer solutions only or on integer and fractional solutions. We ran these tests on instances with $n = 50$ for $T = 3$, $n = 30$ for $T = 6$ and all values of α, for a total of 80 instances, 20 for each value of α. We tested the following sets of inequalities:

- `Set I`: Inequalities (21)–(22) and (24)–(25).
- `Set II`: Inequalities (23) and (26).
- `Set III`: Inequalities (27).
- `Set IV`: Inequalities (28).
- `Set V`: Inequalities (29)–(32).
- `Set VI`: All valid inequalities (21)–(32).
- `Set VII`: None of the valid inequalities (21)–(32). Subtour elimination constraints are separated on integer and fractional solutions.
- `Set VIII`: All valid inequalities (21)–(32). Subtour elimination constraints are separated on integer and fractional solutions.

Note that, when testing sets `I`–`VI`, subtour elimination constraints are separated on integer solutions only while they are separated also on fractional solutions when testing `VII` and `VIII`.

The results are presented in Tables 1 and 2. Table 1 reports the average and maximum percentage improvement of the lower bound at the root node of the branch-and-bound tree for the different sets of valid inequalities and for each value of α with respect to the case where no valid inequality is considered. For each value of α, each row summarizes the results over the 20 instances with the corresponding value of α.

Table 1: Valid inequalities: improvement of the lower bound at the root node

	$\alpha = 1$		$\alpha = 1/2$		$\alpha = 1/4$		$\alpha = 1/8$	
	av. % gap	max % gap	av. % gap	max % gap	av. % gap	max % gap	av. % gap	max % gap
`Set I`	4.68	14.77	4.57	14.18	4.00	13.36	3.07	12.22
`Set II`	2.87	8.91	2.87	8.33	2.46	7.29	1.89	5.84
`Set III`	0.13	0.88	0.03	0.35	0.00	0.29	0.02	0.31
`Set IV`	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
`Set V`	0.05	0.37	0.04	0.35	0.02	0.31	0.00	0.31
`Set VI`	4.77	15.03	4.65	14.50	4.05	13.52	3.01	12.02
`Set VII`	0.11	0.72	0.12	0.89	0.05	0.70	−0.01	0.20
`Set VIII`	4.77	15.03	4.65	14.50	4.05	13.52	3.01	12.02

We can notice that the most effective sets of inequalities are the ones of sets `I` and `II`. It also seems that the entire improvement obtained by adding all inequalities (related to the results of set `VIII`) is attributable to set `I`.

Table 2 reports the results obtained at the end of the computation. The table reports the following values: the number of instances where feasible and optimal solutions are found, the average and maximum improvement of the lower bound at the end of the computation with respect to the case where no valid inequality is considered, and the average and maximum optimality gap. Results are reported for each set of valid inequalities and for each value of α.

Table 2: Valid inequalities: evaluation of performance at the end of computation

	# feasible solutions	# optimal solutions	av. % gap LB	max % gap LB	av % opt. gap	max % opt. gap
	$\alpha = 1$					
`Set I`	20	13	0.17	2.65	0.95	9.33
`Set II`	20	12	0.02	0.69	0.91	7.12
`Set III`	17	11	−0.04	0.63	0.40	4.99
`Set IV`	18	12	0.00	0.01	0.88	9.47
`Set V`	18	13	0.06	1.18	0.73	6.49
`Set VI`	19	13	0.19	2.25	0.72	7.35
`Set VII`	19	18	0.49	5.26	0.33	6.15
`Set VIII`	19	18	0.47	5.27	0.12	2.15

	$\alpha = 1/2$					
`Set I`	20	14	0.13	2.69	0.92	10.03
`Set II`	19	14	0.00	1.16	0.96	7.83
`Set III`	20	12	-0.03	0.80	1.18	14.13
`Set IV`	20	14	0.00	0.01	0.97	8.50
`Set V`	20	13	0.04	1.29	1.25	10.93
`Set VI`	19	14	0.07	1.24	0.66	5.48
`Set VII`	20	16	0.35	5.63	1.08	11.68
`Set VIII`	20	18	0.43	5.63	0.78	10.16
	$\alpha = 1/4$					
`Set I`	16	8	0.01	0.91	0.80	4.74
`Set II`	17	7	-0.04	0.35	1.12	6.14
`Set III`	17	6	0.01	0.39	1.43	10.49
`Set IV`	18	7	0.00	0.01	1.13	5.71
`Set V`	16	7	0.00	1.42	0.88	4.53
`Set VI`	19	8	0.00	0.91	1.88	16.56
`Set VII`	16	4	-0.10	5.18	1.90	10.57
`Set VIII`	18	4	0.00	5.18	3.50	23.55
	$\alpha = 1/8$					
`Set I`	10	2	0.02	0.47	3.06	7.73
`Set II`	11	2	0.00	0.50	4.20	15.79
`Set III`	9	2	-0.02	0.38	3.94	13.83
`Set IV`	10	2	0.00	0.00	3.58	12.06
`Set V`	10	2	0.00	0.47	3.57	14.36
`Set VI`	12	2	0.04	0.80	4.21	12.80
`Set VII`	9	0	-0.03	2.74	5.38	12.76
`Set VIII`	11	0	0.00	1.97	18.94	53.29

The results show that the advantage of separating subtour elimination constraints on integer and fractional solutions (in sets `VII` and `VIII`) is relevant for large values of α, as a larger number optimal solutions is found in this case in sets `VII` and `VIII` with respect to the other sets. However, the advantage decreases when the capacity of the vehicles decreases. Concerning the other sets of valid inequalities, there are slight differences among them and in general set `I` performs better than the others. Concerning the inequalities of sets `III`, `IV` and `V`, both tables show that their contribution to the performance of the branch-and-cut algorithm is negligible. However, we decided to keep them in the following tests as they are in polynomial number and, thus, they do not make the formulation heavier.

On the basis of the results of the preliminary tests shown above, in the following tests we decided to compare three different formulations:

- *Basic*: Formulation (1)–(20) is implemented with no valid inequalities reported in Section 3. Subtour elimination constraints are separated on integer solutions only.

- *All-subtour*: Formulation (1)–(20) is implemented together with all valid inequalities reported in Section 3. As in the basic formulation, subtour elimination constraints are separated only on integer solutions.

- *All*: Formulation (1)–(20) is implemented together with all valid inequalities reported in Section 3. Subtour elimination constraints are separated on integer and fractional solutions.

Tables 3–5 report a summary of the results of the three formulations classified by capacity (α), horizon (T) and number of customers (n). We report, in columns 2–8, the following values: the number of instances in each class, the number of feasible solutions found, the number of optimal solutions, the optimality gap, the gap between the lower bound at the end of computation and the best upper bound found by all formulations, the average CPU time in seconds and the average number of nodes in the branch-and-bound tree. Note that, even if no formulation is able to find a feasible solution for an instance, this does not mean there does not exist a feasible solution for that instance. It simply means that no formulation is able to find one.

Table 3: Performance of the *Basic* formulation

	# instances	# feasible	# optimal	av % opt. gap	av % gap w.r.t best UB	av. CPU time	av. # of B&B nodes
$\alpha = 1$	160	156	116	0.76	0.69	1187.46	34791.19
$\alpha = 1/2$	160	156	136	0.24	0.34	688.82	8901.73
$\alpha = 1/4$	160	154	139	0.27	0.30	659.18	9777.30
$\alpha = 1/8$	160	139	60	2.42	2.76	2350.73	74625.87
$T = 3$	400	399	340	0.53	0.49	707.82	22275.10
$T = 6$	240	206	111	1.57	1.91	2077.77	48272.23
$n = 5$	80	80	80	0.00	0.00	0.94	777.30
$n = 10$	80	80	68	0.40	−0.04	632.15	57096.84
$n = 15$	80	80	65	0.60	0.58	745.76	29845.60
$n = 20$	80	75	48	1.27	1.51	1805.71	51716.59
$n = 25$	80	69	40	1.75	2.39	1905.71	39537.86
$n = 30$	80	62	31	1.49	2.15	2243.13	29826.00
$n = 35$	40	40	32	0.50	0.46	832.48	24982.86
$n = 40$	40	40	34	0.45	0.44	753.68	16499.80
$n = 45$	40	39	29	1.22	1.11	1373.25	24300.46
$n = 50$	40	40	24	1.49	1.26	1918.58	29000.95
All	640	605	451	0.88	1.00	1221.55	32024.02

Table 4: Performance of the *All-subtour* formulation

	# instances	# feasible	# optimal	av % opt. gap	av % gap w.r.t best UB	av. CPU time	av. # of B&B nodes
$\alpha = 1$	160	153	138	0.20	20.41	641.87	9227.39
$\alpha = 1/2$	160	155	136	0.25	20.77	657.91	9588.83
$\alpha = 1/4$	160	155	117	0.68	20.21	1139.33	32432.86

$\alpha = 1/8$	160	138	63	2.35	18.99	2327.19	71517.29
$T = 3$	400	397	345	0.43	18.15	655.34	20376.39
$T = 6$	240	204	109	1.61	23.66	2085.30	47883.60
$n = 5$	80	80	80	0.00	4.92	1.09	643.96
$n = 10$	80	80	68	0.41	14.40	634.80	53474.90
$n = 15$	80	80	65	0.58	19.58	763.64	31720.94
$n = 20$	80	76	46	1.20	21.93	1772.81	47519.78
$n = 25$	80	69	41	2.02	24.90	1896.44	39554.93
$n = 30$	80	59	32	1.41	30.26	2209.85	28000.38
$n = 35$	40	40	34	0.42	22.79	760.43	24451.93
$n = 40$	40	40	34	0.47	22.45	756.10	16870.73
$n = 45$	40	38	28	0.82	24.00	1365.50	24178.88
$n = 50$	40	39	26	1.03	24.76	1625.93	23734.20
All	640	601	454	0.83	20.12	1191.58	30691.59

Table 5: Performance of the *All* formulation

	# instances	# feasible	# optimal	av % opt. gap	av % gap w.r.t best UB	av. CPU time	av. # of B&B nodes
$\alpha = 1$	160	158	152	0.08	20.11	281.99	321.11
$\alpha = 1/2$	160	159	152	0.08	20.49	326.24	667.21
$\alpha = 1/4$	160	158	111	0.73	20.10	1263.09	7702.24
$\alpha = 1/8$	160	137	58	4.45	19.11	2427.26	14224.24
$T = 3$	400	392	331	1.12	18.33	744.35	2838.79
$T = 6$	240	220	142	1.42	22.93	1625.16	10545.21
$n = 5$	80	80	80	0.00	4.92	1.93	524.59
$n = 10$	80	80	68	0.39	14.37	599.28	20949.03
$n = 15$	80	80	64	0.50	19.50	753.61	7392.35
$n = 20$	80	77	57	1.13	21.55	1304.45	7002.66
$n = 25$	80	76	52	1.65	23.81	1384.86	4349.63
$n = 30$	80	67	40	1.95	30.12	2023.51	2749.47
$n = 35$	40	40	31	0.96	23.17	952.60	2125.90
$n = 40$	40	39	32	1.14	22.60	1005.90	1439.70
$n = 45$	40	38	27	4.94	24.15	1341.13	1137.55
$n = 50$	40	35	22	1.81	24.99	1759.53	1020.58
All	640	612	473	1.22	19.97	1074.65	5728.70

All variants are able to find a feasible solution for almost all instances when $T = 3$ (for 399, 397 and 392 over 400 instances, respectively) while, when $T = 6$, the variant which finds the largest number of feasible solutions is the *All* variant, with 220 feasible solutions found over 240 instances, while the other two versions finds a solution for 206 and 204 instances, respectively. Also in terms of optimal solutions found, the results indicate that the *All* variant achieves the largest number of optimal solutions, 331 over 400 with $T = 3$ and 142 over 240 with $T = 6$, for a total number of 473 instances solved to optimality against 451 and 454 of the other two variants. When the average optimality gap is considered, we observe that all variants obtain small average optimality gaps. The relatively larger gap of the *All* variant is due to the instances with small vehicle capacity. We comment on this further in the following.

We now focus the analysis on the impact of the vehicle capacity on the performance of all variants. Table 6 reports a summary of the performance of the three formulations on the basis of the value of α. Each row reports the summary of the results of the 160 instances with the specified value of α.

Table 6: Performance of the three formulations

	# feasible	# optimal	av % opt. gap	av % gap w.r.t best UB	av. CPU time	av. # of B&B nodes
			$\alpha = 1$			
Basic	154	139	0.27	0.30	659.18	9777.30
All-subtour	153	138	0.20	20.41	641.87	9227.39
All	158	152	0.08	20.11	281.99	321.11
			$\alpha = 1/2$			
Basic	156	136	0.24	0.34	688.82	8901.73
All-subtour	155	136	0.25	20.77	657.91	9588.83
All	159	152	0.08	20.49	326.24	667.21
			$\alpha = 1/4$			
Basic	156	116	0.76	0.69	1187.46	34791.19
All-subtour	155	117	0.68	20.21	1139.33	32432.86
All	158	111	0.73	20.10	1263.09	7702.24
			$\alpha = 1/8$			
Basic	139	60	2.42	2.76	2350.73	74625.87
All-subtour	138	63	2.35	18.99	2327.19	71517.29
All	137	58	4.45	19.11	2427.28	14224.24

From Table 6 it is clear that the problem becomes more difficult to solve when the value of α decreases and that the advantage of enhancing the formulation through subtour elimination constraints is lost. This confirms what we noticed from the results of Table 2 and can be explained by the fact that, when α decreases, the vehicle routes become shorter and, thus, the separation procedure for the subtour elimination constraint often leads to no violation found, and therefore turns out to be a waste of time. On the other side, the valid inequalities introduced in Section 3 are more effective when the vehicle capacity decreases, as shown by the number of optimal solutions found.

Finally, in the following tables, we provide a comparison of solution costs with respect to the value of α. In particular, we compare the cost components of the solutions obtained when $\alpha = 1/2, 1/4, 1/8$ with the solutions obtained when $\alpha = 1$. We consider only those instances for which we have the optimal solution for all values of α. Table 7 refers to the instances with low inventory cost while Table 8 to the instances with high inventory cost. For each table, we separate instances with $T = 3$ and $T = 6$ and, for each value of T, results are summarized over the different values of n. We first report the number of instances solved to optimality over all values of α, in the second column, and then the average number of routes in the solution with $\alpha = 1$. The following three groups of 5 columns refer to the different values of α lower than 1. The first column reports the average number of routes while the following four provide the average gap in total cost (TC), transportation cost (TrC), inventory cost at the pickup and delivery locations (IC) and inventory cost at the depot (ID) with respect to the solution with $\alpha = 1$, respectively.

Table 7: Comparison of costs: High inventory cost instances

	# opt.	$\alpha = 1$ av. # routes	av. # routes	$\alpha = 1/2$ % increase w.r.t $\alpha = 1$ TC	TrC	IC	ID	av. # routes	$\alpha = 1/4$ % increase w.r.t $\alpha = 1$ TC	TrC	IC	ID	av. # routes	$\alpha = 1/8$ % increase w.r.t $\alpha = 1$ TC	TrC	IC	ID
$T = 3$																	
$n = 5$	5	1.13	1.4	8.03	15.72	2.52	−1.27	2.4	37.65	71.41	−1.17	−0.31	2.4	38.89	72.88	−0.75	−0.41
$n = 10$	5	1	1	3.39	8.58	4.51	−1.76	1.4	16.01	41.02	5.32	−1.29	2.2	40.49	106.27	13.16	−6.26
$n = 15$	5	1	1	0.13	0	2.6	−1.06	1.6	3.17	10.26	6.74	−2.88	2.6	19.27	63.8	6.11	−2.83
$n = 20$	4	1	1	0.08	0.14	4.42	−1.93	1	3.12	9.49	12.18	−4.73	2.25	13.56	46.11	4.85	−2.07
$n = 25$	2	1	1	0	0	0	0	1	0.98	4.24	4.07	−1.82	1.5	5.49	24.22	4.31	−1.34
$n = 30$	1	1	1	0	0	0	0	1	0.73	0	9.22	−4.05	2	4.08	10.37	6.81	−2.79
$n = 35$	2	1	1	0	0	0	0	1	0.09	0	2.53	−1.04	1	1.84	5.42	8.86	−3.37
$n = 40$	2	1	1	0	0	0	0	1	0.08	0	4.36	−1.84	1	1.21	3.16	13.93	−5.02
$T = 6$																	
$n = 5$	5	3	3	3.96	8.31	4.81	−1.99	4.6	24.02	58.42	6.98	−2.97	5.6	32.93	77.2	14.02	−5.24
$n = 10$	1	3	3	0.01	0	1.43	−0.5	3	3.28	9.24	13.03	−4.41	6	14.8	48.59	18.76	−6.96
All	32	1.41	1.44	1.56	3.27	2.03	−0.85	1.8	8.91	20.41	6.33	−2.53	2.66	17.26	45.8	9.01	−3.63

Table 8: Comparison of costs: Low inventory cost instances

	# opt.	$\alpha = 1$ av. # routes	av. # routes	$\alpha = 1/2$ % increase w.r.t $\alpha = 1$ TC	TrC	IC	ID	av. # routes	$\alpha = 1/4$ % increase w.r.t $\alpha = 1$ TC	TrC	IC	ID	av. # routes	$\alpha = 1/8$ % increase w.r.t $\alpha = 1$ TC	TrC	IC	ID
$T = 3$																	
$n = 5$	5	1	1.4	14.97	15.39	7.34	−3.08	2.4	71.05	71.41	−3.12	0.32	2.4	73.18	72.88	−2.84	0.35
$n = 10$	5	1	1	8.00	8.58	1.67	−0.61	1.4	38.18	41.02	3.80	−0.97	2.2	97.67	106.27	10.64	−5.64
$n = 15$	5	1	1	0.23	0.00	4.90	−2.20	1.4	7.89	10.15	4.68	−1.60	2.4	50.34	63.50	6.25	−2.14
$n = 20$	4	1	1	0.05	0.14	7.57	−3.51	1	7.82	9.49	14.71	−5.89	2.25	37.44	46.07	8.39	−3.60
$n = 25$	1	1	1	0.00	0.00	0.00	0.00	1	1.42	2.09	−6.47	2.90	1	11.55	13.63	−0.80	1.11
$n = 30$	1	1	1	0.00	0.00	−3.02	1.63	1	0.14	0.00	9.14	−3.36	1	7.00	10.14	12.77	−4.53
$n = 35$	2	1	1	0.00	0.00	−2.27	1.07	1	0.01	0.00	0.16	−0.07	1	5.22	5.42	4.20	−1.53
$n = 40$	2	1	1	0.00	0.00	0.00	0.00	1	0.02	0.00	0.97	−0.35	1	2.47	3.10	10.47	−3.58
$T = 6$																	
$n = 5$	5	3	3	8.29	8.31	2.19	−1.10	4.6	50.95	58.42	3.69	−1.85	5.6	69.57	77.20	9.97	−4.15
$n = 10$	1	3	3	0.00	0.00	−0.38	0.18	3	7.74	9.31	9.11	−3.33	5	35.78	47.52	16.12	−5.10
All	32	1.40	1.44	3.15	3.24	1.80	−0.76	1.78	18.52	20.19	3.67	−1.42	2.39	39.02	44.57	7.52	−2.88

The tables show that the value of α has a big impact on total cost, especially in the case of low inventory cost. This is due to the fact that the major cost component which is influenced by the reduction in vehicle capacity is the transportation cost. When the inventory cost is low, the transportation cost covers a wider portion of the total cost and, thus, the increase in total cost is more remarkable with respect to the case when the inventory cost is high. We also notice that the increase in total cost is reduced when the capacity is halved while it starts to be remarkable when it becomes a quarter of the original capacity. Finally, the impact on total cost is smoother when n and T increase. Similar observations can be made when considering the number of routes.

4.3 Computational results: Comparison of management strategies

In this section, we perform a computational study with the aim of showing the value of the integrated policy in the 1-1-IRP-PD. In particular, we compare the solution obtained by solving the 1-1-IRP-PD with the solution obtained through a decentralized policy where:

- The customers, which correspond to the set of the delivery nodes, decide when they want to be served and how much they want to receive. In particular, they apply the classical (s, S) policy, i.e., when the inventory level reaches the minimum level, i.e., s, they order a quantity equal to $S - s$.

- The supplier determines the distribution plan on the basis of the quantities required by the customers. Goods are picked up directly from the depot (supplier's warehouse) or from pickup locations.

The idea is that supplier and pickup locations belong to the same supply chain and thus are controlled by a central decision maker. However, the decision maker has no control over the customers, corresponding to the delivery locations, who independently determine their optimal delivery schedule.

In order to have a fair comparison between the integrated and the decentralized policy, we impose that the final inventory level at the delivery customers is zero. Thus, we reduce the quantity delivered in the last visit in order to achieve a final zero inventory level. In fact, given that in the decentralized policy each customer applies the (s, S) policy, it may happen that there is a positive final inventory level, while this does not happen in the 1-1-IRP-PD where no constraint is considered on the quantity delivered to customers. Moreover, we also modify the instances used in the tests presented in the previous section in order to assure that the inventory cost at the delivery customers is higher than the inventory cost at the supplier and at the pickup locations. This in order to avoid that the integrated policy takes advantage from transferring goods from pickup customers (or supplier) to delivery customers with a lower inventory costs. The instances have been modified as follows. For the instances with high inventory cost, the inventory cost at the supplier is set at 0.1, the inventory cost at the pickup customers is randomly generated in $[0.1, 0.3]$ and the inventory cost at the delivery customers is randomly generated in $[0.3, 0.5]$. For the instances with low inventory cost, the inventory cost at the supplier is set at 0.01, the inventory cost at the pickup customers is randomly generated in $[0.01, 0.03]$ and the inventory cost at the delivery customers is randomly generated in $[0.03, 0.05]$. We tested instances with $\alpha = 1$. Results are summarized in Tables 9-10. The *All* formulation has been used to solve both policies. Table 9 summarizes the behavior of the *All* formulation on the new instances for both policies. We report, for both policies, the number of feasible solutions found, the number of optimal solutions, the average CPU time in seconds and the average number of nodes. Results are classified by horizon and by number of customers.

Table 9: Performance of all variants when solving the integrated and decentralized policies on the new instances

	# instances	# feasible	# optimal	av. CPU time	av. # of B&B nodes
			Integrated policy		
$T=3$	100	100	100	26.94	28.71
$T=6$	60	58	57	452.27	860.72
$n=5$	20	20	20	0.00	0.50
$n=10$	20	20	20	0.85	20.65
$n=15$	20	20	20	6.30	65.15
$n=20$	20	20	20	116.75	788.95
$n=25$	20	20	20	294.95	680.90
$n=30$	20	18	17	944.30	1049.25
$n=35$	10	10	10	9.60	10.40
$n=40$	10	10	10	51.40	92.50
$n=45$	10	10	10	104.50	83.10
$n=50$	10	10	10	91.20	54.60
All	160	158	157	186.44	340.71
			Decentralized policy		
$T=3$	100	100	95	363.94	680.87
$T=6$	60	60	60	138.37	343.47
$n=5$	20	20	20	0.00	0.20
$n=10$	20	20	20	0.15	8.15
$n=15$	20	20	20	1.25	21.10
$n=20$	20	20	20	8.40	74.95
$n=25$	20	20	20	29.50	161.90

$n = 30$	20	20	20	394.15	981.30
$n = 35$	10	10	10	68.40	345.90
$n = 40$	10	10	9	634.70	1936.80
$n = 45$	10	10	9	1026.70	1973.70
$n = 50$	10	10	7	1872.90	2117.90
All	160	160	155	279.35	554.34

We notice that the new instances seem to be easier to solve than the ones used in the previous section. In fact, looking at the first part of the table reporting data on the integrated policy, we note that 157 out of 160 instances are solved to optimality and the average computing time is 186.44 seconds. Looking at Table 5, we see that, on the previous instances with $\alpha = 1$, the *All* variant is able to solve 152 instances to optimality and the average computing time is 281.99 seconds.

Table 10 reports the comparison of solution costs. We report the gap of the solution obtained with the decentralized policy with respect to the solution obtained with the integrated policy in terms of: total cost (TC), transportation cost (TrC), inventory cost at the pickup and delivery locations (IC), inventory cost at the depot (ID) and number of routes. For all cost components we report the minimum, average and maximum gaps. For the number of routes we report the average and the maximum gaps as the minimum is always 0. We consider only instances that were solved to optimality by both policies. Results are classified by horizon, number of customers and inventory cost.

Table 10: Performance of the integrated and decentralized policies

	% gap TC			% gap TrC			% gap IC			% gap ID			% gap # routes	
	min	av.	max	min	av.	max	min	av.	max	min	av.	max	av.	max
$T = 3$	1.12	40.47	63.19	0.00	44.84	65.84	−30.17	−12.29	0.00	49.09	72.12	92.91	97.89	100.00
$T = 6$	2.32	27.66	40.28	0.00	18.17	25.58	−11.66	0.19	10.12	87.14	110.99	129.40	31.58	33.33
$n = 5$	1.12	26.98	56.10	0.00	23.35	56.49	−30.17	−5.10	10.12	49.09	85.99	119.52	53.33	100.00
$n = 10$	15.31	33.08	51.63	13.34	30.83	53.43	−21.30	−5.73	6.01	67.48	90.29	120.32	66.67	100.00
$n = 15$	20.89	38.34	63.19	17.67	36.02	65.84	−19.76	−6.75	5.55	66.13	94.73	125.81	65.00	100.00
$n = 20$	17.80	36.75	52.25	15.38	34.70	54.74	−18.80	−5.76	4.60	63.31	92.34	128.69	66.67	100.00
$n = 25$	17.38	34.65	50.18	14.23	31.70	51.98	−18.94	−6.86	5.50	70.67	91.45	117.13	66.67	100.00
$n = 30$	18.80	35.51	50.50	11.99	31.86	53.14	−14.89	−7.86	6.67	66.84	92.28	129.40	72.55	100.00
$n = 35$	33.81	38.59	48.54	35.67	43.06	51.89	−16.75	−12.48	−6.27	66.69	71.53	76.64	100.00	100.00
$n = 40$	31.89	39.64	46.74	31.62	45.39	58.38	−16.64	−11.20	−6.51	64.80	71.92	75.24	100.00	100.00
$n = 45$	35.61	43.80	56.50	42.99	52.43	60.60	−16.77	−12.72	−9.47	64.34	70.73	75.30	100.00	100.00
$n = 50$	34.00	37.83	43.38	39.30	42.40	44.70	−15.50	−12.04	−9.60	71.32	73.46	75.79	100.00	100.00
high inv. cost	9.35	36.36	52.97	0.00	34.89	65.84	−29.78	−7.75	10.12	54.33	86.13	127.48	73.08	100.00
low inv. Cost	1.12	34.67	63.19	0.00	34.59	65.66	−30.17	−7.54	9.53	49.09	87.30	129.40	72.97	100.00
All	1.12	35.54	63.19	0.00	34.75	65.84	−30.17	−7.65	10.12	49.09	86.70	129.40	73.03	100.00

The results show that the decentralized policy is indeed expensive. In fact, there is an average increase in total cost of 40.47% when $T = 3$ and of 27.66% when $T = 6$. The increase in total cost is not influenced by the inventory cost. In fact, the average increase is very similar for high and low inventory cost. There is a minor impact of the number of customers. The gap tends to increase when the number of customers increases, even though the relation is not monotonic. Finally, the main components of the total cost that generate the increase are transportation cost and inventory cost at the depot.

5 Concluding remarks

We introduced and studied the 1-1-IRP-PD which is an inventory routing problem with pickup and delivery locations. We presented a mathematical formulation of the problem together with different classes of valid inequalities. The problem is solved through a branch-and-cut algorithm. We studied the effectiveness of valid inequalities by performing tests on different problem formulations. The results show that subtour elimination constraints are effective when the vehicle capacity is big and the planning horizon is long. On the other side, the effectiveness of ad-hoc valid inequalities improves when the vehicle capacity decreases. Finally, we performed a computational study to show the benefit of using an integrated strategy like the one which is at the basis of the 1-1-IRP-PD, where all decisions are centralized, versus a decentralized policy where customers optimize their own decisions. The results show that integration is highly beneficial. We believe this is an important message that shows that classical sequential approaches should be updated with more sophisticated, and beneficial, integrated approaches. In fact, complex problems used to be handled by decomposing them in subproblems, as no other viable way was available. However, modern technologies and methodologies enable to face more complex problems, and this study is an example that this is worthwhile.

Acknowledgements

The authors are grateful to two anonymous reviewers whose comments and suggestions allowed them to improve a previous version of this paper.

References

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

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

[3] A. Agra, H. Andersson, M. Christiansen, and L. Wolsey. A maritime inventory routing problem: Discrete time formulations and valid inequalities. *Networks*, 62(4):297–314, 2013.

[4] A. Agra, M. M. Christiansen, and A. Delgado. Mixed integer formulations for a short sea fuel oil distribution problem. *Transportation Science*, 47(1):108–124, 2013.

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

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

[7] C. Archetti, N. Bianchessi, S. Irnich, and M.G. Speranza. Formulations for an inventory routing problem. *International Transactions in Operational Research*, 21(3):353–374, 2014.

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

[9] C. Archetti and M.G. Speranza. The inventory routing problem: the value of integration. *International Transactions in Operational Research*, 23(3):393–407, 2016.

[10] M.J. Arnesen, M. Gjestvang, X. Wang, K. Fagerholt, K. Thun, and J.G. Rakke. A traveling salesman problem with pickups and deliveries, time windows and draft limits: Case study from chemical shipping. *Computers & Operations Research*, 77:20–31, 2017.

[11] M. Battarra, J.-F. Cordeau, and M. Iori. Pickup-and-delivery problems for goods transportation. *Vehicle routing: problems, methods, and applications. MOS/SIAM series on optimization, 2e*, pages 161–192, 2014.

[12] T. Bektaş, G. Laporte, and D. Vigo. Integrated vehicle routing problems. *Computers & Operations Research*, 55:126–126, 2015.

[13] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15(1):1–31, 2007.

[14] L. Bertazzi and M.G. Speranza. Inventory routing problems with multiple customers. *EURO Journal on Transportation and Logistics*, 2(3):255–275, 2013.

[15] P. Chandra and M.L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503–517, 1994.

[16] M. Christiansen. Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33(1):3–16, 1999.

[17] M. Christiansen and K. Fagerholt. Ship routing and scheduling in industrial and tramp shipping. *Vehicle routing: problems, methods, and applications. MOS/SIAM series on optimization, 2e*, pages 381–408, 2014.

[18] M. Christiansen, K. Fagerholt, B. Nygreen, and D. Ronen. Ship routing and scheduling in the new millennium. *European Journal of Operational Research*, 228(3):467–483, 2013.

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

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

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

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

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

[24] J.-F. Cordeau, D. Laganà, R. Musmanno, and F. Vocaturo. A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, 55:153–166, 2015.

[25] J.-F. Côté, G. Guastaroba, and M. G. Speranza. The value of integrating loading and routing. *European Journal of Operational Research*, 257(1):89–105, 2017.

[26] M. Darvish and L.C. Coelho. Sequential versus integrated optimization: Lot sizing, inventory control and distribution. Technical Report CIRRELT-2016-60, CIRRELT, University of Montreal, Canada, 2016.

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

[28] M. Drexl and M. Schneider. A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research*, 241(2):283–308, 2015.

[29] N.C.P. Edirisinghe and R.J.W. James. Fleet routing position-based model for inventory pickup under production shutdown. *European Journal of Operational Research*, 236:736–747, 2014.

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

[31] R. Grønhaug, M. Christiansen, G. Desaulniers, and J. Desrosiers. A branch-and-price method for a liquefied natural gas inventory routing problem. *Transportation Science*, 44(3):400–415, 2010.

[32] H. Hernández-Pérez and J.-J. Salazar-González. A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. *Discrete Applied Mathematics*, 145(1):126 – 139, 2004.

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

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

[35] M. Iori and S. Martello. Routing problems with loading constraints. *TOP*, 18(1):4–27, 2010.

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

[37] S.N. Parragh, K.F. Doerner, and R.F. Hartl. A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, 58(1):21–51, 2008.

[38] H. Pollaris, K. Braekers, A. Caris, G.K. Janssens, and S. Limbourg. Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, 37(2):297–330, 2015.

[39] C. Prodhon and C. Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17, 2014.

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

[41] M. Soysal. Closed-loop inventory routing problem of returnable transport items. *Transportation Research Part D*, 48:31–45, 2016.

[42] M. Stålhane, H. Andersson, M. Christiansen, and K. Fagerholt. Vendor managed inventory in tramp shipping. *OMEGA*, 47:60–72, 2014.

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