Computers & Operations Research 38 (2011) 1731–1746

Contents lists available at ScienceDirect

Computers & Operations Research

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

Analysis of the maximum level policy in a production-distribution system

Claudia Archetti [a], Luca Bertazzi [a], Giuseppe Paletta [b], M. Grazia Speranza [a,*]

[a] Department of Quantitative Methods, University of Brescia, Italy
[b] Department of Economics and Statistics, University of Calabria, Italy

ARTICLE INFO

Available online 9 March 2011

Keywords:
Logistics
Production
Transportation
Inventory
Computational complexity
Exact and hybrid heuristic algorithms

ABSTRACT

We consider a production-distribution system, where a facility produces one commodity which is distributed to a set of retailers by a fleet of vehicles. Each retailer defines a maximum level of the inventory. The production policy, the retailers replenishment policies and the transportation policy have to be determined so as to minimize the total system cost. The overall cost is composed by fixed and variable production costs at the facility, inventory costs at both facility and retailers and routing costs. We study two different types of replenishment policies. The well-known order-up to level (OU) policy, where the quantity shipped to each retailer is such that the level of its inventory reaches the maximum level, and the maximum level (ML) policy, where the quantity shipped to each retailer is such that the inventory is not greater than the maximum level. We first show that when the transportation is outsourced, the problem with OU policy is NP-hard, whereas there exists a class of instances where the problem with ML policy can be solved in polynomial time. We also show the worst-case performance of the OU policy with respect to the more flexible ML policy. Then, we focus on the ML policy and the design of a hybrid heuristic. We also present an exact algorithm for the solution of the problem with one vehicle. Results of computational experiments carried out on small size instances show that the heuristic can produce high quality solutions in a very short amount of time. Results obtained on a large set of randomly generated problem instances are also shown, aimed at comparing the two policies.

© 2011 Elsevier Ltd. All rights reserved.

0. Introduction

The integration of the supply chain is a key factor for the success of medium and large companies. Different models for the integration of inventory and routing have been proposed. The objective is to determine distribution policies that minimize the total system cost, while avoiding stock-outs and respecting storage capacity limitations. We refer to Bertazzi et al. [1], Campbell et al. [2], Cordeau et al. [3] and Federgruen and Simchi-Levi [4] for an in-depth overview of this area of research. A limited number of models have been proposed for the integration of the production, inventory and distribution functions. The objective is to determine the production policy, the retailers replenishment policies and the transportation policy that minimize the total system cost. We refer to Cohen and Lee [5] for a strategic stochastic model of a complete supply chain, in which different functional areas are considered. A hierarchical decomposition approach is proposed to solve the problem. Thomas and Griffin [6] review the coordination issue of the different functional areas at an operational level when deterministic models are used. Blumenfeld et al. [7] analyze the trade-offs between transportation, inventory and production set-up costs over an infinite time horizon. Different shipping policies (direct shipping, shipping through a consolidation terminal and a combination of them) are studied on the basis of several simplifying assumptions. Sarmiento and Nagi [8] and Erengüç et al. [9] review the integration between production and transportation. Chandra and Fisher [10] propose a computational study to evaluate the effect of the coordination between production and distribution planning over a finite time horizon.

We study an integrated system in which one commodity is produced at a facility and shipped to several retailers over a finite time horizon. Shipments from the facility to the retailers are performed by a fleet of vehicles. Each vehicle has a given transportation capacity. The commodity is consumed by the retailers in a deterministic and time-varying way. Each retailer determines a minimum and a maximum level of the inventory of the commodity and can be visited at most once for each time instant and several times during the time horizon. The production policy (i.e. the quantity to produce at each time instant), the retailers replenishment policies (i.e. the quantity to deliver to each retailer at each time instant) and the transportation policy (i.e. the routes traveled by the vehicles at each time instant) have to be determined so as to minimize the total system cost. The cost

* Corresponding author.
E-mail address: speranza@eco.unibs.it (M.G. Speranza).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.03.002

includes the fixed and variable production costs at the production facility, the inventory costs at the facility and at the retailers, and the routing costs. We study two different types of policies: *maximum level* (ML) and *order-up to level* (OU). In the former, if the retailer i is visited at time t, the quantity delivered to i is such that the level of the inventory at i is not greater than its maximum level. In the latter, if the retailer i is visited at time t, then the quantity shipped is such that the level of the inventory at i reaches exactly its maximum level. OU policies are inspired, in a deterministic setting, by the classical stochastic order-up-to level policy, widely studied in inventory theory. We refer to Bertazzi et al. [11] for an application of the deterministic order-up to level policy to an inventory-routing problem and to Bertazzi et al. [12] for an application of the order-up to level policy to an integrated production-distribution system. The problem we study is obviously NP-hard, both when the ML and the OU policies are applied, since for each time instant a VRP has to be solved to obtain the transportation policy.

We first show that when the transportation is outsourced, the problem with OU policy is NP-hard, whereas there exists a class of instances where the problem with ML can be solved in polynomial time. Then, we show the worst-case performance of the OU policy with respect to the more flexible ML policy, proving that, in the worst case, the ratio between the optimal cost of the former policy and the optimal cost of the latter tends to infinity. Since the problem with the ML policy has never been tackled, is very complex and the exact solution would be impractical in general, we design a heuristic algorithm. We decompose the problem into two subproblems, one concerning the production and the other concerning the distribution, solving them sequentially. The subproblem concerning the production is optimally solved, while the subproblem concerning the distribution is solved by applying a heuristic in which at each iteration one retailer is inserted in the solution. For each retailer, a MILP model, referred to as the *single retailer problem*, is optimally solved. This problem is a generalization of the dynamic lot-size problem with time-varying capacity constraints studied in Baker et al. [13]. The exact algorithm we propose is based on the properties of the optimal solution and on feasibility and dominance relations among partial solutions. Finally, the solution obtained by hierarchically solving the two subproblems is improved by applying a procedure which coordinates production and distribution. The optimal solution of MILP models, embedded within heuristics, has become popular in the last decade. It allows the heuristic to explore in-depth promising parts of the solution space. If the MILP is carefully designed, it contributes to finding high quality solutions within a reasonable computational time. A heuristic of this kind is often called *hybrid*.

To evaluate the performance of the heuristic, we present a branch-and-cut algorithm for the solution of the problem in which only one vehicle can be used at each delivery time instant (the *single vehicle case*) and compare the optimal solution of this problem with the solution generated by the hybrid heuristic on the basis of randomly generated problem instances with one vehicle. Finally, we compare the performance of the ML policy with respect to the OU policy. Computational results, obtained on a large set of randomly generated problem instances with several vehicles, show that the ML policy significantly reduces the total cost with respect to the OU policy.

The paper is organized as follows. In Section 1, we describe the problem. In Section 2, we introduce the OU and ML policies, study the computational complexity of the problem when the transportation is outsourced and prove the worst-case performance of the OU policy with respect to the ML policy. In Section 3, we describe the hybrid heuristic we propose for the solution of the problem with the ML policy. In Section 4, we model the *single retailer problem* and describe the exact algorithm to solve it. In Section 5, we formulate the *single vehicle case* and describe the branch-and-cut algorithm for the solution of this case. Finally, in Section 6 we show the computational results that allow us to evaluate the performance of the algorithms and to compare the two policies.

1. Problem description

A single commodity is produced at a facility 0 and shipped to a set $M=\{1,2,\ldots,i,\ldots,n\}$ of retailers over a finite set $T=\{1,2,\ldots,t,\ldots,H\}$ of time instants. Let $M' = M \cup \{0\}$.

The starting level I_{i0} of the inventory of each retailer $i \in M$ is given. A quantity r_{it} is consumed by retailer i at time $t \in T$. The level of the inventory of the retailer i at time t is given by the level I_{it-1} at time $t-1$, plus the quantity x_{it} shipped from the facility to the retailer i at time t, minus the quantity r_{it} consumed at time t, that is

$$I_{it} = I_{it-1} + x_{it} - r_{it} = I_{i0} + \sum_{j=1}^{t} x_{ij} - \sum_{j=1}^{t} r_{ij}. \tag{1}$$

We suppose that the sequence of the operations at each retailer i at time t is: arrival of the commodity, consumption and calculation of the inventory. The inventory level of each retailer i at time t must be non-negative (no stock-out is allowed), that is

$$I_{it} \geq 0, \quad t \in T,\ i \in M. \tag{2}$$

The total inventory cost of the retailer i is

$$\sum_{t \in T} h_i I_{it} = \sum_{t \in T} h_i \left(I_{i0} - \sum_{j=1}^{t} r_{ij} \right) + \sum_{t \in T} \sum_{j=1}^{t} h_i x_{ij}, \tag{3}$$

where h_i is the unit inventory cost of the retailer $i \in M$.

The starting level B_0 of the inventory at the facility is assumed to be 0. The level B_t of the inventory at time t is given by the level B_{t-1} at time $t-1$, plus the quantity y_t produced at time t, minus the total quantity $\sum_i x_{it}$ shipped to the retailers at time t, that is

$$B_t = B_{t-1} + y_t - \sum_{i \in M} x_{it} = \sum_{j=1}^{t} \left(y_j - \sum_{i \in M} x_{ij} \right). \tag{4}$$

We suppose that the sequence of the operations at the production facility at time t is: production, shipment and calculation of the inventory. No stock-out is allowed at the facility, that is

$$B_t \geq 0, \quad t \in T. \tag{5}$$

The total inventory cost at the facility is

$$\sum_{t \in T} h_0 B_t = \sum_{t \in T} \sum_{j=1}^{t} h_0 \left(y_j - \sum_{i \in M} x_{ij} \right), \tag{6}$$

where h_0 is the unit inventory cost of the facility. The total production cost is given by the sum of a fixed set-up cost K, which is charged for each time instant $t \in T$ in which $y_t > 0$, and a variable production cost p, which is charged for each unit produced during the time horizon. The total production cost can be formulated as

$$\sum_{t \in T} (K\delta(y_t) + p y_t), \tag{7}$$

where $\delta(y_t) = 1$ if $y_t > 0$, and 0 otherwise.

Shipments from the facility to the retailers are performed by an unlimited fleet of vehicles. Each vehicle v available at time $t \in T$ has a given capacity C and can visit multiple retailers in each route. Each retailer cannot be visited by more than one vehicle at each time instant. Therefore, the maximum number of vehicles that can be used at each time instant is n. The transportation cost

c_{is} from i to s, with $i,s \in M'$, is known. The total transportation cost is given by the routing cost over all the time instants. If R_{tv} is the route traveled by vehicle v at time t and $z_{istv}=1$ if node $s \in M'$ is the successor of node $i \in M'$ in the route R_{tv} and 0 otherwise, then the total transportation cost is

$$\sum_{t \in T} \sum_{v \in V} \sum_{i \in M'} \sum_{s \in M'} c_{is} z_{istv}, \tag{8}$$

where $V=\{1,2,\ldots,v,\ldots,n\}$ is the set of vehicles. The total quantity loaded on each vehicle must be not greater than the transportation capacity:

$$\sum_{i \in R_{tv}} x_{it} \le C, \quad v \in V,\ t \in T, \tag{9}$$

and the routes followed by the vehicles must satisfy the classical routing constraints (see [14]).

The problem is to determine, for each time instant $t \in T$, the quantity y_t to produce at the facility, the quantity x_{it} to ship to each retailer i and the set of routes R_{tv}, $v \in V$, that visit all the retailers served at time t. This problem is referred to as problem *IPD* (*integrated production-distribution*).

2. Policy analysis

We study two types of replenishment policies:

1. *ML* (*maximum level*): If retailer i is visited at time t, the quantity x_{it} is such that the level of the inventory in i, after delivery and consumption, is not greater than its maximum level U_i. This policy is obtained by adding to problem *IPD* the following constraints:

$$I_{it} \le U_i, \quad i \in M,\ t \in T. \tag{10}$$

The corresponding problem is referred to as problem *IPD-ML*. Note that constraints (1) and (10) imply that $x_{it} \le U_i + r_{it}$, $\forall i \in M$, $\forall t \in T$.

2. *OU* (*order-up to level*): If retailer i is visited at time t, then the quantity x_{it} is such that the level of the inventory in i reaches exactly the maximum level U_i at the end of the delivery time instant, after delivery and consumption. The *OU* policy is obtained by adding to problem *IPD* the following stronger constraints:

$$x_{it} \ge (U_i + r_{it})\delta(x_{it}) - I_{it-1}, \quad i \in M,\ t \in T, \tag{11}$$

$$x_{it} \le U_i + r_{it} - I_{it-1}, \quad i \in M,\ t \in T, \tag{12}$$

$$x_{it} \le (U_i + r_{it})\delta(x_{it}), \quad i \in M,\ t \in T. \tag{13}$$

This problem, referred to as problem *IPD-OU*, has been studied in Bertazzi et al. [12], where a heuristic has been proposed and evaluated on a large set of randomly generated problem instances.

Note that while the OU policy is very natural in a decentralized management where each retailer takes its own decisions, the ML policy becomes an interesting feasible option in an integrated management of the system. The increased flexibility of the ML policy can be exploited in an integrated management to reduce the global cost.

2.1. Computational complexity

Both problem *IPD-ML* and problem *IPD-OU* are NP-hard, since they reduce to the VRP in the class of instances, where the time horizon is composed of one time instant, the fixed and variable production costs are zero, the inventory costs are zero and all the retailers need to be served. It is interesting to study the computational complexity of the problems where one of the three areas (production, inventory management and transportation) is outsourced. When the production or the inventory management is outsourced (i.e. the decision-maker has not to take into account them since they are managed by external companies), the resulting problems remain NP-hard because of the complexity of the routing, while the complexity is unknown when the transportation is outsourced. Let us study this case, which is very relevant from a practical point of view. A typical situation is that the transportation service is performed by one or several external carriers. A contract regulates the relation between the shipper and the external carriers. A given transportation capacity Q is made available to the shipper at each time instant $t \in T$. Obviously, the routing part of the problem is charged to the carriers.

In the case the *ML* policy is applied, the problem, referred to as *PI-ML* (*production inventory-ML*), can be formulated as follows:

$$\min \sum_{t \in T} \left(h_0 B_t + K\delta(y_t) + p y_t + \sum_{i \in M} h_i I_{it} \right)$$

subject to (1), (2), (4), (5), (10), the capacity constraints

$$\sum_{i \in M} x_{it} \le Q, \quad t \in T, \tag{14}$$

and the non-negativity and integrality constraints of the decision variables.

Although the complexity of the problem *PI-ML* is unknown in the general case, the following class of instances, in which the demand of each retailer is constant over time and the starting level of the inventory at each retailer is equal to 0, can be optimally solved in polynomial time.

Theorem 1. *If $r_{it}=r_i$, $\forall i \in M$ and $\forall t \in T$, and $I_{i0}=0$, $\forall i \in I$, there exists a polynomial time algorithm for the solution of problem PI-ML.*

Proof. See the Appendix.

In the case the *OU* policy is applied, the problem, referred to as *PI-OU* (*production inventory-OU*), can be formulated as the problem *PI-ML*), with the only exception of the maximum level constraints (10) that are replaced by the order-up to level constraints (11)–(13). The following theorem shows that this problem is NP-hard, even if $K=p=h_0=0$ and $r_{it}=r$, for $i \in M$ and $t \in T$.

Theorem 2. *problem PI-OU is NP-hard.*

Proof. See the Appendix.

2.2. Worst-case performance of OU vs ML

We show here the worst-case performance of the optimal solution of the problem *IPD-OU* with respect to the optimal solution of problem *IPD-ML*. Let z^*_{ML} be the optimal cost of the problem with the *ML* policy and z^*_{OU} be the optimal cost of the problem with the *OU* policy.

Theorem 3. *There exists a class of instances such that $z^*_{OU}/z^*_{ML} \to \infty$.*

Proof. See the Appendix.

Therefore, in the worst case, any solution obtained by solving the problem *IPD-OU*, included the optimal solution, can be very far from the optimum if used as a heuristic solution of the problem *IPD-ML*. Note that the instance used in the proof corresponds to a situation where facility and retailers are very close to each other. Little would change if all distances would be different to one another but proportional to ε. The instance may be adapted to the case where all retailers are far from the production location and close to one another. Also in this case the ratio tends to ∞.

3. A hybrid heuristic

In this section, we propose a hybrid heuristic, referred to as algorithm $\mathcal{H}$, to solve the problem *IPD-ML*. This algorithm is composed of three successive steps. First, the problem is solved by assuming infinite production at the facility. The aim is to determine the quantity to deliver to each retailer at each time instant and the routes to travel at each time instant in order to minimize the sum of the inventory cost at the retailers and of the routing cost. This step is referred to as *distribution* and is solved by applying a heuristic in which a single retailer is processed at each iteration. When a retailer is processed, the problem of inserting this retailer in the current solution, referred to as the *single retailer problem*, is optimally solved. In the second step, referred to as *production*, the subproblem concerning the production only is optimized, given the delivery quantities obtained in the first step. The aim is to determine the quantity to produce at each time instant in order to minimize the sum of the inventory cost at the facility and of the fixed and variable production costs. Finally, in the third step, referred to as *improve*, the solution obtained in the first two steps is improved iteratively, by removing and reinserting two retailers at a time as long as the solution is improved. Let us now describe in detail the three steps.

3.1. Distribution

In the *distribution* step, the quantity to deliver to each retailer and the routes to travel at each time instant are obtained by solving the following subproblem concerning the distribution only:

$$\min\sum_{i\in M}\sum_{t\in T}h_iI_{it}+\sum_{t\in T}\sum_{v\in V}\sum_{i\in M'}\sum_{j\in M'}c_{ij}z_{ijtv} \tag{15}$$

subject to (1), (2), (4), (5), (9), (10), to $y_t=\sum_i(U_i+r_{it})$, for $t\in T$, to the routing constraints and to the non-negativity and integrality constraints of the decision variables. Setting $y_t=\sum_i(U_i+r_{it})$, for $t\in T$, is equivalent to assuming infinite production at the facility, as $\sum_i(U_i+r_{it})$ is an upper bound on the quantity that can be delivered to the retailers at each time t, due to the constraints (10). Using expression (3), the objective function (15) can be also written as

$$\sum_{t\in T}h_s\left(I_{s0}-\sum_{j=1}^{t}r_{sj}\right)+\min\sum_{t\in T}(H-t+1)h_sx_{st}+\sum_{t\in T}\sum_{v\in V}\sum_{i\in M'}\sum_{j\in M'}c_{ij}z_{ijtv},$$

where $\sum_{t\in T}h_s(I_{s0}-\sum_{j=1}^{t}r_{sj})$ is constant.

This subproblem is heuristically solved by the following algorithm, referred to as $\mathcal{H}^D$. In this algorithm, first the set M of the retailers is sorted according to a given criterion. Then, following this ordering, at each iteration a retailer, say s, is inserted in the current solution by optimally solving the corresponding *single retailer problem*, that is the problem of determining the quantity to deliver to s at each time instant $t\in T$ and, for each time instant in which s is served, either finding the route in which to insert s or building a new route to visit s. The aim is to minimize the sum of the inventory cost at s and the insertion cost of s in the routes. To solve this problem, in Section 4 we formulate a mixed-integer linear programming model and we design an exact algorithm.

Let $L\subseteq M$ be a given subset of retailers, $\mathcal{C}$ be a given criterion to sort the set M of retailers and *improve without production* be a procedure that improves the current solution ignoring the production costs. The set L and the criterion $\mathcal{C}$ are defined in Section 6, while the procedure *improve without production* is described in Section 3.3. The algorithm $\mathcal{H}^D$ is described in Algorithm 1.

Algorithm 1. $\mathcal{H}^D$

Sort the set M of retailers on the basis of the criterion $\mathcal{C}$.
Renumber the retailers accordingly.
for each retailer $s=1,2,\dots,n$ **do**
  Solve the *single retailer problem* exactly.
  If $s\in L$, then perform the procedure *improve without production.*
end for

3.2. Production

Let $\hat{x}_{it}$ be the value of the variable x_{it} and R_{tv_t} be the route traveled at time t by vehicle v_t, obtained by applying the algorithm $\mathcal{H}^D$. In the *production* step, the quantity y_t to produce at each time instant $t\in T$ is determined, given the quantity $\hat{x}_{it}$, $i\in M$ and $t\in T$, obtained in the previous step. The aim is to minimize the sum of the inventory cost at the production facility and of the fixed and variable production cost. The following subproblem is solved:

$$\min\sum_{t\in T}(h_0B_t+K\delta(y_t)+py_t) \tag{16}$$

subject to (4) and (5), $y_t\geq 0$, for $t\in T$, and $x_{it}=\hat{x}_{it}$, for $i\in M$ and $t\in T$.

This subproblem can be optimally solved in polynomial time, since it corresponds to the classical uncapacitated dynamic lot size problem introduced by Wagner and Whitin [15]. We solve it by using the algorithm described in Bertazzi et al. [12].

3.3. Improve

Let $\hat{y}_t$ be the value of each variable y_t, $t\in T$, obtained by optimally solving the *production* problem. In the *improve* step, the solution obtained in the previous two steps is improved iteratively. At each iteration, two retailers, say s' and s'', are temporarily removed from the current solution. Then, first the retailer s'' and then the retailer s' are reinserted in the current solution by solving the *single retailer problem* for retailers s'' and s', respectively. Finally, the *production* step is applied again to determine the optimal quantity to produce at each time instant, given the new quantity to deliver to each retailer at each time instant. If this reduces the total cost, then the solution is modified accordingly. This iteration is repeated as long as an improvement in the total cost is achieved. The procedure *improve without production*, used in the heuristic $\mathcal{H}^D$, is identical to the procedure *improve*, with the only exception of the *production* step, which is not reapplied.

4. The single retailer problem

In this section, we describe the *single retailer problem*, that is the problem of determining, given a partial solution in which some of the retailers may have been already inserted, the quantity to deliver to a given retailer s at each time instant $t\in T$ and, for each time instant in which s is served, either the route in which to insert s or a new route to visit s. The aim is to minimize the sum of the inventory cost at the retailer s and the insertion cost of the retailer s in the routes. To solve this problem, we first formulate a mixed-integer linear programming model and then we propose an exact algorithm.

4.1. The model

Let M^s be the set of retailers already inserted in the partial solution available when the retailer s has to be processed. Let $\hat{x}_{it}$ be the quantity to deliver to each retailer $i \in M^s$ at each time $t \in T$, R_{tv_t} be the sequence of the retailers visited by each vehicle $v_t = 1,2,\ldots,\hat{v}_t$ used at each time $t \in T$, $\hat{y}_t$ be the quantity to produce at the facility at each time $t \in T$. The set V_t of vehicles available to serve the retailer s at time t is composed of the vehicles $1,2,\ldots,\hat{v}_t$ already used at time t in the partial solution, plus an additional new vehicle $\hat{v}_t+1$. Note that more than $\hat{v}_t+1$ vehicles are never needed at time t to serve s, as each retailer can be visited at most once at each time instant. For each $t \in T$, the residual capacity C_{tv_t} of vehicle $v_t \in V_t$ is $C - \sum_{i \in R_{tv_t}} \hat{x}_{it}$ for $v_t = 1,2,\ldots,\hat{v}_t$, and C for $v_t = \hat{v}_t+1$, because the route $R_{t\hat{v}_t+1}$ is empty. Let Δ_{stv_t} be an estimate of the cost to insert the retailer s in the route R_{tv_t}, $v_t = 1,\ldots,\hat{v}_t+1$, that is computed by using the cheapest insertion cost rule: for each $v_t = 1,\ldots,\hat{v}_t$, $\Delta_{stv_t} = min_{(i,j) \in R_{tv_t}} \{c_{is} + c_{sj} - c_{ij}\}$, while, for $v_t = \hat{v}_t+1$, $\Delta_{stv_t} = c_{0\,s} + c_{s0}$. We denote by $\delta(v_t)$ the binary variable that takes value 1 if the retailer s is served by the vehicle v_t at time t and 0 otherwise, and by $\delta(x_{st})$ the binary variable that takes value 1 if $x_{st} > 0$ and 0 otherwise. Finally, let B_t^s be the level of the inventory at the facility at each time $t \in T$ before the retailer s is served. For each $t \in T$, B_t^s is equal to $\sum_{j=1}^{t} (\hat{y}_j - \sum_{i \in M^s} \hat{x}_{ij})$, due to (4). The *single retailer problem* aims at inserting the retailer s in the partial solution at minimum cost and can be formulated as follows:

Single retailer problem:

$$\min \sum_{t \in T} h_s I_{st} + \sum_{t \in T} \sum_{v_t \in V_t} \Delta_{stv_t} \delta(v_t), \tag{17}$$

$$I_{st} = I_{s0} + \sum_{j=1}^{t} x_{sj} - \sum_{j=1}^{t} r_{sj}, \quad t \in T, \tag{18}$$

$$0 \le I_{st} \le U_s, \quad t \in T, \tag{19}$$

$$\sum_{v_t \in V_t} \delta(v_t) = \delta(x_{st}), \quad t \in T, \tag{20}$$

$$x_{st} \le \sum_{v_t \in V_t} C_{tv_t} \delta(v_t), \qquad t \in T, \tag{21}$$

$$\sum_{j=1}^{t} x_{sj} \le B_t^s, \quad t \in T, \tag{22}$$

$$x_{st} \ge 0, \quad t \in T, \tag{23}$$

$$\delta(v_t) \in \{0,1\}, \quad t \in T, \tag{24}$$

$$\delta(x_{st}) \in \{0,1\}, \quad t \in T. \tag{25}$$

The objective function (17) expresses the minimization of the sum of the inventory cost of retailer s and of the insertion cost of s in the routes traveled by the vehicles. Constraints (18) and (19) define the level of the inventory and the minimum and maximum level at time t, respectively. Constraints (20) and (21) guarantee that exactly one vehicle serves the retailer s at time t whenever $x_{st} > 0$. Constraint (22) guarantee that no stock-out occurs at the facility and, finally, constraints (23)–(25) are the non-negativity and integrality constraints. This problem is a generalization of the dynamic lot-size problem with time-varying production capacity constraints studied in Baker et al. [13], which is known to be NP-hard (see [16]). In fact, the *single retailer problem* reduces to it when $|V_t| = 1$ and $B_t^s = +\infty$ for $t \in T$, $U_s = +\infty$ and $I_{s0}=0$. In this case, the constraint (19) becomes $I_{st} \ge 0$ for $t \in T$, the constraint (20) becomes $\delta(v_t) = \delta(x_{st})$ for $t \in T$, and the constraint (21) becomes $x_{st} \le C_{tv_t}\delta(v_t)$ for $t \in T$. Δ_{stv_t} represents the fixed cost associated with the delivery by using vehicle v_t at time $t \in T$, and the variable production cost is equal to 0.

4.2. An exact algorithm

In this section, we describe the exact algorithm we propose for the solution of the *single retailer problem*. This algorithm determines an optimal delivery plan, representing the quantity to deliver to the retailer s at each time $t \in T$, the vehicles used to serve s and the sequence of the retailers served by these vehicles, working on a search tree. The root is labeled by $(H+1,0)$ and indicates that no delivery to s is performed at time $H+1$. Each other node of the tree is labeled by a pair (t,v_t), $t \in T$ and $v_t \in V_t$, and indicates that a delivery to s is performed at time $t \in T$ by vehicle $v_t \in V_t$. Each leaf (t,v_t) of the tree indicates that the first delivery to retailer s is performed at time t. Each node (t,v_t), excluding the leaves, generates $\sum_{j=1}^{t-1} (\hat{v}_j+1)$ branches, one for each time instant $j=1,2,\ldots,t-1$ and each vehicle $v_j \in V_j$. The maximum depth of the search tree is at most H and each leaf (t_l,v_{t_l}) identifies a solution of the *single retailer problem* through the path $\sigma_{(t_l,v_{t_l})} = \{(H+1,0),(t_1,v_{t_1}),(t_2,v_{t_2}),\ldots,(t_l,v_{t_l})\}$, where $t_1 \le H$, $t_l \ge 1$ and $t_k > t_{k+1}$, $k=1,\ldots,l-1$, from the root to the leaf (t_l,v_{t_l}). Going backward, the node $(t_{l-1},v_{t_{l-1}})$ says that the next delivery is performed at time t_{l-1} by using vehicle $v_{t_{l-1}}$, and so on, until the node (t_1,v_{t_1}) says that the last delivery is performed at time t_1 by using vehicle v_{t_1}. An example of the complete search tree for an instance with $H=3$ and $|V|=2$ is given in Fig. A1 (see the Appendix). The leaf in the dotted circle identifies a path $\sigma_{(1,2)} = \{(H+1,0),(3,2),(2,1),(1,2)\}$. In the delivery plan corresponding to this path, the retailer s is served at time 1 with vehicle 2, at time 2 with vehicle 1 and at time 3 with vehicle 2. For the same reason, each path $\sigma_{(t,v_t)}$ from the root to node (t,v_t) defines a partial delivery plan from t to H. Thus, node (t,v_t) of path $\sigma_{(t,v_t)}$ represents a subproblem of the *single retailer problem*, called $\mathcal{R}_{\sigma_{(t,v_t)}}$, where the decisions on retailer s have been made at each time instant j between t and H, whereas the decisions regarding each time instant j from 1 to $t-1$ must be taken.

During the algorithm, a list of active subproblems (active list) is maintained. For any subproblem $\mathcal{R}_{\sigma_{(k,v_k)}}$ in the active list, the corresponding node (k,v_k) is not a leaf. Whenever a subproblem $\mathcal{R}_{\sigma_{(k,v_k)}}$ is removed from the active list, it is decomposed into the set of feasible and promising smaller subproblems $\mathcal{R}_{\sigma_{(t,v_t)}}$, with $t=1,\ldots,k-1$ and $v_t = 1,\ldots,\hat{v}_t+1$. Any of these subproblems not corresponding to a leaf is added to the active list, while for any subproblem corresponding to a leaf, say $\mathcal{R}_{\sigma_{(t,v_t)}}$, the corresponding cost $g_{\sigma_{(t,v_t)}}$ is compared to the cost g^* of the best current complete delivery plan. If $g_{\sigma_{(t,v_t)}} < g^*$, then g^* is set equal to $g_{\sigma_{(t,v_t)}}$ and $\sigma_{(t,v_t)}$ is saved as the best current complete delivery plan. Otherwise, the subproblem $\mathcal{R}_{\sigma_{(t,v_t)}}$ is eliminated.

In the following, we first show how to compute the optimal quantity to deliver $q^s_{(t,v_t)}$ and then we describe the algorithm.

The optimal quantity to deliver: Let $(t,\ v_t)$ be a node generated by $(k,\ v_k)$, where $k > t$, of the path $\sigma_{(k,v_k)}$ and $q^s_{(t,v_t)}$ be the optimal quantity to deliver at time t by using vehicle v_t, given the partial delivery plan $\sigma_{(k,v_k)}$. Two different situations can occur. In the first case, the partial delivery plan $\sigma_{(k,v_k)}$ completely satisfies the demand of retailer s from k to H. In this case, the problem is to determine the quantity to deliver at time t by using vehicle v_t in order to satisfy, partially or completely, the demand from t to $k-1$. The second case is when part of the demand of retailer s from k to H is unmet by the partial delivery plan $\sigma_{(k,v_k)}$, due to the constraints on the maximum level of the inventory at retailer s, on the residual capacity of the vehicles or on the stock-out at the facility. In this case, the only way to satisfy the unmet demand $u_{\sigma_{(k,v_k)}}$ is to have the level of inventory of retailer s at time $k-1$ at least equal to $u_{\sigma_{(k,v_k)}}$. So $u_{\sigma_{(k,v_k)}}$ can be treated as additional demand at time $k-1$. Therefore, the problem is to determine the quantity to deliver at time t by using vehicle v_t to satisfy the demand from t to $k-1$ plus the unmet demand $u_{\sigma_{(k,v_k)}}$. Once again, either the quantity delivered at time t by using vehicle v_t completely satisfies this total demand or there is unmet demand that will be satisfied going backward. If feasible, the optimal quantity $q^s_{(t,v_t)}$ to deliver at time t by using vehicle v_t is equal to the quantity that satisfies the total demand $\sum_{j=t}^{k-1} r_{sj}$ from t to $k-1$ plus the unmet demand $u_{\sigma_{(k,v_k)}}$. The reason is that it is always better to satisfy the demand of retailer s as late as possible since this decreases the inventory cost. Consequently, it is never convenient to leave an unmet demand if this is not necessary: This will not change the transportation cost but will increase the inventory cost. Let $I^0_{st-1} = \max\{I_{s0} - \sum_{j=1}^{t-1} r_{sj}, 0\}$, that is the part of the starting inventory at time 0 still available at time $t-1$, and $B_{j,\sigma_{(k,v_k)}}$ be the inventory level at the facility at each time j, given the partial delivery plan $\sigma_{(k,v_k)}$. $B_{j,\sigma_{(k,v_k)}} = B^s_j$, for each time $1 \le j \le k-1$, and $B_{j,\sigma_{(k,v_k)}} = B^s_j - \sum_{l \le j:(l,v_l) \in \sigma_{(k,v_k)}} q^s_{(l,v_l)}$, for each time j such that $k \le j \le H$.

The following result holds.

Theorem 4. *The optimal quantity $q^s_{(t,v_t)}$ to deliver at node (t, v_t) generated by node (k, v_k) of the partial delivery plan $\sigma_{(k,v_k)}$ is equal to*

$$\min\left\{\max\left\{\sum_{j=t}^{k-1} r_{sj} + u_{\sigma_{(k,v_k)}} - I^0_{st-1}, 0\right\}, U_s - I^0_{st-1} + r_{st}, C_{tv_t}, \min_{j=t,t+1,\ldots,H} B_{j,\sigma_{(k,v_k)}}\right\}.$$

Proof. See the Appendix.

Note that a stock-out situation at the production facility will never occur if the production plan is $\hat{y}_t = \sum_i (U_i + r_{it})$ for $t \in T$. Thus, in the first phase of the heuristic, no violation of these constraints is detected. Only in the final phase of the algorithm described in Section 3.3 such a violation is possible (the *single retailer problem* is infeasible) since the production plan is changed by the solution of the production subproblem, as described in Section 3.2.

The delivery plan $\sigma_{(t,v_t)}$ is composed by the delivery of the quantity $q^s_{(t,v_t)}$ at time t by using vehicle v_t and then by the delivery plan $\sigma_{(k,v_k)}$. Therefore, the corresponding cost $g^s_{(t,v_t)}$ is equal to the sum of the contribution to the total cost due to the delivery of the quantity $q^s_{(t,v_t)}$ at time t by using v_t, and to the cost of the delivery plan $\sigma_{(k,v_k)}$, i.e. $g^s_{(t,v_t)} = \Delta_{stv_t} + h_s(H+1-t) q^s_{(t,v_t)} + g^s_{\sigma_{(k,v_k)}}$.

With respect to the feasibility of a complete delivery plan, a node (t, v_t) is a feasible leaf if the delivery plan $\sigma_{(t,v_t)}$ completely satisfies the demand of customer s from 1 to H. In particular, when $I_{s0}=0$ and $r_{s1}=0$, each feasible leaf (t, v_t) has $t=1$ and $u_{\sigma_{(1,v_1)}} = 0$, whereas in the other cases, (t, v_t) is a feasible leaf if $1 \le t \le t'_s$, where t'_s is the first integer number so that $I_{s0} - \sum_{j=1}^{t'_s} r_{sj} \le 0$, and $u_{\sigma_{(t,v_t)}} = 0$. In this case, the demand of customer s from 1 to $t-1$ is satisfied by I_{s0}.

The optimal delivery plan is given by the best among the delivery plans associated to the paths from the root and the feasible leaves. Thus the optimal cost will be given by

$$g^* = \min_{\sigma_{(t,v_t)}} \{g_{\sigma_{(t,v_t)}} : \sigma_{(t,v_t)} \text{ is a feasible leaf}\}.$$

Description of the algorithm: The algorithm to solve the *single retailer problem* can be formally described as follows. Let $\mathcal{F}_{\sigma_{(t,v_t)}}$ be a set of conditions that allows us to state that any completion of the partial delivery plan $\sigma_{(t,v_t)}$ is infeasible. Finally, let $\mathcal{O}_{\sigma_{(t,v_t)}}$ be a set of conditions that allows us to state that any completion of the partial delivery plan $\sigma_{(t,v_t)}$ cannot be optimal. We will define these conditions later in this section. The algorithm can be described as follows.

(1) *Initialization*: Rename the *single retailer problem* as the subproblem $\mathcal{R}_{\sigma_{(H+1,0)}}$. Add $\mathcal{R}_{\sigma_{(H+1,0)}}$ to the active list and set $u_{\sigma_{(H+1,0)}} = 0$, $g_{\sigma_{(H+1,0)}} = 0$ and $g^* = \infty$. Set $k=H+1$.
(2) *Termination check*:
   2.1. If the active list is empty and $g^* = \infty$, then stop (the *single retailer problem* is infeasible: stock-out occurs at the facility).
   2.2. If the active list is empty and $g^* < \infty$, then
      - For each (k, v_k) belonging to the best delivery plan, set $\hat{x}_{sk} = q^s_{(k,v_k)}$ and insert s in the route R_{kv_k} by using the cheapest insertion rule.
      - For each (k, v_k) not belonging to the best delivery plan, set $\hat{x}_{sk} = 0$ and stop (the optimal complete delivery plan has been found).
   2.3. If the active list is not empty, then remove the subproblem $\mathcal{R}_{\sigma_{(k,v_k)}}$ most recently added to the active list, along with the corresponding $g_{\sigma_{(k,v_k)}}$ and $u_{\sigma_{(k,v_k)}}$.
(3) *Loop*: set $t=k-1$ and $v_t=1$. Then,
   3.1. Compute $q^s_{(t,v_t)}$, $g_{\sigma_{(t,v_t)}}$ and $u_{\sigma_{(t,v_t)}}$. Verify the feasibility of the completion of the subproblem $\mathcal{R}_{\sigma_{(t,v_t)}}$ by using the set of conditions $\mathcal{F}_{\sigma_{(t,v_t)}}$. If it can be eliminated, go to Step 3.4.
   3.2. If $\mathcal{R}_{\sigma_{(t,v_t)}}$ corresponds to a feasible leaf and $g_{\sigma_{(t,v_t)}} < g^*$, set $g^* = g_{\sigma_{(t,v_t)}}$ and save $\sigma_{(t,v_t)}$ as the current best delivery plan, set $v_t=v_t+1$, and then, if $v_t \le \hat{v}_t+1$, go to Step 3.1, otherwise go to Step 2.
   3.3. Verify if the completion of the subproblem $\mathcal{R}_{\sigma_{(t,v_t)}}$ can be optimal by using the set of conditions $\mathcal{O}_{\sigma_{(t,v_t)}}$. If it can be eliminated, go to Step 3.4, otherwise add $\mathcal{R}_{\sigma_{(t,v_t)}}$ to the active list.
   3.4. Set $v_t=v_t+1$. If $v_t \le \hat{v}_t+1$, go to Step 3.1, otherwise set $t=t-1$. If $t \ge 1$, then set $v_t=1$ and go to Step 3.1, otherwise go to Step 2.

Fathoming criteria: We now prove some results that allow us to fathom the search tree. Suppose that a partial delivery plan $\sigma_{(t,v_t)}$, corresponding to the node (t, v_t), has been obtained. The following theorem defines the conditions $\mathcal{F}_{\sigma_{(t,v_t)}}$ for the infeasibility of any completion of the partial plan. In this case, the node (t, v_t) can be fathomed.

Theorem 5. *No completion of the partial delivery plan $\sigma_{(t,v_t)}$ can be feasible when any of the following conditions holds*:

$$\max\left\{\sum_{j=1}^{t-1} r_{sj} + u_{\sigma_{(t,v_t)}} - I_{s0}, 0\right\} > \min_{j=t-1,\ldots,H} B_{j,\sigma_{(t,v_t)}}, \tag{26}$$

$$u_{\sigma_{(t,v_t)}} > U_s, \tag{27}$$

$$u_{\sigma_{(1,v_1)}} > 0, \tag{28}$$

$$q^s_{(t,v_t)} = 0. \tag{29}$$

Proof. See the Appendix.

We now derive other fathoming conditions from a cost analysis of the possible completions of a partial delivery plan $\sigma_{(t,v_t)}$. These conditions are based on a lower bound of the optimal cost of the problem $\mathcal{R}_{\sigma_{(t,v_t)}}$, given by the sum of a lower bound on the inventory cost and a lower bound on the transportation cost. The contribution to the inventory cost due to the deliveries performed from 1 to $t-1$ can be written as the sum of two components:

$$h_s \sum_{k=1}^{t-2} \sum_{j=1}^{k} x_{sj} + h_s \sum_{k=t-1}^{H} \sum_{j=1}^{t-1} x_{sj}.$$

The first component represents the contribution due to the inventory levels from time 1 to $t-2$, while the second from $t-1$ to H. Since $\sum_{j=1}^{k} x_{sj} \geq \max\{\sum_{j=1}^{k} r_{sj} - I_{s0}, 0\}$, then the first component is not lower than $h_s \sum_{k=1}^{t-2} \max\{\sum_{j=1}^{k} r_{sj} - I_{s0}, 0\}$. Since $\sum_{j=1}^{t-1} x_{sj}$ does not depend on k, the second component can be written as $h_s(H-t+2)\sum_{j=1}^{t-1} x_{sj}$. Moreover, since $\sum_{j=1}^{t-1} x_{sj} \geq \max\{\sum_{j=1}^{t-1} r_{sj} - I_{s0} + u_{\sigma_{(t,v_t)}}, 0\}$, then it is not lower than $h_s(H-t+2)$ $\max\{\sum_{j=1}^{t-1} r_{sj} - I_{s0} + u_{\sigma_{(t,v_t)}}, 0\}$. Therefore, a lower bound on the contribution to the inventory cost due to the deliveries performed from 1 to $t-1$ is

$$LBI_{t-1} = h_s \left(\sum_{k=1}^{t-2} \max\left\{ \sum_{j=1}^{k} r_{sj} - I_{s0}, 0 \right\} + (H-t+2) \max\left\{ \sum_{j=1}^{t-1} r_{sj} - I_{s0} + u_{\sigma_{(t,v_t)}}, 0 \right\} \right).$$

We now compute a lower bound on the contribution to the transportation cost due to the deliveries performed from 1 to $t-1$. A lower bound on the number of delivery time instants from 1 to $t-1$ is

$$m = \left\lceil \frac{\max\left\{ \sum_{j=1}^{t-1} r_{sj} - I_{s0} + u_{\sigma_{(t,v_t)}}, 0 \right\}}{\min\{U_s + \max_{j=1,2,\ldots,t-1} r_{sj}, C\}} \right\rceil, \tag{30}$$

as $\max\{\sum_{j=1}^{t-1} r_{sj} - I_{s0} + u_{\sigma_{(t,v_t)}}, 0\}$ is the quantity to deliver from 1 to $t-1$ and $\min\{U_s + \max_{j=1,2,\ldots,t-1} r_{sj}, C\}$ is the maximum quantity that can be delivered at each time instant between 1 and $t-1$. Thus, a lower bound on the contribution to the transportation cost due to the deliveries from 1 to $t-1$ can be obtained as follows:

1. For $k=1,2,\ldots,t-1$, compute the minimum insertion cost of retailer s among all the routes traveled at time k with residual transportation capacity greater than 0:
   $$\Delta^*_{sk} = \min_{v_k = 1,2,\ldots,\hat{v}_k+1} \{\Delta_{skv_k} : C_{kv_k} > 0\}.$$
2. Sort Δ^*_{sk}, with $k=1,2,\ldots,t-1$, in non-decreasing order. Let $\hat{V}_m$ be the set of the first m Δ^*_{sk}'s on the basis of this sorting, where m is the lower bound on the number of delivery time instants computed as in (30).
3. Compute the lower bound as $LBT_{t-1} = \sum_{k \in \hat{V}_m} \Delta^*_{sk}$.

We can now state two theorems that provide the conditions $\mathcal{O}_{\sigma_{(t,v_t)}}$ on the optimality of any completion of the partial delivery plan $\sigma_{(t,v_t)}$. Let g^* be the cost of the current best delivery plan from 1 to H.

Theorem 6. *No completion of the partial delivery plan $\sigma_{(t,v_t)}$ can be optimal when*:

$$g^* \leq LBI_{t-1} + LBT_{t-1} + g^s_{(t,v_t)}. \tag{31}$$

Proof. See the Appendix.

Let us now denote by g^*_t the cost of the current best partial delivery plan σ^*_t from t to H and by u^*_t the corresponding unmet demand.

Theorem 7. *No completion of the partial delivery plan $\sigma_{(t,v_t)}$ can be optimal when*

$$g^*_t \leq g^s_{(t,v_t)} \quad \text{and} \quad u^*_t \leq u_{\sigma_{(t,v_t)}}. \tag{32}$$

Proof. See the Appendix.

5. The single vehicle case

In this section, we study the particular case of the problem *IPD-ML*, referred to as the *single vehicle case*, in which one vehicle only is available for each time $t \in T$. We present a branch-and-cut algorithm to solve this problem. Even if a single vehicle only is allowed, the problem remains really difficult to be solved exactly. The optimal solution allows us to exactly evaluate on a set of instances the performance of the hybrid heuristic described in the Section 2.

5.1. The model

The model can be formulated as follows:

$$\min \sum_{i \in M} \sum_{t \in T} h_i I_{it} + \sum_{t \in T} (h_0 B_t + K\delta(y_t) + p y_t) + \sum_{t \in T} \sum_{i \in M'} \sum_{s \in M'} c_{is} z_{ist} \tag{33}$$

subject to (1), (2), (4), (5), (9), (10), the routing constraints and the constraints that define the decision variables. Note that in the model, for the sake of simplicity, the index v is omitted in the variables z_{istv}. The routing constraints include, for each route, the following flow conservation constraints and the subtour elimination constraints. The flow conservation constraints can be formulated as

$$\sum_{s \in M'} z_{ist} + \sum_{s \in M'} z_{sit} = 2\delta(x_{it}), \quad i \in M', \ t \in T, \tag{34}$$

while the subtour elimination constraints (see [17,18]) are:

$$\sum_{i \in S} \sum_{s \in S} z_{ist} \leq \sum_{i \in S} \delta(x_{it}) - \delta(x_{kt}), \quad k \in S, \ S \subseteq M, \ t \in T. \tag{35}$$

The flow conservation constraints are added to the model, while the subtour elimination constraints are added as cuts only when violated.

To strengthen the LP relaxation of this model, the following valid inequalities are proposed.

Proposition 1. *The inequalities*

$$y_t \leq \sum_{i \in M} \sum_{j=t}^{H} r_{ij} \delta(y_t), \quad t \in T \tag{36}$$

are valid.

Proof. It is never convenient to have a final inventory level greater than 0, both at the facility and at the retailers. Thus, the quantity produced at the facility at each time instant t is never greater than the maximum quantity needed to satisfy the demand of the retailers from t to H. □

Proposition 2. *The inequalities*

$$B_{t-1} \leq \sum_{i \in M} \sum_{j \in T} r_{ij} (1 - \delta(y_t)), \quad t \in T \tag{37}$$

are valid.

Proof. Similarly to the uncapacitated lot-sizing problem, it is never optimal to produce at time t if the inventory level B_{t-1} at the facility at time $t-1$ is greater than 0. □

Proposition 3. *The inequalities*

$$y_t \geq \frac{K}{h_0 j}(\delta(y_t)+\delta(y_{t+j})-1), \quad t=1,2,\ldots,H-1,\ j=t+1,t+2,\ldots,H \tag{38}$$

are valid.

Proof. If there is a production process at time t and at time $t+j$, then the minimum quantity to produce in t is $K/h_0 j$, otherwise it is better to increase the production in t and not to produce in $t+j$. □

Let us now define t' to be the first time instant within which a production has to be necessarily made, i.e. if the production facility does not produce anything within t' at least one retailer incurs in a stock-out condition. It is easy to calculate the value of t'. Let t'_i be the first time instant for which $I_{i0}-\sum_{t=1}^{t'} r_{it}<0$. Then, $t'=\min_{i\in M} t'_i$.

The following two valid inequalities are related to the time instant t'.

Proposition 4. *The inequalities*

$$\sum_{t=1}^{t'} \delta(y_t) \geq 1, \tag{39}$$

$$\sum_{t=1}^{t'}\sum_{i\in M} z_{0it} \geq \left\lceil \frac{\sum_{i\in M}\max\left\{0, \sum_{t=1}^{t'} r_{it}-I_{i0}\right\}}{C}\right\rceil \tag{40}$$

are valid.

Proof. Inequalities (39) say that at least a production process has to be started within t'. Inequalities (40) say that the minimum number of routes within t' is equal to the lowest integer greater than the ratio between the minimum quantity which has to be produced and the capacity of the vehicles. □

This set of valid inequalities is integrated with the inequalities proposed in Archetti et al. [19] for the inventory-routing problem, which are valid for problem *IPD-ML*. We briefly report here, to make the paper self-contained, these inequalities:

$$I_{it-k} \geq \left(\sum_{j=0}^{k} r_{it-j}\right)\left(1-\sum_{j=0}^{k}\delta(x_{it-j})\right), \quad i\in M,\ t\in T,\ k=0,1,\ldots,t-1, \tag{41}$$

$$\delta(x_{it}) \leq \delta\left(\sum_{i\in M} x_{it}\right), \quad i\in M,\ t\in T, \tag{42}$$

$$z_{ist} \leq \delta(x_{it}), \quad i\in M',\ s\in M',\ t\in T. \tag{43}$$

5.2. The exact algorithm

We can now describe the branch-and-cut algorithm we have implemented to solve the *single vehicle case*. The model we used is defined by the objective function (33) and the constraints (1), (2), (4), (5), (9), (10), (34), (36)–(43) and the constraints that define the decision variables. All valid inequalities are added to the model at the beginning of the optimization process. The reason is that preliminary results show that it is more effective to insert all of them at the beginning instead of checking, at each node of the branch and bound tree, which of them are violated. The subtours elimination constraints (35) with $k=\text{argmax}_i\{\delta(x_{it})\}$ are introduced by using the separation algorithm of Padberg and Rinaldi [20]. Separation is performed at each node of the tree. Whenever a violated subtours elimination constraint is identified, it is added to the current subproblem which is then reoptimized; otherwise, branching occurs at the current node. Branching is performed first on variables $\delta(y_t)$, then on variables $\delta(x_{it})$ and finally on variables z_{ist}. The search is developed according to a best bound first strategy. An initial upper bound is obtained by applying the algorithm $\mathcal{H}$.

6. Computational results

The hybrid heuristic described in Section 3 has been implemented in Fortran and run on an Intel Core(TM)2, 2.40 GHz 3.25 GB RAM personal computer, whereas the branch-and-cut algorithm described in Section 5 has been implemented in C++ using ILOG Concert 2.3 and CPLEX 10.1 run on an AMD Athlon 64 Dual Core 2.89 GHz and 3.37 GB RAM. The goals of this experimental campaign are to identify the maximum size of the instances that can be solved exactly, to test the effectiveness of the heuristic by calculating, whenever possible, the error generated with respect to the optimum and to compare the OU and ML policies. The tests have been carried out on four classes of randomly generated problem instances.

The instances of the first class have been generated as follows:

Time horizon H: 6.
Number of retailers n: 14 to evaluate the performance of the exact algorithm and to test the effectiveness of the heuristic; 50 and 100 to compare the OU and ML policies.
Quantity r_{it} consumed by retailer $i\in M$ at time $t\in T$: constant over time, i.e. $r_{it}=r_i$, $t\in T$, $i\in M$, and randomly generated as an integer number in the interval [5,25]. Note that even if the consumption is constant over time, the problem is NP-hard.
Maximum level U_i of the inventory at retailer $i\in M$: $r_i g_i$, where g_i is randomly selected from the set $S=\{2,3,6\}$ and represents the number of time units needed in order to consume the quantity U_i.
Starting level I_{i0} of the inventory at the retailer $i\in M$: U_i-r_i.
Inventory cost at retailer $i\in M$, h_i: randomly generated in the intervals [1,5] and [6,10].
Inventory cost at the facility h_0: 3 and 8.
Variable production cost p: $10h_0$.
Fixed set-up cost K: $100p$.
Transportation cost c_{is}: $\lfloor\sqrt{(X_i-X_s)^2+(Y_i-Y_s)^2}+0.5\rfloor$, where the points (X_i, Y_i) and (X_s, Y_s), with $i\in M'$ and $s\in M'$, are obtained by randomly generating each coordinate as an integer number in the interval [0,500] and in the interval [0,1000].
Transportation capacity C: $\overline{U}$, $3/2\overline{U}$ and $2\overline{U}$, where $\overline{U}=\max_{i\in M}\{U_i+r_i\}$.

In all cases, random selections have been performed in accordance to a uniform distribution. For each $n\in\{14,50,100\}$, a total of 24 combinations are obtained on the basis of the different values or intervals of the parameters: two values for h^0, two intervals for h_i, two intervals to generate the coordinates (X_i, Y_i) and three values for the capacity C. In Table A1 (see the Appendix) we summarize the characteristics of the 24 instances we test for each class.

The second class of instances is identical to the first with the only difference that the parameter p is changed from $10h_0$ to $100h_0$. This class simulates situations with a greater impact of the production costs with respect to the inventory costs. The third class is identical to the first with the only difference that the coordinates of the retailers and of the facility are multiplied by a factor of 5. This class simulates situations with large transportation costs. Finally, in the fourth class $h_i=0$, $i\in M$. In this case only the inventory costs at the facility are relevant to the decision maker (typically the facility), while the inventory costs at the retailers are not accounted for in the objective function. Since in the first class (see Table A1 in the Appendix) the instances k and $k+6$, for $k=1,2,\ldots,6$ and for $k=13,14,\ldots,18$, differ only for the inventory costs at the retailers, we generated the fourth class by taking the instances 1–6 and 13–18 of the first class and the instances 7–12 and 19–24 of the second class, and then we set $h_i=0$, $i \in M$. For a given size of instances, measured by the number of retailers n, we have $24*4=96$ instances.

The generated instances and the computational results are available at the following URL: www-c.eco.unibs.it/~bertazzi/ml.zip.

We first test the performance of the algorithms. The main goal is to test the performance of the heuristic $\mathcal{H}$. The branch-and-cut algorithm is used to compute, on a set of instances with 15 nodes that is with 14 retailers, the errors with respect to optimum generated by the heuristic $\mathcal{H}$. Then, we evaluate the effectiveness of the fathoming rules. Finally, we compare the ability of the policy *ML* to reduce the costs with respect to the more traditional, but more rigid, *OU* policy.

6.1. Performance of the algorithms

To measure the errors generated by the heuristic $\mathcal{H}$ with respect to optimum, we considered instances with $n=14$ retailers and imposed the availability of only one vehicle. We generated five instances for each of the 96 different types, 24 for each of the four classes, for a total of 480 instances. We compared the heuristic solution value with the best solution found by using the branch-and-cut algorithm with a maximum running time of 2 h. Through these tests we also derived the maximum size of the instances that can be solved exactly. Recall that the algorithm $\mathcal{H}^D$ is based on the subset L of retailers and a given criterion $\mathcal{C}$ to order the set M of retailers. In the computational experiment, L is defined as follows. Let l_i be a parameter of the algorithm introduced to stop the iterative process. l_1 is a given number, while $l_i = l_{i-1} + \lfloor (n - l_{(i-1)})/2 \rfloor$, for $2 \le i \le n$. In the computational experiments, the value of the parameter l_1 is set to $\lfloor n/6 \rfloor$. Let L be the set of the l_i's. The criterion $\mathcal{C}$ is the non-decreasing order of $I_{i0}H/\sum_t r_{it}$.

The results are shown in Tables 1 and 2. Table 1 refers to instances of the first and second class. The first column of Table 1 shows the name of the instance. Instances 1–24 and 25–48 are instances of the first and second class, respectively. Similarly, in Table 2 instances 49–72 and 73–96 are instances of the third and fourth class, respectively. The other columns have the same meaning in both tables. The second column gives the average CPU time, expressed in seconds, over the five tested instances. When an instance could not be solved to optimality the time computed for that instance in the average is 7200 s, the maximum time allowed. The third column shows the number of instances solved to optimality. The fourth column 'average opt. gap' provides the percentage difference between the optimum value of the objective function and the lower bound when the branch-and-cut ends. This value is 0 whenever the instance is solved to optimality. The average is 0 when all the five tested instances have been solved to optimality. The last column of each table 'average error' gives the average error generated by the heuristic with respect to the optimum, when available, or with respect to the lower bound when the instance could not be solved to optimality within 2 h.

Tables 1 and 2 allow us to comment on the performance of the branch-and-cut algorithm. We observe that most of the tested instances with 15 nodes could be solved to optimality. The computational time is often very small, of the order of seconds. However, a few instances could not be solved to optimality. The average time, reported in the column 'CPU' is large only when one or more instances could not be solved to optimality.

Looking at the last column of Tables 1 and 2, we can observe that the performance of the heuristic strongly depends on the characteristics of the instance. Note that the error is very close to the error with respect to optimum as most of the values are calculated with respect to the optimum. In a few cases the average error overestimates the error with respect to optimum because it is computed with respect to the lower bound. The average error of the heuristic on the four classes of instances of Tables 1 and 2 is 2.25%, 0.30%, 3.65% and 1.00%, respectively.

We have analyzed the efficacy of the valid inequalities proposed in Section 5.1 and the results are shown in Table 3 where, for each class of instances, we reported the percentage gap between the optimal solution value and, respectively, the linear relaxation of the problem including all valid inequalities (second column), the linear relaxation of the problem including only valid inequalities proposed by Archetti et al. [21], and, finally, the linear relaxation of the problem without any valid inequality. As shown in the table, the valid inequalities proposed have a strong influence in reducing the LP gap.

Table 1
Performance of the algorithms on instances with 15 nodes: first and second class of instances (1–48).

Instance	CPU	Number of optima	Average opt. gap	Average error
ABS1-15	19.4	5	0.00	0.27
ABS2-15	78.4	5	0.00	0.33
ABS3-15	10.2	5	0.00	0.57
ABS4-15	868.4	5	0.00	1.18
ABS5-15	2905.2	3	0.32	1.91
ABS6-15	1457.8	4	0.06	1.76
ABS7-15	4	5	0.00	2.10
ABS8-15	61.4	5	0.00	1.87
ABS9-15	117.8	5	0.00	2.77
ABS10-15	5.8	5	0.00	5.57
ABS11-15	8.6	5	0.00	4.03
ABS12-15	2738.4	4	0.27	3.60
ABS13-15	2.6	5	0.00	0.33
ABS14-15	35.2	5	0.00	0.85
ABS15-15	28.4	5	0.00	2.39
ABS16-15	106	5	0.00	0.97
ABS17-15	70.2	5	0.04	1.53
ABS18-15	67.8	5	0.00	2.06
ABS19-15	3.2	5	0.00	1.91
ABS20-15	21.4	5	0.00	2.39
ABS21-15	68.8	5	0.00	3.23
ABS22-15	5.2	5	0.00	4.50
ABS23-15	10	5	0.00	4.08
ABS24-15	1996.2	4	0.21	3.77
Average	445.43	4.79	0.04	2.25
ABS25-15	0.8	5	0.00	0.04
ABS26-15	2.4	5	0.00	0.06
ABS27-15	7.6	5	0.00	0.10
ABS28-15	1.2	5	0.00	0.15
ABS29-15	4.2	5	0.00	0.24
ABS30-15	25.8	5	0.00	0.23
ABS31-15	1	5	0.00	0.29
ABS32-15	6	5	0.00	0.26
ABS33-15	26.2	5	0.00	0.39
ABS34-15	2	5	0.00	0.65
ABS35-15	2.8	5	0.00	0.48
ABS36-15	24.6	5	0.00	0.40
ABS37-15	1.2	5	0.00	0.02
ABS38-15	4.4	5	0.00	0.16
ABS39-15	24.6	5	0.00	0.45
ABS40-15	1.6	5	0.00	0.14
ABS41-15	6.8	5	0.00	0.21
ABS42-15	34.8	5	0.00	0.29
ABS43-15	1.8	5	0.00	0.27
ABS44-15	7	5	0.00	0.36
ABS45-15	45.2	5	0.00	0.50
ABS46-15	3.2	5	0.00	0.54
ABS47-15	3.8	5	0.00	0.50
ABS48-15	33.2	5	0.00	0.45
Average	11.34	5.00	0.00	0.30

Table 2
Performance of the algorithms on instances with 15 nodes: third and fourth class of instances (73–96).

Instance	CPU	Number of optima	Average opt. gap	Average error
ABS49-15	2	5	0.00	1.13
ABS50-15	14.2	5	0.00	3.25
ABS51-15	137	5	0.00	4.33
ABS52-15	2.4	5	0.00	1.48
ABS53-15	11	5	0.00	2.35
ABS54-15	113.2	5	0.00	3.46
ABS55-15	2.2	5	0.00	2.16
ABS56-15	20.4	5	0.00	4.11
ABS57-15	150.6	5	0.00	5.00
ABS58-15	3.2	5	0.00	3.36
ABS59-15	8.2	5	0.00	4.40
ABS60-15	376.4	5	0.00	4.21
ABS61-15	1.6	5	0.00	1.75
ABS62-15	18.6	5	0.00	4.34
ABS63-15	140	5	0.00	6.02
ABS64-15	2.4	5	0.00	1.62
ABS65-15	14	5	0.00	3.55
ABS66-15	193	5	0.00	5.15
ABS67-15	2	5	0.00	2.54
ABS68-15	30.4	5	0.00	4.71
ABS69-15	197.2	5	0.00	5.73
ABS70-15	2.2	5	0.00	3.28
ABS71-15	11.8	5	0.00	4.78
ABS72-15	494.6	5	0.00	4.80
Average	81.19	5.00	0.00	3.65
ABS73-16	4.6	5	0.00	1.59
ABS74-15	6.2	5	0.00	1.72
ABS75-15	3065.6	3	0.53	1.38
ABS76-15	3.8	5	0.00	2.45
ABS77-15	8.2	5	0.00	1.89
ABS78-15	4327.6	2	0.67	1.11
ABS79-15	3.8	5	0.00	1.27
ABS80-15	5.4	5	0.00	3.11
ABS81-15	2123.8	4	0.26	1.56
ABS82-15	6	5	0.00	1.89
ABS83-15	29.2	5	0.00	2.00
ABS84-15	2895.2	3	0.94	1.61
ABS85-15	3.4	5	0.00	0.19
ABS86-15	3	5	0.00	0.20
ABS87-15	28.6	5	0.00	0.11
ABS88-15	2.2	5	0.00	0.27
ABS89-15	3.4	5	0.00	0.21
ABS90-15	22.4	5	0.00	0.06
ABS91-15	2.6	5	0.00	0.17
ABS92-15	4.4	5	0.00	0.41
ABS93-15	41	5	0.00	0.18
ABS94-15	3	5	0.00	0.22
ABS95-15	4.8	5	0.00	0.23
ABS96-15	38	5	0.00	0.08
Average	526.51	4.67	0.10	1.00

Table 3
Performance of the branch-and-cut algorithm.

	% LP gap with (36)–(43)	% LP gap with (41)–(43)	% LP gap
1st class	5.89	22.78	32.68
2nd class	0.72	11.31	12.47
3rd class	28.13	38.23	61.97
4th class	5.86	17.27	18.48

The core of the heuristic is the procedure for the solution of the *single retailer problem*. We tested the effectiveness of the fathoming rules presented in Section 3 to speed-up the heuristic. In Table 4, we show that the large majority of the nodes of the tree are eliminated and only a very small percentage of the leaves are explicitly examined. Tests were carried out on instances with 50 and 100 retailers. Although the *single retailer problem* is NP-hard, the running time of the exact procedure is negligible, always below 1 s, for the tested instances.

Table 4
Performance of the procedure for the solution of the *single retailer problem*.

n	50	100
% Nodes eliminated for infeasibility at facility (26)	9.42	8.63
% Nodes eliminated for infeasibility at retailers (27)	4.91	4.82
% Nodes eliminated for infeasibility at retailers (28)	3.45	3.99
% Nodes eliminated as $q^s_{(t,v_t)\sigma_t} = 0$ (29)	10.01	14.18
% Nodes eliminated for non-optimality (31)	20.67	20.11
% Nodes eliminated for non-optimality (32)	41.31	41.12
% Nodes feasible solutions (leaves of the search tree)	10.23	7.15

6.2. Performance of OU vs ML

The *OU* policy is very well established and represents an easy reference point in practice for the ML policy. It is interesting to compare these policies to see if the savings that can be obtained with the latter policy with respect to the former one are large enough to motivate its use in practice. In the following, we compare the results obtained by algorithm $\mathcal{H}$ with the results given by the heuristic presented by Bertazzi et al. [12] for the OU policy.

In Tables 5–8, we show the errors generated by both policies with respect to the best solution found on instances with 50 and 100 retailers. As in the case with 14 retailers, we generated, for each dimension $n=50$ and 100, five instances for each of the 96 different types, 24 for each of the four classes, for a total of 480 instances. Each table presents the results for a class of instances. In the upper part of each table we provide information on the quality of the solutions. In the three additional parts we provide information on the structure of the solutions. Computational times are expressed in seconds. The total production cost is the sum of the inventory cost at the production facility and the production cost. The total distribution cost is the sum of the inventory cost at the retailers and the transportation cost. The top part of each table reports the average and the maximum error with respect to the best solution found, the number of times the algorithm finds the best solution (we have a total of 120 instances for each of the four classes) and the computational time. The figures in the second, third and fourth part of the table are averages on the 120 instances.

The results show that the ML policy significantly outperforms the OU policy in terms of quality of the solution, on each class of instances. On average over the four classes, the solution value given by the OU policy on the instances with 50 and 100 retailers is, respectively, about 33% and 37% higher than the one given by the ML policy. We can also note that the total quantity delivered to the retailers during the time horizon by applying the OU policy is much larger than the one obtained by applying the ML policy. This is due to the fact that the quantity delivered by applying the OU policy is always such that the maximum level is reached, even when not necessary. Many times the level of the inventory of each retailer at the end of the time horizon is very large. Such level increases when the last delivery time approaches the last time of the horizon. Instead, since the ML policy is more flexible, the quantity delivered in the last delivery time instant is such that the level of the inventory at the end of the time horizon is always 0. Therefore, the total production cost of the OU policy is much bigger than the one of the ML policy. The ML policy gives rise to more visits to the retailers, a smaller inventory cost at the retailer and a larger inventory cost at the facility.

When compared on a short horizon, the OU policy may turn out worse than on a long horizon, because of the impact of the final inventory levels. This is the reason why we performed a different series of experiments. Tables 9–12 show the results obtained by comparing the ML and OU policies on the instances

Table 5
OU vs *ML*: average values: first class of instances (1–24).

n	50		100	
Distribution policy	*ML*	*OU*	*ML*	*OU*
Average percent error	0.00	48.44	0.00	54.02
Maximum error	0.00	86.27	0.00	102.52
Number of best solutions	120	0	120	0
Time	11.30	4.29	188.03	41.77
Delivered quantity	2126.20	3073.60	4009.43	6068.79
Number of visits	90.80	63.16	195.59	123.67
Maximum number of vehicles	4.47	8.00	7.33	14.09
Number of production times	2.00	2.03	2.95	3.15
Inventory cost at prod. fac.	4857.37	3496.82	4705.00	758.38
Production cost	127941.00	178013.00	236691.10	341726.70
Inventory cost at retailers	20388.30	47942.63	38628.61	89842.35
Transportation cost	14692.72	17340.78	27046.85	33415.10
Total production cost	132798.40	181509.80	241396.10	342485.00
Total distribution cost	35081.02	65283.41	65675.46	123257.50

Table 6
OU vs *ML*: average values: second class of instances (25–48).

n	50		100	
Distribution policy	*ML*	*OU*	*ML*	*OU*
Average percent error	0.00	16.14	0.00	21.01
Maximum error	0.00	37.01	0.00	39.46
Number of best solutions	120	0	120	0
Time	12.40	5.12	216.68	100.18
Delivered quantity	2126.20	2460.44	4009.00	4831.26
Number of visits	90.03	64.87	191.42	130.14
Maximum number of vehicles	4.09	8.28	7.05	14.10
Number of production times	1.00	1.00	1.00	1.00
Inventory cost at prod. fac.	22148.28	14006.08	43539.30	20815.25
Production cost	1224411.00	1401725.00	2259951.00	2710091.00
Inventory cost at retailers	21065.99	45154.57	39956.70	91971.93
Transportation cost	14445.12	15836.98	26510.63	29250.78
Total production cost	1246559.00	1415731.00	2303490.00	2730906.00
Total distribution cost	35511.11	60991.55	66467.33	121222.70

Table 7
OU vs *ML*: average values: third class of instances (49–72).

n	50		100	
Distribution policy	*ML*	*OU*	*ML*	*OU*
Average percent error	0.00	40.00	0.00	47.46
Maximum error	0.00	64.55	0.00	73.49
Number of best solutions	120	0	120	0
Time	9.48	3.30	167.84	42.50
Delivered quantity	2126.20	3020.11	4009.00	6013.24
Number of visits	81.07	63.98	168.21	125.20
Maximum number of vehicles	4.10	8.00	6.60	13.68
Number of production times	2.02	2.00	2.90	3.12
Inventory cost at prod. fac.	4574.77	3568.09	3576.76	940.09
Production cost	128032.70	174239.10	236445.00	337886.10
Inventory cost at retailers	24883.24	48971.07	47908.18	93113.27
Transportation cost	67453.08	85704.32	123306.70	166978.50
Total production cost	132607.40	177807.20	240021.80	338826.20
Total distribution cost	92336.32	134675.40	171214.80	260091.70

repeated 10 times, considering in this way 10 periods with different initial level of the inventory of each retailer. In each period successive to the first, just the initial level of the inventory at each retailer is different in each instance with respect to the previous instance. In fact, the initial level is equal to the final level of the previous period. Each table is organized as Tables 5–8. The figures in the second, third and fourth part of the table are averages on the 120 instances and 10 periods.

Table 8
OU vs *ML*: average values: fourth class of instances (73–96).

n	50		100	
Distribution policy	*ML*	*OU*	*ML*	*OU*
Average percent error	0.00	27.87	0.00	25.93
Maximum error	0.00	39.35	0.00	44.88
Number of best solutions	120	0	120	0
Time	10.86	3.28	181.09	60.45
Delivered quantity	2126.20	2774.87	4009.00	5111.14
Number of visits	74.95	68.85	155.35	141.87
Maximum number of vehicles	4.94	9.16	8.28	15.12
Number of production times	1.50	1.48	1.65	1.60
Inventory cost at prod. fac.	6104.85	4901.58	11330.80	9464.14
Production cost	676175.90	852678.70	1246548.00	1559441.00
Inventory cost at retailers	0.00	0.00	0.00	0.00
Transportation cost	13259.70	16543.67	23892.41	30398.38
Total production cost	682280.80	857580.30	1257878.00	1568905.00
Total distribution cost	13259.70	16543.67	23892.41	30398.38

Table 9
OU vs *ML*: average values: first class of instances (1–24)—10 periods.

n	50		100	
Distribution policy	*ML*	*OU*	*ML*	*OU*
Average percent error	0.00	11.51	0.00	14.87
Maximum error	0.00	32.93	0.00	38.93
Number of best solutions	120	0	120	0
Time	17.65	3.20	255.71	52.77
Delivered quantity	4123.30	4154.93	7824.69	7934.45
Number of visits	157.56	72.78	339.16	149.01
Maximum number of vehicles	6.30	11.10	10.45	18.13
Number of production times	2.89	2.15	4.39	2.73
Inventory cost at prod. fac.	5719.45	3017.25	2265.26	3067.94
Production cost	242634.20	240164.90	454297.80	450973.30
Inventory cost at retailers	11594.20	46804.07	20066.34	92349.42
Transportation cost	24769.46	21945.48	46770.94	42890.69
Total production cost	248353.70	243182.20	456563.10	454041.20
Total distribution cost	36363.66	68749.55	66837.28	135240.10

Table 10
OU vs *ML*: average values: second class of instances (25–48)—10 periods.

n	50		100	
Distribution policy	*ML*	*OU*	*ML*	*OU*
Average percent error	0.00	1.40	0.00	2.24
Maximum error	0.00	4.39	0.28	7.40
Number of best solutions	120	0	117	3
Time	20.20	3.65	310.06	70.86
Delivered quantity	4123.30	4155.92	7822.66	7934.74
Number of visits	148.60	73.84	326.55	153.04
Maximum number of vehicles	6.53	14.72	10.86	25.77
Number of production times	1.00	1.00	1.63	1.00
Inventory cost at prod. fac.	38808.67	16259.61	39851.94	28757.96
Production cost	2322815.00	2340805.00	4392111.00	4419223.00
Inventory cost at retailers	13547.15	46875.31	22015.05	93689.24
Transportation cost	24182.24	21911.71	46314.86	42920.70
Total production cost	2361624.00	2357064.00	4431963.00	4447981.00
Total distribution cost	37729.39	68787.02	68329.91	136609.90

The results show that the ML policy still outperforms the OU policy. However, the differences in terms of total cost are much less relevant. On average, the solution value given by the OU policy on the instances with 50 and 100 retailers is, respectively, about 5% and 7% higher than the one given by the ML policy. However, the relative behavior of the two policies strongly depends on the class of instances. It is interesting to observe that the two policies distribute the costs in a quite different way.

Table 11
OU vs *ML*: average values: third class of instances (49–72)—10 periods.

n	50		100	
Distribution policy	*ML*	*OU*	*ML*	*OU*
Average percent error	0.00	6.60	0.00	9.39
Maximum error	0.53	20.57	0.00	27.11
Number of best solutions	119	1	120	0
Time	14.48	2.98	230.44	51.45
Delivered quantity	4123.30	4163.48	7822.66	7963.53
Number of visits	127.00	73.57	267.81	149.91
Maximum number of vehicles	7.68	10.21	12.91	15.98
Number of production times	2.71	2.45	4.13	3.14
Inventory cost at prod. fac.	6300.67	3331.87	3014.53	3005.63
Production cost	241668.40	242121.10	452961.30	454664.10
Inventory cost at retailers	18963.60	47013.67	35412.71	92629.07
Transportation cost	112010.70	108825.90	212949.00	213971.80
Total production cost	247969.00	245452.90	455975.80	457669.70
Total distribution cost	130974.30	155839.60	248361.70	306600.90

Table 12
OU vs *ML*: average values: fourth class of instances (73–96)—10 periods.

n	50		100	
Distribution policy	*ML*	*OU*	*ML*	*OU*
Average percent error	0.13	0.52	0.01	1.49
Maximum error	1.30	1.65	0.33	6.80
Number of best solutions	94	26	116	4
Time	15.19	3.43	240.45	61.49
Delivered quantity	4123.30	4156.13	7822.66	7930.91
Number of visits	102.62	77.50	209.22	167.08
Maximum number of vehicles	9.76	13.19	17.10	22.89
Number of production times	1.60	1.54	1.96	1.80
Inventory cost at prod. fac.	8590.34	8154.87	13412.38	13556.02
Production cost	1280876.00	1290340.00	2401913.00	2433863.00
Inventory cost at retailers	0.00	0.00	0.00	0.00
Transportation cost	21885.54	21992.18	40931.32	43519.81
Total production cost	1289466.00	1298494.00	2415325.00	2447419.00
Total distribution cost	21885.54	21992.18	40931.32	43519.81

7. Conclusions

The focus of this paper is on the analysis of the maximum level (ML) policy, where each retailer can be replenished with any quantity that does not cause the inventory to exceed the maximum level established by the retailer. The ML policy is more flexible than the known order-up to level (OU) policy that forces the quantity delivered to be exactly what is needed to reach the maximum level of the inventory. We have shown that specific situations exist where the OU policy performs very badly with respect to the ML policy. Such result, though relevant, is mostly of theoretical interest and then our successive analysis focused on the algorithms to find optimal and heuristic solutions of the integrated production-distribution system with the ML policy. An exact algorithm has allowed us to verify that the quality of the solutions obtained by the proposed heuristic is high. We then compared the solutions obtained by the ML policy with the ones obtained by the OU policy. The comparison shows that if the testing horizon is relatively short, the ML policy dominates the OU policy. When a sequence of instances is considered, thus creating a long horizon, the cost difference between the two policies decreases. As expected, the ML policy guarantees smaller costs. The savings are achieved both on the production and on the distribution sides of the system.

The OU policy is a natural policy in a traditional management of the production-distribution system in which each retailer applies its own replenishment policy. In an integrated style of management, such as in vendor managed inventory, the OU policy should be replaced by the ML policy, that is a more flexible policy that allows significant savings when the total system cost is optimized.

Acknowledgments

The authors wish to thank an anonymous referee for her/his useful comments and suggestions that helped the authors improve a previous version of this paper.

Appendix A. Proof of Theorem 1

First we note that, in order to guarantee feasibility, the total demand over the time horizon has not to be greater than the total transportation capacity made available over the time horizon, that is $\sum_{i \in M} r_i H \le QH$, as $I_{i0}=0$, $i \in M$. Therefore, $\sum_{i \in M} r_i \le Q$. Consider now the part of the optimal solution corresponding to any two consecutive production time instants, say t' and t''. Then, $y^*_{t'} = \sum_{i \in M} r_i(t''-t')$ and $\sum_{i \in M}\sum_{j=t'}^{t''-1} x^*_{ij} = \sum_{i \in M} r_i(t''-t')$. We show that the values y^* and x^* that satisfy the above conditions are feasible and optimal. The values are feasible, as the transportation capacity is satisfied since $\sum_{i \in M} r_i \le Q$. Moreover, these values are optimal. Suppose that $y^*_{t'} = \sum_{i \in M} r_i(t''-t')+\theta$. If $\theta > 0$, then a better solution (with lower inventory costs and equal production costs) can be obtained by moving the production of θ to t'', since θ is not consumed between t' and t''. If $\theta < 0$, this means that either θ is produced before t' (when $t' > 1$) or the production plan is infeasible (when $t'=1$). In the former case, a better solution (with lower inventory costs and equal production costs) can be obtained by moving the production of θ to time t'. A similar reasoning can be carried out for $\sum_{i \in M}\sum_{j=t'}^{t'-1} x^*_{ij}$. Moreover, the same holds if t' is the last production time instant in the optimal solution, i.e. $t''=H+1$.

Thus, given two consecutive production time instants t' and t'', we know the optimal production quantity at t' and also the total quantity to be distributed between t' and $t''-1$. Therefore, if these two instants are two consecutive production time instants in the optimal solution, the minimum total cost between t' and $t''-1$ can be found by finding the optimal values of the inventory variables B and I and the distribution variables x through the minimization of the total inventory and production costs between t' and $t''-1$. Let $T_{t',t''} = \{t', t'+1, \ldots, t''-1\}$. The minimum cost between t' and $t''-1$, denoted by $c(t',t''-1)$, can be found through the solution of the following linear programming model:

$$\min \sum_{t \in T_{t',t''}} \left(h_0 B_t + K + p\sum_{i \in M} r_i(t''-t') + \sum_{i \in M} h_i I_{it} \right), \tag{44}$$

$$I_{it} = \sum_{j=t'}^{t} x_{ij} - r_i(t-t'), \quad i \in M,\ t \in T_{t',t''} \tag{45}$$

$$I_{it} \le U_i, \quad i \in M,\ t \in T_{t',t''} \tag{46}$$

$$B_t = \sum_{i \in M} r_i(t''-t') - \sum_{i \in M}\sum_{j=t'}^{t} x_{ij}, \quad t \in T_{t',t''} \tag{47}$$

$$\sum_{i \in M} x_{it} \le Q, \quad t \in T_{t',t''} \tag{48}$$

$$I_{it} \ge 0, \quad i \in M,\ t \in T_{t',t''} \tag{49}$$

$$B_t \ge 0, \quad t \in T_{t',t''} \tag{50}$$

$$x_{it} \ge 0, \quad i \in M,\ t \in T_{t',t''} \tag{51}$$

Note that in the objective function $K + p\sum_{i \in M} r_i(t''-t')$ is constant.

The problem *PI-ML* is now reduced to the problem of finding the optimal sequence of production time instants. This corresponds to the problem of determining a shortest path between

Table A1
Characteristics of the instances in each of the first three tested classes.

Instance	h_0	h_i	X_i, Y_i, X_s, Y_s	C
1	3	[6,10]	[0,500]	$2\overline{U}$
2	3	[6,10]	[0,500]	$3/2\overline{U}$
3	3	[6,10]	[0,500]	$\overline{U}$
4	8	[6,10]	[0,500]	$2\overline{U}$
5	8	[6,10]	[0,500]	$3/2\overline{U}$
6	8	[6,10]	[0,500]	$\overline{U}$
7	3	[1,5]	[0,500]	$2\overline{U}$
8	3	[1,5]	[0,500]	$3/2\overline{U}$
9	3	[1,5]	[0,500]	$\overline{U}$
10	8	[1,5]	[0,500]	$2\overline{U}$
11	8	[1,5]	[0,500]	$3/2\overline{U}$
12	8	[1,5]	[0,500]	$\overline{U}$
13	3	[6,10]	[0,1000]	$2\overline{U}$
14	3	[6,10]	[0,1000]	$3/2\overline{U}$
15	3	[6,10]	[0,1000]	$\overline{U}$
16	8	[6,10]	[0,1000]	$2\overline{U}$
17	8	[6,10]	[0,1000]	$3/2\overline{U}$
18	8	[6,10]	[0,1000]	$\overline{U}$
19	3	[1,5]	[0,1000]	$2\overline{U}$
20	3	[1,5]	[0,1000]	$3/2\overline{U}$
21	3	[1,5]	[0,1000]	$\overline{U}$
22	8	[1,5]	[0,1000]	$2\overline{U}$
23	8	[1,5]	[0,1000]	$3/2\overline{U}$
24	8	[1,5]	[0,1000]	$\overline{U}$

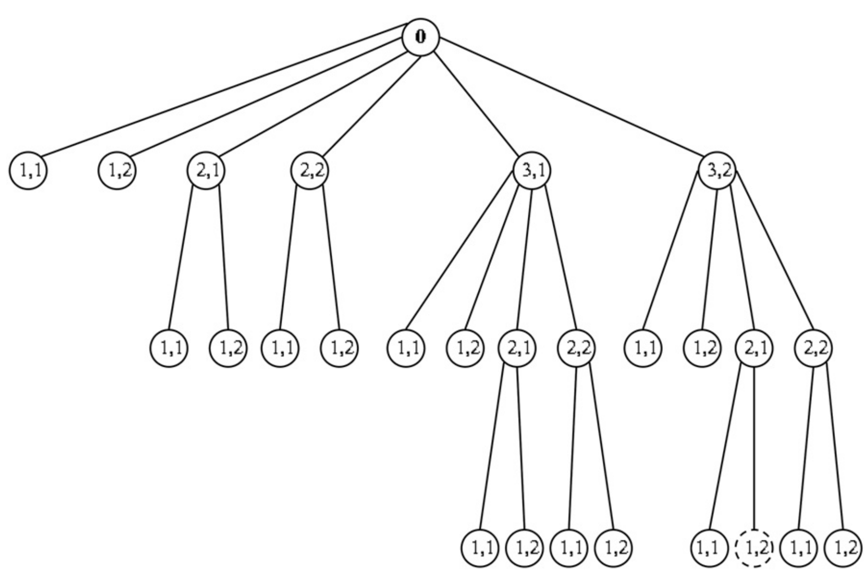

Fig. A1. The search tree for an instance with $H=3$ and $|V|=2$.

node 1 and node $H+1$ in an acyclic network $G(V,A)$ in which the set of nodes V is composed of the set of the possible production time instants $t \in T$ and the time $H+1$, while the set of arcs A is composed of the arcs (t',t'') between each node $t' \in V$ and each node $t'' \in V$ such that $t' < t''$. The cost on each arc (t',t'') is the cost $c(t',t''-1)$.

Since the shortest path on an acyclic network can be found in polynomial time, the problem *PI-ML* can be solved in polynomial time. □

Appendix B. Proof of Theorem 2

The knapsack problem is defined as follows: given n objects of weight $w_i \in Z^+$ and value $v_i \in Z^+$ $(i=1, 2,\ldots,n)$, a capacity $b < \sum_{i=1}^n w_i$ and a positive integer B. Is there a subset A of $\{1, 2,\ldots,n\}$ with $\sum_{i \in A} w_i \le b$ and $\sum_{i \in A} v_i \ge B$?

Archetti et al. [19] proved that the knapsack problem remains NP-hard even if it is assumed that $\frac{2}{3}\sum_{i=1}^n w_i \le b < \sum_{i=1}^n w_i$.

Given an instance of the knapsack problem with n objects of weight w_i and value v_i $(i=1,2,\ldots,n)$ and with capacity b such that $\frac{2}{3}\sum_{i=1}^n w_i \le b < \sum_{i=1}^n w_i$, we construct an instance of the *PI-OU* where $M=\{1,2,\ldots,n\}$, $T=\{1,2\}$, $r_{it}=r$ for $i \in M$ and $t \in T$, starting inventory $I_{i0}=r$ for $i \in M$, maximum level $U_i = w_i - r \ge 2r$ for $i \in M$, inventory cost $h_i = v_i/(U_i - r)$ for $i \in M$, transportation capacity $\sum_{i=1}^n U_i \le Q < \sum_{i=1}^n (U_i + r)$, $K=p=h_0=0$, $Q=b$. Note that since for each i, $w_i - r \ge 2r$ implies that $r \le \frac{1}{3} w_i$ and $U_i = w_i - r$ implies that $U_i \ge w_i - \frac{1}{3} w_i = \frac{2}{3} w_i$, then $\frac{2}{3}\sum_{i=1}^n w_i \le Q < \sum_{i=1}^n w_i$.

In any optimal solution of this instance, each retailer has to be served exactly once during the time horizon. In fact, if retailer i is served in $t=1$, the quantity U_i is received in $t=1$ and 0 in $t=2$. Thus, the inventory level during the time horizon is $I_{i1}=U_i$ and $I_{i2}=U_i-r$ and the total inventory is $2U_i-r$. If retailer i is served in $t=2$, the quantity 0 is received in $t=1$ and U_i+r in $t=2$. The inventory level during the time horizon is $I_{i1}=0$ and $I_{i2}=U_i$ and the total inventory is U_i. Therefore, each retailer prefers to be served in $t=2$. Let us introduce a binary variable z_i that takes value 1 if retailer i is visited in $t=2$, 0 otherwise. The variable z_i corresponds to $\delta(x_{i2})$ defined in Section 2, where x_{i2} takes value 0 when the retailer i is not served at time 2 and U_i+r otherwise. The objective function of the problem *PI-OU* can be equivalently formulated as follows:

$$\min \sum_{i \in M} h_i((2U_i - r)(1 - z_i) + U_i z_i),$$

that is equivalent to

$$\max \sum_{i \in M} h_i (U_i - r) z_i.$$

Since the capacity is sufficient to serve all retailers at time $t=1$, the only constraint needed in the formulation of the problem *PI-OU* is on the limited capacity at time $t=2$. Therefore, the problem *PI-OU* can be equivalently formulated as follows:

$$\max \sum_{i \in M} h_i (U_i - r) z_i, \tag{52}$$

$$\sum_{i \in M} (U_i + r) z_i \le Q, \tag{53}$$

$$z_i \in \{0,1\} \quad i \in M. \tag{54}$$

Since we have defined $h_i = v_i/(U_i - r)$, $U_i = w_i - r$, the knapsack problem has 'yes' answer if and only if the optimal solution of (52)–(54) is greater than or equal to B. □

Appendix C. Proof of Theorem 3

Consider the following instance: time horizon H, number of retailers $n=H$; for each retailer i: quantity consumed $r_{it}=r$ for all $t \in T$, starting level of the inventory $I_{i0}=0$, maximum level of the inventory $U_i=rH$ and inventory cost $h_i = 1/\varepsilon$; at the facility: inventory cost h_0, fixed set-up cost $K = n\varepsilon$, variable production cost $p = \varepsilon$; transportation cost $c_{is} = \varepsilon$ for $i,s \in M'$, transportation capacity $C=r(H+1)$.

Consider the *OU* policy. Since $I_{i0}=0$, then the first delivery to each retailer i has to be performed at time 1. The delivery quantity at time 1 is $U_i+r=r\,(H+1)$. Since no further shipments are needed during the time horizon H, the corresponding level of the inventory at each retailer i is the following: $I_{i1} = rH, I_{i2} = r(H-1), \ldots, I_{iH} = r$. Therefore, the total inventory of each retailer i is $(H+1)Hr/2$. The quantity produced at time 1 is $nr(H+1)$. No further production is needed during the time horizon. Therefore, the corresponding level of the inventory at the facility is equal to 0 for each time instant $t \in T$. The total production cost is $K+pnr(H+1) = (1+r(H+1))n\varepsilon$. A full load is sent separately to each retailer i at time 1. Therefore, the total transportation cost is $\sum_i 2c_{0i}$, that is $2n\varepsilon$. Thus, the optimal cost of the *OU* policy is

$$z^*_{OU} = n\frac{(H+1)Hr}{2\varepsilon} + (r(H+1)+3)n\varepsilon.$$

The optimal cost of the *ML* policy is not greater than the cost of the following feasible solution: $x_{it}=r$, for $i \in M$ and $t \in T$, $y_t=nr$, for $t \in T$, with one vehicle that visits all the retailers at each time $t \in T$. The level of the inventory at each retailer i and at the facility is equal to 0 at each time $t \in T$. The total production cost is $(K+pnr)H$, that is $(1+r)Hn\varepsilon$. The total transportation cost is $(n+1)H\varepsilon$. Thus,

$$z^*_{ML} \le (nr+2n+1)H\varepsilon.$$

Therefore, on this instance:

$$\frac{z^*_{OU}}{z^*_{ML}} \ge \frac{n\frac{(H+1)Hr}{2\varepsilon} + (r(H+1)+3)n\varepsilon}{(nr+2n+1)H\varepsilon} \to \infty \quad \text{for } \varepsilon \to 0. \quad \square$$

Appendix D. Proof of Theorem 4

The optimal quantity $q^s_{(t,v_t)}$ to deliver at time t by using vehicle v_t, with (t, v_t) generated by node (k, v_k) of the partial delivery plan $\sigma_{(k,v_k)}$, is equal to the quantity that satisfies the total demand $\sum_{j=t}^{k-1} r_{sj}$ from t to $k-1$ plus the unmet demand $u_{\sigma_{(k,v_k)}}$, minus I^0_{st-1}, i.e. $q^s_{(t,v_t)} = \max\{\sum_{j=t}^{k-1} r_{sj} + u_{\sigma_{(k,v_k)}} - I^0_{st-1}, 0\}$. However, this quantity may be infeasible, due to the constraints of the problem. More precisely:

1. If the constraint on the maximum level of the inventory is violated, then $q^s_{(t,v_t)} \le U_s + r_{st} - I^0_{st-1}$.
2. If the residual capacity of vehicle v_t used at time t is violated, then $q^s_{(t,v_t)} \le C_{tv_t}$.
3. If the stock-out constraints at the facility are violated, then $q^s_{(t,v_t)} \le \min_{j=t,t+1,\ldots,H} B_{j,\sigma_{(k,v_k)}}$. □

Appendix E. Proof of Theorem 5

Condition (26) is valid as the left-hand side is the quantity that has to be delivered from 1 to $t-1$, given the partial delivery plan $\sigma_{(t,v_t)}$, and the right-hand side is the minimum level of the inventory at the facility from t to H before this quantity is delivered. If (26) holds, any completion of σ_t implies infeasibility at the facility. Condition (27) is valid as, in any completion of the delivery plan $\sigma_{(t,v_t)}$, the inventory level I_{st-1} at time $t-1$ must be at least $u_{\sigma_{(t,v_t)}} > U_s$, that contradicts the constraints (19). Condition (28) obviously holds. Finally, condition (29) holds as $q^s_{(t,v_t)}$ is equal to zero when t is not a delivery time instant for the vehicle v_t, there is no residual transportation capacity or the inventory at the facility is not enough to avoid stock-out (see Theorem 4). $\square$

Appendix F. Proof of Theorem 6

Adding $LBI_{t-1}+LBT_{t-1}$ to $g^s_{(t,v_t)}$, we have a lower bound on the cost of any delivery plan obtained starting from $\sigma_{(t,v_t)}$. Therefore, if this lower bound is greater than or equal to g^*, any completion of $\sigma_{(t,v_t)}$ cannot be optimal. $\square$

Appendix G. Proof of Theorem 7

Since $u^*_t \le u_{\sigma_{(t,v_t)}}$, any delivery plan from 1 to H in which $\sigma_{(t,v_t)}$ is applied from t to H is dominated by a delivery plan from 1 to H equal to the former from 1 to $t-1$ and to σ^*_t from t to H. $\square$

References

[1] Bertazzi L, Savelsbergh M, Speranza MG. Inventory routing. In: Golden B, Raghavan S, Wasil E, editors. The vehicle routing problem, latest advances and new challenges. Springer; 2008. p. 49–72.
[2] Campbell AM, Clarke L, Kleywegt AJ, Savelsbergh MWP. The inventory routing problem. In: Crainic TG, Laporte G, editors. Fleet management and logistics. Boston: Kluwer Academic Publishers; 1998. p. 95–113.
[3] Cordeau J-F, Laporte G, Savelsbergh MWP, Vigo D. Short-haul routing. In: Laporte G, Barnhart C, editors. Handbooks in operations research and management science: transportation, vol. 14. Amsterdam: Elsevier; 2007. p. 367–428.
[4] Federgruen A, Simchi-Levi D. Analysis of vehicle routing and inventory-routing problems. In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL, editors. Handbooks in operations research and management science, vol. 8. North-Holland; 1995. p. 297–373.
[5] Cohen MA, Lee HL. Strategic analysis of integrated production-distribution systems: models and methods. Operations Research 1988;35:216–28.
[6] Thomas DJ, Griffin PM. Coordinated supply chain management. European Journal of Operational Research 1996;94:1–15.
[7] Blumenfeld DE, Burns LD, Diltz JD, Daganzo CF. Analyzing trade-offs between transportation, inventory and production costs on freight networks. Transportation Research 1985;19B:361–80.
[8] Sarmiento AM, Nagi R. A review of integrated analysis of production-distribution systems. IIE Transactions 1999;31:1061–74.
[9] Erengüç ŞS, Simpson NC, Vakharia AJ. Integrated production/distribution planning in supply chains: an invited review. European Journal of Operational Research 1999;115:219–36.
[10] Chandra P, Fisher ML. Coordination of production and distribution planning. European Journal of Operational Research 1994;72:503–17.
[11] Bertazzi L, Paletta G, Speranza MG. Deterministic order-up-to level policies in an inventory routing problem. Transportation Science 2002;36:119–32.
[12] Bertazzi L, Paletta G, Speranza MG. Minimizing the total cost in an integrated vendor-managed inventory system. Journal of Heuristics 2005;11:393–419.
[13] Baker KR, Dixon P, Magazine MJ, Silver EA. An algorithm for the dynamic lot-size problem with time-varying production capacity constraints. Management Science 1978;24:1710–20.
[14] Toth P, Vigo D. The vehicle routing problem. SIAM monographs on discrete mathematics and applications, Philadelphia; 2002.
[15] Wagner HM, Whitin TM. Dynamic version of the economic lot size model. Management Science 1958;5:89–96.
[16] Florian M, Lenstra JK, Rinnooy Kan HG. Deterministic production planning: algorithms and complexity. Management Science 1980;26:669–79.
[17] Fischetti M, Salazar-González JJ, Toth P. Solving the orienteering problem through branch and cut. INFORMS Journal of Computing 1998;10:133–48.
[18] Gendreau M, Laporte G, Semet F. A branch-and-cut algorithm for the undirected selective traveling salesman problem. Networks 1998;32:263–73.
[19] Archetti C, Bertazzi L, Hertz A, Speranza MG. A hybrid heuristic for an inventory-routing problem. INFORMS Journal on Computing, to appear.
[20] Padberg M, Rinaldi G. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Review 1991;33:60–100.
[21] Archetti C, Bertazzi L, Laporte G, Speranza MG. A branch-and-cut algorithm for a vendor managed inventory routing problem. Transportation Science 2007;41:382–91.