Int. J. Production Economics 146 (2013) 359–370

Contents lists available at ScienceDirect

Int. J. Production Economics

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

Hybrid heuristic for the inventory location-routing problem with deterministic demand

W.J. Guerrero [a,b,*], C. Prodhon [a], N. Velasco [b,c], C.A. Amaya [b]

[a] ICD-LOSI, UMR CNRS STMR, Université de Technologie de Troyes, 12 Rue Marie Curie, CS 42060, 10004 Troyes Cedex, France
[b] Departamento de Ingeniería Industrial, School of Engineering, Universidad de los Andes, Cra 1 Este No 19A - 40, Bogotá, Colombia
[c] ICD-LOSI, UMR CNRS, Université de Technologie de Troyes, France

ARTICLE INFO

Article history:
Received 21 June 2012
Accepted 31 July 2013
Available online 9 August 2013

Keywords:
Location-routing problem
Inventory-routing problem
Matheuristics
Vehicle routing problem

ABSTRACT

The Inventory Location-Routing Problem with deterministic demand can be seen as an approach to both optimize a supply chain design and minimize its operational costs. This problem considers that vehicles might deliver products to more than one retailer per route and that inventory management decisions are included for a multi-depot, multi-retailer system with storage capacity over a discrete time planning horizon. The problem is to determine a set of candidate depots to open, the quantities to ship from suppliers to depots and from depots to retailers per period, and the sequence in which retailers are replenished by an homogeneous fleet of vehicles. A mixed-integer linear programming model is proposed to describe the problem and to provide bounds on the solutions. It is strengthened by two sets of valid inequalities with an analysis of their impact. Since the model is not able to solve the targeted instances exactly within a reasonable computation time, a hybrid method, embedding an exact approach within a heuristic scheme, is presented. Its performance is tested over three sets of instances for the inventory location routing, location-routing and inventory-routing problems. Results show important savings achieved when compared to a decomposed approach and the capability of the algorithm to solve the problem.
© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The design of a supply chain is considered as a strategic level decision. It consists of identifying the optimal number of plants to open and their locations so that logistics costs are minimal. On the other hand, the management of a supply chain is usually to tackle tactical and/or operational decisions and it concerns the cooperation between facilities in order to obtain, transform, store and distribute materials, which also entails logistical costs (Melo et al., 2009). Balancing strategic with operational objectives is the challenge.

Most of the facility location models consider distribution to be performed by dedicated routes, i.e., one vehicle visits one client at most (see Gebennini et al., 2009). However, in the case where orders are much smaller than vehicle capacity, this assumption is not longer true. The effects of ignoring routing decisions when locating depots are studied by Shen and Qi (2007) and Salhi and Rand (1989). When vehicles are not performing single-visit tours, locating depots, so that the sum of the distances between depots and retailers is minimized, is not an optimal solution. A more appropriate model is the one depicted in Location-Routing problems, that propose to optimize location decisions simultaneously with routing decisions. Examples are described by Prins et al. (2007), Belenguer et al. (2011) and a review is presented by Nagy and Salhi (2007). Nevertheless, these papers deal with the single period version or they simplify the multi-period problem by weighting the service to customers to be the same on each period of the horizon. Recently, Prodhon (2011) solved a periodic version, but no inventory decision is managed.

Then, Miranda and Garrido (2009) discuss the impact of ignoring inventory decisions when designing a supply chain. They conclude that the assignment scheme of retailers to depots has a direct impact on depot operation cost because ordering and holding costs might be significantly modified when the aggregated demand varies.

In addition, inventory and routing decisions are strongly interdependent (Bell et al., 1983). Distribution and stock management decisions affect each other for two reasons: First, the set of minimal cost routes is built as a function of the quantities to deliver per period, which are determined by the inventory policies; and second, ordering costs required to design inventory policies include, among others, the transportation cost resulting from the choice of the sequence in which the retailers will be served. The optimal trade-off between inventory and distribution costs is known as the Inventory-Routing Problem (IRP) in Bertazzi et al. (2002), and Andersson et al. (2010).

Designing a supply chain becomes more complex if inventory and routing problems are included in the location decision-making. However, it is essential to balance short-term decisions with longer

* Corresponding author at: Universidad de los Andes Cra 1 Este # 19A-40, Bogotá, Colombia. Tel.: +571 3394949x3833
E-mail addresses: wi-guerr@uniandes.edu.co (W.J. Guerrero), caroline.prodhon@utt.fr (C. Prodhon), nvelasco@uniandes.edu.co (N. Velasco), ca.amaya@uniandes.edu.co (C.A. Amaya).

0925-5273/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2013.07.025

term thinking. As a result, for the Inventory Location-Routing Problem (ILRP), the resulting supply chain design includes an insight into detailed topics in order to decide how to satisfy future demand at minimum cost. Interest arises mainly from two contexts:

(i) When a temporary location is required. It is the case for companies that strategically lease depots and pay rent. Consequently, they are more flexible and might conveniently change locations periodically. It is also the case for humanitarian missions managing disaster relief inventories (Whybark, 2007; Balcik et al., 2010) with limited financial resources through donations. These activities are often performed for a short time. Further, in the field of military logistics, temporary location decisions are often made in order to distribute ammunition and other supplies. In all cases, location costs (e.g., rent) and operational costs (distribution and inventory holding) could have similar orders of magnitude.
(ii) When long-term objectives require a supply chain design allowing different frequencies of replenishment for each retailer and distribution to be performed by vehicles capable of visiting more than one retailer per route. It is the case when assuming single period routing decisions (assuming routing to be the same every period) or dedicated routes (routes visiting a single retailer) are not realistic enough. The large retail sector or pharmaceutical and medical equipment supply are some examples. Again, depot opening costs should be scaled on the modeled horizon to be in balance with the operational costs. Furthermore, even if the future demand is not considered in the long-term, including inventory and routing costs allows incorporating within the location–allocation structure the effects of non-constant distribution activities and the effects of the interactions between inventory and routing decisions. Then, location decisions based on a set of routing scenarios (one per period) will perform outstandingly better on the long-run than one based on a single routing scenario.

Note that these applications suggest that demand might have an unpredictable nature while our model assumes known data. Our contribution is to solve the deterministic version of the problem in order to take the first step before solving a stochastic version with recourse. Even more, we also develop a decision-aid tool for "what-if" analysis. Think that an analyst might be interested in having better estimates of costs given the possibility of restructuring the supply chain under specific future demand assumptions. Few papers simultaneously work on the three problems: depot location, vehicle routing and optimizing inventory policies. Table 1 summarizes a literature review on models and solution methods for the ILRP. Columns **Ret.** and **Dep.** denote if inventory decisions are made either at retailers, at depots, or both.

Most consider a single period routing, location decisions within a discrete set, demand splitting or backlogging not allowed and stochastic demand. The cost structure to be minimized comprises fixed opening costs for depots, expected holding and stock-out costs, and routing costs. Considering deterministic demand, Ambrosino and Scutellà (2005) propose a linear model for the ILRP and show that for the single period case (LRP), the model implemented in CPLEX 7.0 is not able to find optimal solutions within 25 h for instances with 13 depots and 95 retailers. For stochastic demand, Ma and Davidrajuh (2005) propose an iterative sequential optimization approach where the problem is tackled as a series of sub-problems and never with a global perspective.

In addition, two different characterizations of this problem exist. First, some research papers tackle a LRP integrating in the objective function an EOQ-like component (Wilson model) aiming to minimize the expected inventory management cost at retailers, resulting in a non-linear model. The second approach fixes quantities to be delivered to retailers and optimizes inventory policies at depots instead.

This paper studies the ILRP as the issue of locating depots considering depot fixed opening costs, operational and tactical costs such as routing and stock management cost are included. The mathematical model and some valid inequalities are presented in Section 2. Section 3 describes a hybrid heuristic and a computational study is presented in Section 4. Conclusions are given in Section 5.

2. Problem definition

This paper tackles the design of a two-echelon supply chain considering both strategic and tactical/operational costs. This design comprises the location of the depots supplied by a factory and serving the deterministic demand of retailers, and the assignment of the latter to a depot over a given horizon. Each retailer is assigned to a single depot in the interest of facilitating monitoring and tracing of products. The costs include the depot opening costs, the delivery costs (dedicated routes to depots, non-dedicated to retailers) and the inventory costs at both depots and retailers, including an obsolescence penalty cost (that could be 0 or positive).

Formally, let J be a set of n retailers facing a deterministic non-constant demand $d_{jt}, \forall j \in J, \forall t \in H$, with t a period and $H = \{1, \ldots, p\}$ a discrete and finite planning horizon. Also, a set of m candidate depots I is available to replenish retailers. The ILRP is defined on a complete, weighted and directed graph $G = (V, A, C)$. $V = \{J \cup I\}$ is the set of nodes in the graph. C is the cost matrix c_{ij} associated with the traveling cost from node i to node j in the set of arcs A in the network. We consider a homogeneous unlimited fleet of vehicles, thus a set K of r $(r \geq n)$ identical vehicles are available. Each node $i \in V$ is associated with a storage capacity W_i. Also, each depot $j \in I$ is associated to an opening cost O_j and ordering cost s_i (dedicated route from the factory or production cost). The vehicle capacity is Q units of product and the fixed cost of using a vehicle at least once in the planning horizon is F. Let B_i be the initial inventory at facility $i \in V$. $H_0 = \{0\} \cup H$ and $H' = H \cup \{p+1\}$ are horizons including a dummy period used to

Table 1
A classification on combined inventory-location-routing problems and methods.

Authors	Demand	Ret.	Dep.	Model	Solution method
Liu and Lee (2003)	Stochastic	✓		Non-linear	Route first-locate second
Liu and Lin (2005)	Stochastic	✓		Non-linear	Sequential/improv. stage
Ambrosino and Scutellà (2005)	Determ.	✓	✓	Linear	Commercial solver
Ma and Davidrajuh (2005)	Stochastic	✓	✓	Non-linear	Sequential
Shen and Qi (2007)	Stochastic		✓	Non-linear	Branch-and-bound
Javid and Azad (2010)	Stochastic		✓	Non-linear	Tabu search Simulated Annealing
Mete and Zabinsky, 2010	Stochastic		✓	Non-linear	Sequential stochastic programming
Sajjadi and Cheraghi (2011)	Stochastic	✓		Non-linear	Sequential/improv. stage

model initial and final conditions in inventory levels. The holding plus obsolescence penalty cost for one unit of product kept at node $j \in V$ from period $t \in H_0$ until period $l \in H'$ is q_{jtl}. Backlogging or stock-out are not allowed.

Let the decision variables be $y_i = 1$ iff depot $i \in I$ is opened; $f_{ij} = 1$ iff retailer $j \in J$ is assigned to depot $i \in I$, $x_{ijkt} = 1$ iff the arc $(i,j) \in A$ is crossed from i to j by vehicle $k \in K$ on period $t \in H$, T_i be the maximum number of vehicles used from depot $i \in I$ over H. Inventory decisions at echelon e are denoted by the variable w^e. The quantity replenished from depot i to retailer j in period t to satisfy the demand in period l using the vehicle k is denoted by w^2_{ijtlk} (the superscript 2 denotes inventory decisions for the second echelon). The quantity of product used from initial stock at retailer j to satisfy demand in period $t \in H'$ is denoted by w^2_{j0t}. At the first echelon, $z_{li} = 1$ iff depot $i \in I$ is replenished in period $l \in H$, 0 otherwise. The quantity to replenish in depot $i \in I$ that is delivered in period $t \in H$ to satisfy the demand in period $l \in H'$ is w^1_{itl} (the superscript 1 denotes the first echelon). Then, the ILRP model can be stated as follows:

$$\min \sum_{i \in I}\left(O_i y_i + FT_i + \sum_{l \in H} s_i z_{li}\right) + \sum_{i \in I}\sum_{t \in H_0}\sum_{l=t|l>0}^{p+1} q_{itl} w^1_{itl} + \sum_{t \in H'} q_{j0t} w^2_{j0t} + \sum_{i \in I}\sum_{j \in J}\sum_{t \in H}\sum_{l=t}^{p+1}\sum_{k \in K} q_{jtl} w^2_{ijtlk} + \sum_{i \in V}\sum_{j \in V}\sum_{k \in K}\sum_{t \in H} c_{ij} x_{ijkt} \quad (1)$$

Subject to :
$$\sum_{i \in I}\sum_{k \in K}\sum_{l=1}^{t} w^2_{ijltk} + w^2_{j0t} = d_{jt} \quad \forall j \in J, \ \forall t \in H \quad (2)$$

$$\sum_{l=0}^{t} w^1_{ilt} = \sum_{k \in K}\sum_{l=t}^{p+1}\sum_{j \in J} w^2_{ijtlk} \quad \forall i \in I, \ \forall t \in H \quad (3)$$

$$\sum_{t \in H'} w^2_{j0t} = B_j, \quad \forall j \in J \quad (4)$$

$$\sum_{t \in H'} w^1_{i0t} = B_i \cdot y_i, \quad \forall i \in I \quad (5)$$

$$\sum_{r=0}^{t}\sum_{l=t}^{p+1} w^1_{irl} \le W_i \cdot y_i \quad \forall i \in I, \ \forall t \in H \quad (6)$$

$$\sum_{l=t}^{p+1}\left(w^2_{j0l} + \sum_{r=1}^{t}\sum_{k \in K} w^2_{ijrlk}\right) \le W_j, \quad \forall i \in I, \ \forall j \in J, \ \forall t \in H \quad (7)$$

$$\sum_{l=t}^{p+1} w^1_{itl} \le W_i \cdot z_{ti} \quad \forall i \in I, \ \forall t \in H \quad (8)$$

$$\sum_{i \in I} f_{ij} = 1, \quad \forall j \in J \quad (9)$$

$$f_{ij} \le y_i, \quad \forall j \in J, \ \forall i \in I \quad (10)$$

$$\min(Q, W_j) \cdot \sum_{u \in J} x_{iukt} \ge \sum_{l=t}^{p+1} w^2_{ijtlk} \quad \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall k \in K \quad (11)$$

$$\min(Q, W_j) \cdot \sum_{u \in J \cup \{i\}} x_{ujkt} \ge \sum_{l=t}^{p+1} w^2_{ijtlk} \quad \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall k \in K \quad (12)$$

$$\sum_{j \in V} x_{ijkt} - \sum_{j \in V} x_{jikt} = 0, \quad \forall t \in H, \ \forall i \in V, \ \forall k \in K \quad (13)$$

$$\sum_{i \in V}\sum_{k \in K} x_{ijkt} \le 1 \quad \forall t \in H, \ \forall j \in V \quad (14)$$

$$\sum_{i \in V}\sum_{k \in K} x_{jikt} \le 1 \quad \forall \ t \in H, \ \forall j \in V \quad (15)$$

$$\sum_{i \in I}\sum_{j \in J} x_{ijkt} \le 1 \quad \forall t \in H, \ \forall k \in K \quad (16)$$

$$\sum_{k \in K}\sum_{j \in J} x_{ijkt} \le T_i \quad \forall t \in H, \ \forall i \in I \quad (17)$$

$$\sum_{i \in I}\sum_{l=t}^{p+1}\sum_{j \in J} w^2_{ijtlk} \le Q \quad \forall k \in K, \ \forall t \in H \quad (18)$$

$$\sum_{u \in J} x_{iukt} + \sum_{u \in V \setminus \{j\}} x_{ujkt} \le 1 + f_{ij} \quad \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall k \in K \quad (19)$$

$$\sum_{i \in S}\sum_{j \in S} x_{ijkt} \le |S| - 1 \quad \forall t \in H, \ \forall k \in K, \ \forall S \subseteq J \quad (20)$$

$$x_{ijkt} \in \{0,1\} \quad \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall k \in K \quad (21)$$

$$y_i \in \{0,1\} \quad \forall i \in I \quad (22)$$

$$z_{it} \in \{0,1\} \quad \forall i \in I, \ \forall t \in H \quad (23)$$

$$f_{ij} \in \{0,1\} \quad \forall i \in I, \ \forall j \in J \quad (24)$$

$$T_i \in \mathbb{N} \quad \forall i \in I \quad (25)$$

$$w^2_{ijtlk} \in \mathbb{R}^+ \quad \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall l \in H'|l \ge t, \ \forall k \in K \quad (26)$$

$$w^2_{j0t} \in \mathbb{R}^+ \quad \forall j \in J, \ \forall t \in H' \quad (27)$$

$$w^1_{itl} \in \mathbb{R}^+ \quad \forall i \in I, \ \forall t \in H_0, \ \forall l \in H|l \ge t \quad (28)$$

The objective function equation (1) is the sum of the opening costs, ordering costs at depots, the costs of using a vehicle at least once, holding costs at depots and retailers with the distribution costs. Constraints (2) force the satisfaction of the demand at every retailer. Inventory flow conservation through echelons is forced by constraints (3). The sum over the horizon of the quantity kept on stock from period zero up to period p+1 is equal to the initial stock as stated by constraints (4) and (5). Capacity of depots is guaranteed by (6) meaning that satellite depots are not cross-docking points. Retailers are capacitated as shown in Eq. (7). Ordering decisions at depots are forced by constraint set (8). Each retailer must be allocated to a single opened depot as stated by Eqs. (9) and (10). Constraints (11) and (12) guarantee that if a retailer is replenished on period l with route k, it must be visited accordingly. If the triangle inequality is not guaranteed, then visits without replenishment should be forbidden by adding the constraints (29):

$$\sum_{i \in V} x_{ijkt} \le \sum_{i \in I}\sum_{l=t}^{p+1} w^2_{ijtlk} \quad \forall j \in J, \ \forall t \in H, \ \forall k \in K \quad (29)$$

Concerning distribution, constraints (13)–(20) force a feasible routing. First, traditional vehicle flow conservation constraints are presented in equations (13)–(15). Note that equations (16) force each vehicle to perform one route per period at the most. This is a common assumption in most vehicle routing problems. In real-life, vehicles could perform several trips per day; this problem is known as the multi-trip vehicle routing problem (Brandão and Mercer, 1997). In that case, duration constraints are included to limit the time each vehicle works per day. A more sophisticated approach than the presented, where vehicles are allowed to perform multi-trips in order to reduce the fleet size cost, such that the duration of the tasks scheduled to vehicles remains feasible, is future research.Further, equations (17) link the cost of using vehicles with the routing decisions. Vehicles limited capacity is forced by equations (18). The set of equations (19) state that a retailer j can be linked to a depot i only if j is assigned to depot i (f_{ij}=1). Finally, equations (20) are standard subtour elimination constraints and constraints (21)–(28) state the nature of the decision variables.

This model can be enforced by additional inequalities. However, the ones for the traveling salesman problem (TSP) or vehicle routing problems (VRP) are not valid in this case because quantities to deliver and retailers to visit are decision variables. Inequalities presented for the IRP are not valid either since: (i) the holding cost is considered to be time-dependent to include seasonal effects and obsolescence penalty costs, (ii) there is a multi-depot environment, (iii) the depots do not have a fixed location. Nonetheless, two valid inequalities for the ILRP are presented.

Theorem 1. *The inequalities (30) are valid for the ILRP with deterministic demand.*

$$min(W_i, Q) \cdot \sum_{k \in K} \sum_{j \in J} x_{ijkt} \geq w^1_{ilt} \quad \forall i \in I, \ \forall l \in H_0, \ \forall t \in H | l \leq t \tag{30}$$

Proof. If a depot i is replenished at any period l to satisfy demand on period t —i.e., $w^1_{ilt} > 0$ - then:

- At least w^1_{ilt}/Q vehicles must depart from depot i in period t to satisfy such a demand - i.e., $\sum_{k \in K} \sum_{j \in J} x_{ijkt} \geq w^1_{ilt}/Q$, if the capacity of the vehicle is tight (i.e., $\min(W_i, Q) = Q$)
- At least one vehicle must depart from depot i in period t to satisfy such a demand – i.e., $\sum_{k \in K} \sum_{j \in J} x_{ijkt} \geq w^1_{ilt}/W_i$, if the vehicle capacity constraint is loose (i.e., $\min(W_i, Q) = W_i$). □

Theorem 2. *The inequalities (31) are valid for the ILRP with deterministic demand.*

$$\sum_{i \in V} \sum_{k \in K} \sum_{l=1}^{t} x_{ijkl} \geq \left\lfloor \frac{\sum_{l=1}^{t} d_{jl} - B_j}{Q} \right\rfloor, \quad \forall \ j \in J, \ \forall \ t \in H \tag{31}$$

Proof. The set of constraints (31) require that the minimal number of times a retailer is visited up to period t equals the total demand that can not be satisfied with the initial inventory $(\sum_{l=1}^{t} d_{jl} - B_j)$ divided by the vehicle capacity. □

3. Hybrid heuristic

Exact procedures can only solve the model for very small sized instances within a reasonable computation time (as will be shown in Section 5). Thus, heuristic methods seem to be a more suitable alternative to find high quality solutions on larger instances. The proposed framework is based on this kind of approach and tries to solve subproblems, not considering several decision levels in independent phases, but exchanging information when moving between solution spaces.

It should be emphasized that most of the previous ILRP models consider an inventory policy (mainly EOQ) with ordering costs that are modeled independently from distribution, while in real life it depends on the routing performed by vehicles. In addition, the embedded Supply Chain Design Problem (SCDP) with estimated distribution costs (neglecting the routing construction) might be solved to optimality using commercial solvers in reasonable computation time. Taking advantage of this last property, the problem is decomposed into decisions that are computed by exact methods and the ones obtained heuristically. Then, the suggested pattern makes exact and heuristic procedures cooperate and this leads to a hybrid heuristic that can be seen as a matheuristic (Raidl and Puchinger, 2008). Thus, the ILRP resolution induces:

- A supply chain design S fully described by three elements: (i) the set I of the depots to be opened, (ii) F the array indicating, for each retailer, its assigned depot, (iii) W the $(m+n) \times (p+1) \times (p+1)$ matrix in which each element w_{itl} indicates, for each facility $i \in V$, the quantity of product arriving in period $t \in H_0$ that will remain in stock until period $l \in H'$.
- A routing evaluation $R_{(S)}$ as the set of routes indicating the sequence in which retailers will be replenished on each period for a given supply chain structure S.

Note that a single structure S might have several feasible sets of routes $R_{(S)}$. In the following subsections, the main components of the approach will be described in order to subsequently assemble the proposed algorithm.

3.1. Supply chain design

Assume the $m \times n \times p$ matrix C^* to be known in which each element c^*_{ijt} represents the cost of delivering product from depot $i \in I$ to retailer $j \in J$ in period $t \in H$. It is an estimated sum of distribution cost and ordering costs from depots to retailers. Then, the MIP presented in (2) could be modified to obtain a SCDP.

Decision variables x_{ijkt} would be replaced by $\hat{x}_{ijt}, \ \forall i \in I, \forall j \in J, \forall t \in H$ representing a binary variable equal to 1 iff depot i replenishes the retailer j in period t, 0 otherwise. w^2_{ijtlk} is replaced by $\hat{w}^2_{ijtl} \forall i \in I, \ \forall j \in J, \ \forall t \in H, \ \forall l \in H'$ representing the quantity replenished by depot i in period t to stock until period l at retailer j. Accordingly, the objective function (1) would be replaced by

$$\min \ \sum_{i \in I} \left(O_i y_i + FT_i + \sum_{l \in H} s_i z_{li} \right) + \sum_{i \in I} \sum_{l \in Ho} \sum_{t \in H'} q_{jtl} w^1_{itl} + \sum_{t \in H'} q_{j0t} w^2_{jot}$$
$$+ \sum_{i \in I} \sum_{j \in J} \sum_{t \in H} \sum_{l \in H'} q_{jtl} \hat{w}^2_{ijtl} + \sum_{i \in I} \sum_{j \in J} \sum_{t \in H} c^*_{ijt} \hat{x}_{ijt} \tag{32}$$

The index k is easily removed from constraints (2), (3), (7), (18), and (26). Eqs. (4)–(6), (8)–(10), (21)–(25), (27), and (28) remain unchanged. The following constraints are added to complete the SCDP formulation.

$$\min(Q, W_j) \cdot \hat{x}_{ijt} \geq \sum_{l=t}^{p+1} \hat{w}^2_{ijtl} \quad \forall i \in I, \ \ \forall j \in J, \ \ \forall t \in H \tag{33}$$

$$f_{ij} \geq \hat{x}_{ijt} \quad \forall \ i \in I, \quad t \in H, \ \ \forall j \in J \tag{34}$$

$$\sum_{j \in J} \sum_{l=t}^{p+1} \hat{w}^2_{ijtl} \leq T_i \cdot Q \quad \forall t \in H, \ \ \forall i \in I \tag{35}$$

Eq. (33) relates distribution activities to inventory flow from depot i to retailer j on period t. Eq. (34) forbids replenishment from i to j if retailer j is not allocated to depot i. Constraint (35) states that the minimum number of vehicles T_i to use from depot i times the vehicle capacity Q is larger than the total quantity to replenish. To sum up, S is a partial solution for the ILRP and it might be computed to optimality by solving the presented MIP model. This supply chain design generator will be denoted by the acronym SCDP for simplicity.

Since the distribution cost depends on the embedded routing which can only be solved once the quantities to deliver per period are known, the initial matrix C^* is estimated to be a random fraction of the direct delivery cost. Then, each element $c^*_{ijt} = \xi_1 \cdot c_{ij} \ \forall i \in I, \ \forall j \in J, \ \forall t \in H$, where ξ_1 is a uniform random variable $\sim Unif[\alpha, \ 2 \cdot \alpha]$. The parameter α is fixed a priori.

C^* is updated every time feasible routing costs are computed. This update is performed through Eq. (36) with the information of a feasible ILRP solution and represents the cost of detour of the route. If no replenishment is made from depot i to retailer j on period t $(\sum_{l \in H} \hat{w}^2_{ijtl} = 0)$, c^*_{ijt} remains unchanged. In this context, the estimation of C^* is modified every time feasible routes are computed (by the routing operators). By doing so, it is expected C^* to be a better input for the supply chain design generator (SCDP solver). This is how the routing operator will cooperate with subsequent supply chain design generations.

$$c^*_{ijt} = \sum_{u \in V} \sum_{k \in K} (c_{uj} x_{ujkt} + c_{ju} x_{jukt}) - \sum_{u \in V} \sum_{v \in V} \sum_{k \in K} c_{uv} x_{ujkt} x_{jvkt}, \quad \text{if } \sum_{l \in H} \hat{w}^2_{ijtl} > 0 \tag{36}$$

For example, consider a feasible solution where at period t, the arcs (u, j) and (j, v) are traversed by a vehicle that departed from depot i. In this particular example, the updated cost for c^*_{ijt} is equal to the cost of making a detour to visit retailer j, which is $c_{uj}+c_{ju}-c_{uv}$. In this context, the estimation of C^* is modified every time feasible routes are computed (by the routing operators). By doing so, it is expected C^* to be a better input for the supply chain design generator (SCDP solver). This is how the routing operator will cooperate with subsequent supply chain design generations.

3.2. Randomized routing heuristic

Even when the supply chain S is designed, the remaining routing decisions $R_{(S)}$ are difficult to solve since the problem reduces to the well known VRP for every period and every depot. Extensive research is being proposed to solve this NP-hard problem (Toth and Vigo, 2001; Golden et al., 2008).

In addition, to reinforce the search, it is better to consider the presolved assignment as a means to provide a subset of promising depots to open and then to tackle a multi-depot VRP per period, and even an LRP. A simple heuristic procedure that provides good solutions for the LRP is the RECWA, the randomized extended Clarke and Wright algorithm, implemented as in Prins et al. (2006). It is an extension for the multi-depot case of the Clarke and Wright saving's algorithm (Clarke and Wright, 1964) (see Mendoza, 2009 for results on the single-depot version). A randomization on the selected merge allows some diversification over the iterations. However, retailers allocation to depots must be the same along the horizon, which is not warranted by solving an LRP per period and requires a repairing operator.

The repairing operator evaluates for each retailer, the assigned depots during the planning horizon. Retailers are evaluated in decreasing order of their total demand. The most frequent depot allocation is fixed for each retailer if capacity constraints hold. A randomized Clarke and Wright algorithm is then performed for each depot, for each period while fixing the allocation decisions.

3.3. Local search

To improve the routing and the inventory on the global solution, a local search (LS) is used. The hierarchy and description of the neighborhoods explored in our LS are:

- *Move*: The visit of a retailer is shifted from its current position to a different position within the same route or to other routes departing from the same depot at the same period.
- *Swap*: The positions of two different retailers are exchanged. Both exchanged visits must share depot and period, and might or might not be in the same route.
- *2-Opt*: Two non-consecutive arcs are removed from the solution and new arcs are included to assure feasibility of the solution. The removed arcs might or might not belong to the same route but they must share the same depot and period.
- *Shift delivery date*: A single retailer is removed from the solution and new inventory policies are designed. Its first delivery date is shifted to the earliest possible date. On the shifted first delivery date, the retailer will be replenished with its maximum storage capacity or the maximum available

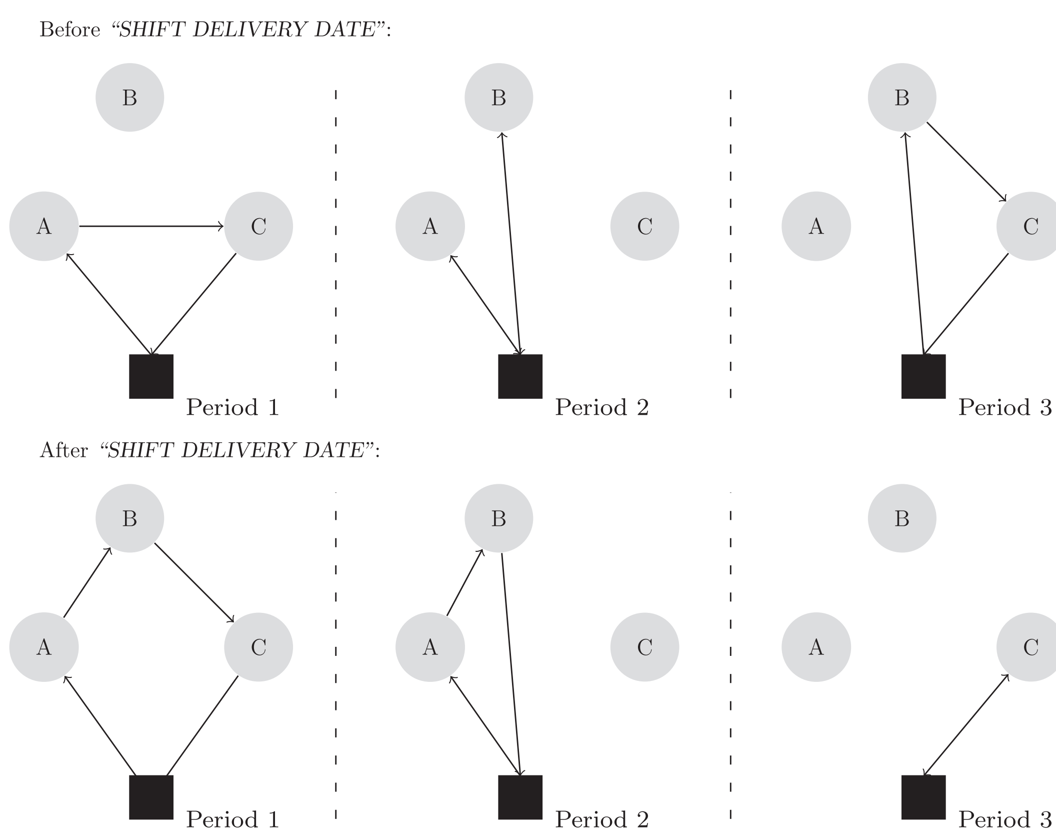

Fig. 1. *Shift delivery date* example.

capacity to minimize any chances of stockout. Replenishment in subsequent periods is decided analogously if stock is not sufficient to satisfy future demands. The retailer with the modified inventory policy is re-inserted in the solution and is allocated to the same depot as before. Routing costs are evaluated by best insertion procedure. The new inventory policy for the retailer also affects the inventory holding cost of the corresponding depot. If quantity on stock at the depot is not enough to satisfy the new demand, it is increased accordingly. Fig. 1 presents an example in which the first delivery date of retailer **B** (at period 2) is shifted to the earliest possible date (period 1 assuming that the depot has available capacity). Delivery at period 3 is shifted to period 2 and at period 3, shipment is not longer required. Note that the maximum number of vehicles might be reduced (it does for this specific example) and the inventory policies for retailer B have changed, so the inventory policies for the depot have to be reviewed. We have preferred to shift the first delivery date only in order to perform fast computations. This neighborhood might be extended to any delivery date increasing the complexity of the operator.

- *2-shifted delivery date*: Same as *Shift delivery date* but considering a couple of customers sharing the same depot. This movement aims to synchronize the deliveries of retailers over time.
- *Depot reallocation*: The allocation of a retailer is shifted to another depot. Inventory policies at the retailer remain the same in this case. The set of scheduled visits over the planning horizon is inserted into the routes departing from the new depot in the given period. A best insertion cost is computed. Depot capacity over the complete planning horizon must be taken into consideration.
- *Depot allocation swap*: Two retailers allocated to different depots are exchanged in their depot allocation. The set of scheduled visits over the planning horizon is removed from the solution. Without changing inventory policies for the retailer (schedule visits remain unchanged), a best insertion procedure is implemented. Depot inventory policies must be re-evaluated if available stock is not sufficient to supply the new assigned retailer.

Our LS is embedded on a variable neighborhood descent (VND) structure (Hansen and Mladenovic, 2003). It evaluates each neighborhood and when an improving movement is found, the search starts again from the first neighborhood. If no improving movement is found in the current neighborhood, the search continues with the next one. LS applies the first movement that improves the solution except for *2-shifted delivery date* that uses best-improving movement strategy.

Furthermore, LS requires an important computational effort. It could be performed with some probability $\pi_{LS} \leq 1$ to make the algorithm run faster.

3.4. Intensification

With the purpose of evaluating the interactions between inventory and routing decisions, an intensification component is proposed. Algorithm 1 presents how to re-evaluate inventory-routing decisions with a dedicated procedure. Similar to a Large-Neighborhood Search (Pisinger and Ropke, 2010), inventory-routing decisions are destroyed and repaired. Thus, for n_{inten} iterations, a dynamic lot-sizing problem (DLSP) is first solved with a MIP solver (line 3). This problem consists of determining the optimal quantities to stock at the depots and retailers if location–allocation decisions are fixed. It is called “dynamic” because demand is allowed to vary over periods, not because decisions are re-optimized dynamically (Wagner and Whitin, 1958). The MIP presented in Section 3.1 is reduced by fixing y and f variables (location–allocation decisions) with the values in the supply chain design S at the current solution.

Algorithm 1. Procedure: *Intensification* $(C^*, S, R_{(S)})$.

1: $n_p = 0, \mathscr{L} = \{\}, \mathscr{L}' = \{\}, C_{best} = \infty$
2: **for** $k_3 = 1$ to n_{inten} **do**
3: $S \leftarrow \text{DLSP}(C^*, S, \mathscr{L}')$
4: $R_{(S)} \leftarrow \text{RCWA}\ (S)$
5: LocalSearch $(S, R_{(S)})$
6: **if** $C(S, R_{(S)}) < C_{best}$ **then**
7: $C_{best} = C(S, R_{(S)})$
8: $S_{best} \leftarrow S$
9: $R_{best} \leftarrow R_{(S)}$
10: **else**
11: $n_p = n_p + 1$
12: **if** $n_p = n_0$ **then**
13: createTabuLists $(\mathscr{L}, \mathscr{L}')$
14: $C^* \leftarrow \text{perturbation}(C^*, \mathscr{L}, \mathscr{L}')$
15: $n_p = 0$
16: **end if**
17: **endif**
18: **end for**

Once the inventory policies for retailers and depots have been repaired, routing decisions are recomputed. A randomized version of the Clarke and Wright algorithm is proposed to build $R_{(S)}$ (line 4). The solution is improved by the local search detailed in (3.3) (line 5). The function $C(S, R_{(S)})$ returns the cost of a solution S given routes $R_{(S)}$.

To avoid local optima and diversify initial S at line 3, a perturbation procedure is applied (line 13) every n_p iterations without improvement. The first perturbation consists on randomly selecting a retailer and its nearest neighbor. Both are included into a list $\mathscr{L}$. The procedure is repeated until $\mathscr{L}$ has more than 10% of the elements of J. For a single random period t, the delivery cost from j to its corresponding depot i will be $C^*_{ijt} = 0, \forall j \in \mathscr{L}$. This strategy targets the synchronization of deliveries in a given period t for retailers with close proximity. Additionally, cuts are added to the MIP by forcing the solution to visit all retailers j at a random period $t_j, \forall j \in \mathscr{L}'$. $\mathscr{L}'$ is generated as a random and independent subset of J.

3.5. Post-optimization

Given the best solution $(S, R_{(S)})$ found by the hybrid heuristic, a dedicated post-optimization procedure to intensify allocation-routing decisions is proposed in the form of an iterated local search (ILS) (Lourenço et al., 2003). The pseudo-code of the procedure is presented in Algorithm 2. In line 3, S and $R_{(S)}$ are mutated by modifying the allocation of a fixed percentage γ of randomly selected retailers. Without changing their inventory policies, they are reassigned to a different random depot that is already open and has available capacity. If reallocation is not feasible for the set of opened depots, a new random depot is opened. A best insertion procedure is implemented to include the visits within $R_{(S)}$ over the planning horizon. In line 4, the local search procedure is applied. If the current solution improves the best solution found, the best solution is updated at lines 5–9. If the opposite is true, the current solution is discarded and mutation is repeated on the best solution. The procedure is repeated for up to N_1 improving iterations or up to N_2 iterations without improvement of the best solution. In the worst case, $N_1 + N_2$ mutations and local search procedure calls are performed.

Algorithm 2. Procedure: $ILS(S, R_{(S)})$.

$i=1$ and $j=1$
while $i \le N_1$ and $j \le N_2$ **do**
  mutate $(S, R_{(S)})$
  LocalSearch $(S, R_{(S)})$
  if $C(S, R_{(S)}) < C_{best}$ **then**
    $C_{best} = C(S, R_{(S)})$
    $S_{best} \leftarrow S$
    $R_{best} \leftarrow R_{(S)}$
    $i = i+1$
  else
    $S \leftarrow S_{best}$
    $R_{(S)} \leftarrow R_{best}$
    $j = j+1$
  end if
end while

3.6. Algorithm overview

The components described in Sections 3.1 and 3.4 are integrated in the multi-start hybrid heuristic reinforced by the ILS procedure presented in Section 3.5. Algorithm 3 details the complete procedure. At line 7, a supply chain design S is computed by solving a SCDP linear model using a commercial solver as explained in Section 3.1. The components of the initial solution constructed by this operator are the location-allocation and inventory decisions. Then, the randomized routing heuristic detailed in Section 3.2 is performed to optimize the routing decisions $R_{(S)}$ (line 8) and to potentially improve allocation decisions. With some probability π_{LS} the local search procedure is applied. Routing, inventory and allocation decisions are potentially improved at this step. Subsequently, the update of the C^* matrix is performed (line 12), as explained in Section 3.1.

The intensification procedure is called in line 14 to potentially improve inventory-routing decisions (see Section 3.4) in multi period context. Lines 17–21 update the best found solution S_{best} and R_{best}. A new solution is explored (lines 6–22) until no improvement is perceived or a maximum of MAX_{it} iterations are performed. A tabu list τ is created at line 23 to limit the search of the supply chain design generator. τ is used in the next call of the SCDP procedure, where a new solution S is forced to close the depot belonging to the tabu list. τ is cleared once the procedure SCDP is performed. The post-optimization procedure described in Section 3.5 is performed in the multi-depot context in lines 25–28. Every decision component is fixed except for allocation-routing which might be improved by this operator.

Algorithm 3. Main algorithm (overview).

1: $S, R(S) \leftarrow 0$
2: $S_{best}, R_{best} \leftarrow 0$
3: $C_{best} = \infty$; $\tau \leftarrow \varnothing$
4: **for** $k_1 = 1$ to m **do**
5:   $C^* \leftarrow random \cdot C$
6:   **while** (Improvement or less than MAX_{it} iterations) **do**
7:     $S \leftarrow \mathrm{SCDP}(C^*, \tau)$
8:     $R_{(S)} \leftarrow$ RECWA (S)
9:     **if** $random < \pi_{LS}$ **then**
10:       LocalSearch $(S, R_{(S)})$
11:     **end if**
12:     Update $(C^*, R_{(S)})$
13:     **if** $p > 1$ **then**
14:       $Intensification(C^*, S, R_{(S)})$
15:       Update $(C^*, R_{(S)})$
16:     **end if**
17:     **if** $C_{best} > C(S, R_{(S)})$ **then**
18:       $C_{best} = C(S, R_{(S)})$
19:       $S_{best} \leftarrow S$
20:       $R_{best} \leftarrow R_{(S)}$
21:     **end if**
22:   **end while**
23:   $\tau \leftarrow$ RandomDepotClosure (S)
24: **end for**
25: **if** $m > 1$ **then**
26:   $S \leftarrow S_{best}$
27:   $R_{(S)} \leftarrow R_{best}$
28: $ILS(S, R_{(S)})$
29: **end if**

4. Computational study

The algorithms were coded in language Mosel and solved with Xpress-IVE 7.0, 64-bits. Tests are performed on an Intel Xeon with 2.80 Ghz processor and 12 GB of RAM.

4.1. Instances

Since there are no available benchmark instances for the problem under consideration, 20 ILRP instances were randomly generated. They have the following features: $m : \{5\}$ depots, $n : \{5, 7, 15\}$ retailers, $p : \{5, 7\}$ periods. The names of the instances correspond to its size. They are labeled as $m{-}n{-}p{-}x$ where m indicates the number of depots, n the number of retailers, p the number of periods and x is used to itemize and differentiate instances with the same size ($x \in \{a, b, c, \ldots\}$).

Demand at retailer j for period t is generated with a Normal distribution: $d_{jt} \sim N(\mu_j, \sigma_j)$, were $\mu_j \in [5, 15]$ and $\sigma_j \in [0, 5]$. The opening costs for depots O_i are generated randomly with a Normal distribution with parameters (μ_i, σ_i) chosen from the set of pairs $\{(1000, 20), (5000, 100), (8000, 300)\}$ while the replenishment cost s_i is chosen from the set $\{100, 500\}$.

The coordinates (X_i, Y_i) for facility $i \in V$ are randomly generated in a square of size 100×100. The function NINT $(\cdot)$ approximates to the closest integer value. Transportation cost $c_{ij} = \mathrm{NINT}(100 \cdot \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2})$. Vehicle capacity Q is a random value $= 5 \cdot b$ where b is a random integer in the interval [3,15]. The cost of using a vehicle F is selected from the set $\{350, 1000, 5000\}$. Depot capacity W_i is randomly generated in the interval $[D/3, D]$, where $D = \sum_{j \in J} \sum_{t \in H} d_{jt}$. Retailer's capacity W_j are randomly generated in the interval $[g_j, 3 \cdot g_j]$ where $g_j = \max_t \{d_{jt}\}$. Initial inventories B_j were chosen from the set $\{0, d_{j1}\}$ for retailers and B_i from the set $\{0, 10 \cdot D/n\}$ for depots. Inventory holding costs for a single period $t \in Ho$ at retailers and depots $j \in V$, $q_{j,t,t+1}$ are generated in the interval [0.03, 0.50]. The inventory holding costs for k periods as $q_{j,t,t+k} = \sum_{l=t}^{t+k-1} q_{j,l,l+1} + k \cdot \xi_2$, where ξ_2 represents the unitary obsolescence penalty cost. it is generated as $\xi_2 \sim Unif[0.01, 0.02]$.

Further, larger instances for classical subproblems are used to test the validity of the algorithm. 18 LRP instances with capacitated vehicles, 5 capacitated depots and 20, 50 and 100 retailers and single period available in http://prodhonc.free.fr were used to test our hybrid heuristic. They are also solved by Prins et al. (2006). Likewise, 40 "order-up to level" benchmark instances for the IRP with single depot, single vehicle, 5–40 customers, three periods and holding cost between [0.01, 0.05] are considered. They are available at http://www-c.eco.unibs.it/~bertazzi/abls.zip. Our model is slightly adapted to an "order-up to level" policy required on these IRP instances. A set

Table 2
Impact measure of valid inequalities.

Instance	UB	Version	GAP/UB (%)					
			20 s	100 s	500 s	1000 s	5000 s	8200 s
5-5-5-a	93 718.5	V.0	–	94.31	71.06	67.84	62.82	43.94
		V.1	117.56	90.60	35.17	35.17	35.16	35.16
		V.2	83.88	83.88	68.32	68.32	29.75	29.75
		V.3	–	–	79.88	72.63	46.81	39.18
5-5-5-b	62 494.5	V.0	–	42.69	39.42	39.42	34.49	16.67
		V.1	–	57.93	53.66	46.36	23.63	16.42
		V.2	74.76	74.76	43.46	38.79	35.03	30.57
		V.3	–	29.38	28.85	28.85	28.85	28.85
5-5-5-c	69 760.3	V.0	–	110.10	77.08	70.68	58.11	58.11
		V.1	109.93	89.24	78.69	73.81	65.91	54.89
		V.2	–	–	76.47	76.47	54.83	54.83
		V.3	–	–	81.29	79.97	64.81	60.38
5-5-7-a	77 404.1	V.0	–	–	53.78	53.78	48.01	48.01
		V.1	–	–	–	133.90	57.56	57.56
		V.2	–	69.24	69.24	69.24	38.70	38.70
		V.3	–	–	86.30	65.16	60.66	58.35
5-5-7-b	110 940	V.0	–	–	147.89	147.89	88.48	88.48
		V.1	–	–	90.25	90.25	90.25	89.10
		V.2	–	–	–	–	128.55	128.55
		V.3	–	–	–	–	–	132.23
5-5-7-c	94 150.2	V.0	–	61.95	50.90	50.43	37.01	37.01
		V.1	–	55.09	42.96	42.96	26.05	26.05
		V.2	–	57.39	24.08	24.08	24.08	24.08
		V.3	–	82.69	65.76	65.76	57.51	53.26

of constraints forcing the replenishment of a retailer j so that the stock raises up to W_j if visited is included.

4.2. MIP solver performance and valid inequalities impact analysis

Bounds for the ILRP are computed by solving the MIP presented in Section 2 using a commercial solver before solving it using the proposed hybrid heuristic. This section is intended to show the impact of the valid inequalities proposed by Theorems 1 and 2. Four configurations of the solver are proposed. V.0 corresponds to the MIP presented in (2). V.1 corresponds to V.0 with Eq. (30). V.2 corresponds to V.0 with Eq. (31). Finally, V.3 corresponds to the complete strengthened formulation composed by V.0 plus Eqs. (30) and (31). Despite the quality of the final solution, the presolve procedures and the preliminary heuristics of the commercial solver were deactivated here to isolate the effect of adding the valid inequalities. Best found feasible solution (UB) within a time limit of 8200 s with default settings and percent gap at 20, 100, 500, 5000 and 8200 s are reported in Table 2 for a subset of instances to be brief. Besides, we are not able to guarantee the quality of the lower bound provided by the solver resulting from the continuous relaxation of the model given its variations due to the new valid inequalities among the four versions. Then, a fair comparison might be made by comparing the gap of all versions to the same bound value.

It is possible to analyze, for example, on instance 5-5-7-a that V.2. is the fastest to provide (within 100 s) a feasible solution that is 69.24% larger than UB. Not adding any valid inequality would result in having within 500 s the first feasible solution that is 53.78% larger than UB. V.2 computes the best solution among the compared versions within 5000 s and 8200 s. V.1 and V.3 are outperformed by V.0 and V.2 in this case. Results show that any version dominates systematically. On one hand, adding both sets of constraints does not seem to speed up the process. On the other, excluding them all (V.0) often provides feasible (but low quality) solutions on short computing times. In any case, the computational burden is significant considering that the instances are smaller than potential real-life instances. A more sophisticated method to dynamically add valid inequalities in a Branch and Cut frame is future research. In subsequent research, we decided to keep both the additional valid inequalities and presolve procedures.

4.3. Preliminary tests for the hybrid heuristic

The parameters to calibrate are described next:

- α is used to compute C and to build an initial solution by the SCDP procedure as explained in Section 3.1. Preliminary tests indicate that α should be between 0.2 and 0.4.
- MAX_{it} represents a limit on the iterations performed in the first phase of the algorithm. The considered levels are 4 and 100. For less than 4 iterations, solution quality is not acceptable. 100 iterations are equivalent to iterating until no improvement is achieved.
- π_{LS}: the probability of performing a local search procedure in the first phase of the algorithm. The considered levels were 40% and 100%.
- N_1 and N_2 represent the number of iterations with and without improvement respectively. They are used in the post-optimization procedure. These are tested considering levels 15 and 30 independently.

The minimum number of instances required for the tuning test N is such that the acceptable standard error (SE) is larger than $\mathrm{SD}\cdot\sqrt{1/N}$, where SD stands for the standard deviation of the gap to UB in the sample as proposed by Cobb (1998). An approximate confidence interval for the performance of our algorithm is in the form: Mean Gap $\pm 2\times SE$. It has about 95% chance of containing the true value for this gap. We consider an acceptable SE to be below 0.25%. In a preliminary test, the maximum estimated standard deviation for a combination of parameters SD=1.44. That is, at least 33 instances are required to build acceptable intervals.

We account for 32 candidate combinations of parameters considering 5 parameters with 2 levels each. The one providing the most stable results and competitive computing times was selected. We ranked for each instance (15 for the ILRP and 18 for the LRP), the best gap and best CPU independently. The non-parametric test of Friedman proved that at least two combinations of parameters in the sample represent populations with different mean ranks (p-value$_{GAP}$=0.56, p-value$_{CPU}$=0.76). Instead of choosing the parameter combination with minimal GAP or CPU, the combination with the best trade-off between mean rank for gaps and mean rank for CPU is chosen. This approach provides a more stable algorithm, as the chances of obtaining acceptable quality solutions are less variable. Be aware that deeper research on parameter tuning might lead to lower average gaps. For the following analysis: $\alpha=0.4$, $MAX_{it}=4$, $\pi_{LS}=100\%$, and $N_1=N_2=15$.

4.4. Results

Table 3 presents, for ILRP instances, the comparison between 3 heuristics: (1) A time-constrained commercial solver with a time limit of 2.5 h; (2) our hybrid heuristic; and (3) a sequential heuristic (H1) that aims to emulate the traditional sequential approach. H1 is equivalent to compute a supply chain design using the commercial solver (fixing location–allocation decisions) and to make inventory-routing decisions through the procedure described in Section 3.4. In detail, column UB presents the best feasible solution found by the solver within the time limit. Columns two and three present the gap between the solution found in 60 and 500 s and UB. Column five (CPU UB) presents the time when UB was found by the solver in seconds. Columns six to eight present the average results of our heuristic in three runs. The

Table 3
CPU times and average gap for HH, H1 and solver for random ILRP instances.

Instance	SOLVER				HYBRID HEURISTIC			H1		
	GAP (60s)	GAP (500s)	UB	CPU UB	COST	GAP (%)	CPU (s)	COST	GAP (%)	CPU (s)
5-5-5-a	3.7	0	93718.5	173	93625.3	−0.10	25.2	93625.3	−0.10	4.7
5-5-5-b	10.5	0.88	62494.5	2176	62206.9	−0.46	22.4	62897.7	0.65	4.0
5-5-5-c	23.9	0.34	69760.3	1356	70881.0	1.61	48.0	70964.2	1.73	7.9
5-5-5-d	3.9	0.01	93801.2	592	93451.2	−0.37	28.6	93451.2	−0.37	4.7
5-5-5-e	4.3	0	93851.0	212	94600.6	0.80	14.1	97788.7	4.2	4.3
5-5-7-a	9.7	2.95	77404.1	1093	70966.5	−8.32	320.7	75969.1	−1.85	89.3
5-5-7-b	50.8	25.07	110940.0	4243	107478.5	−3.12	394.9	112299.6	1.23	46.7
5-5-7-c	28.9	28.88	94150.2	8163	94152.9	0.00	328.3	100416.6	6.66	212.0
5-5-7-d	13.8	0.02	87744.2	3951	87744.2	0.00	62.1	91750.1	4.57	8.8
5-5-7-e	–	1.67	69025.9	1819	67275.4	−2.54	176.4	71017.7	2.89	44.8
5-7-5-a	110.1	1.53	68485.2	873	69739.3	1.83	55.0	71522.9	4.44	7.9
5-7-5-b	–	3.09	76339.1	1317	78662.5	3.04	114.8	79122.5	3.65	51.0
5-7-5-c	–	–	–	–	138998.0	–	4964.9	141696	–	1409
5-7-5-d	–	7.52	99988.9	7432	100001.0	0.01	222.3	106584.2	6.60	26.2
5-7-5-e	48.7	0.01	62010.1	1404	62234.0	0.36	81.2	63451.9	2.33	11.5
Average	28.0	5.14	82836.7	2486	86134.5	−0.52	457.3	88819.2	2.62	128.8

Table 4
CPU and average gap for HH, H1 and solver for large ILRP instances.

Instance	Solver		Hybrid heuristic			H1		
	UB	CPU UB	Cost	GAP (%)	CPU (s)	Cost	GAP (%)	CPU (s)
5-15-5-a	156 959.0	26 726	113 434.3	−27.73	1863.5	124 376.3	−20.76	101.8
5-15-5-b	233 359.1	22 055	172 743.3	−25.98	2001.3	177 696.7	−23.85	443.6
5-15-5-c	–	–	210 333.0	–	14 301.9	212 849.0	–	4076.9
5-15-5-d	403 222.5	28 844	165 939.7	−58.84	1530.7	176 477.0	−56.23	330.2
5-15-5-e	–	–	228 467.7	–	22 661.3	236 799.7	–	4880.7
Average	264 513.5	25 875	178 183.6	−37.52	8471.8	185 639.7	−33.61	1966.6

Table 5
Benchmark on location-routing problem instances.

Instances	#	BKS	Hybrid heuristic[a]					Prins et al. (2006)[b]		
			Cost	Gap	CPU (s)	$\%_{min}$	$\%_{max}$	Cost	Gap	CPU (s)
20R-5D-1P	4	45 087	45 154	0.13	0.97	0.10	0.19	45 144	0.10	0.17
50R-5D-1P	8	74 113	74 782	0.68	4.69	0.26	1.23	74 701	0.81	2.14
100R-5D-1P	6	199 165	201 957	1.37	28.7	1.04	1.74	202 210	1.55	22.1
Average		109 346	110 590	0.79	11.9	0.48	1.17	110 636	0.90	8.35

[a] Intel Xeon with 2.8 GHz processor and 12 GB of RAM.
[b] Pentium 4 with 2.4 GHz processor and 512 MB of RAM.

average solution cost, the average gap to UB (GAP) and average computation time in seconds (CPU) are reported respectively. Columns nine and ten present the gap to UB and CPU for H1.

On average, our hybrid heuristic outperforms the commercial solver preset as a truncated search by 0.52% with an average computing time of 457 s. 6 out of 15 new best solutions are found, and other 4 solutions have a gap to UB inferior to 0.4%. When compared to the solutions computed on a similar average computation time (500 s), our method outperforms the solver by 5.66% and improves the solutions of 10 out of 15 instances. Besides, the only interest of H1 is its speed. The traditional approach (H1) provides solutions that are about 2.62% more expensive than UB found by the solver and more than 3% higher than our hybrid heuristic.

5-7-5-c is a difficult instance. The solver was not able to find a feasible solution within the preset time limit. In fact, the first feasible solution (UB) has a cost of 176191.5 computed in 6.7 h (24180 s). The traditional approach computes a solution of 149459.2 (−15.17% lower) in 1408.8 s. Furthermore, the solution found by our hybrid heuristic has a cost 21.1% inferior to UB computed almost 8 times faster.

Similarly, Table 4 compares the ILRP instances with 15 clients to the commercial solver with a time limit of 9 h. The solver was able to find integer solutions for only 3 out of 5 instances. After presolve procedures, instance 5-15-5-c has 49 775 variables and 32 562 constraints. Instance 5-15-5-e has 45 525 variables and 77 488 constraints. Columns four to six present the average cost, gap to UB and average computational time of our heuristic in three

Table 6
Benchmark on inventory-routing problem instances.

Instance	♯	z*	Hybrid heuristic[a]					Benchmark[b]		
			Cost	Gap (%)	CPU (s)	$\%_{min}$	$\%_{max}$	Archetti et al. (2011)	Bertazzi et al. (2002)	CPU (s)
5R-1D-3P	5	1418.7	1418.7	0	3.0	0	0	0	2.88	3.0
10R-1D-3P	5	2228.7	2236.2	0.34	7.8	0	0.49	0	0.78	12.8
15R-1D-3P	5	2493.5	2520.4	1.08	28.9	0.03	2.44	0	2.56	41.4
20R-1D-3P	5	3053.0	3160.2	3.51	82.6	1.71	6.61	0.02	3.83	104.2
25R-1D-3P	5	3451.1	3532.9	2.37	191.1	0.38	4.71	0	2.99	258.8
30R-1D-3P	5	3643.2	3731.0	2.41	413.2	0.94	4.81	0.02	3.60	515.0
35R-1D-3P	5	3846.9	4016.9	4.42	812.0	2.29	7.16	0.04	4.46	808.8
40R-1D-3P	5	4125.7	4393.9	6.50	1256.4	5.4	7.7	0.06	6.46	1168.6
Average	5	3032.6	3126.29	2.58	349.36	1.35	4.24	0.02	3.45	364.08

[a] Intel Xeon with 2.8 GHz processor and 12 GB of RAM.
[b] Intel Dual Core 1.86 GHz processor and 3.2 GB RAM.

runs per instance. The proposed heuristic solved every instance with an average time of 2.33 h. Furthermore, we improved the average BKS in 37.52%. On the other hand, even if H1 solved every instance in less than 1.4 h, solutions are 3.91% higher than the hybrid heuristic.

High robustness is also achieved by our algorithm. The square coefficient of variation for the solution value is always inferior of $1.0\,e^{-3}$ for every instance tested. Therefore, there is little interest in executing more runs of the algorithm for the same instance.

Table 5 sums up the results on classical LRP instances. Average results for instances with 5 depots, 20 retailers (20R-5D-1P), 50 retailers (50R-5D-1P) and 100 retailers (100R-5D-1P) are presented. Column (♯) presents the number of instances per data set. The average of best known solution (BKS) in each set are taken from http://prodhonc.free.fr. The average cost (cost), gap to BKS (gap) and computation time (CPU) of our hybrid heuristic for three runs and the minimum and maximum gap ($\%_{min}$, $\%_{max}$) to BKS are shown in columns 4-8 respectively. They show an average gap of 0.79% computed in 11.9 s. The average gap to BKS is always between 0.48% and 1.17%. We compare our methodology with the dedicated method of Prins et al. (2006) (coded in C++ and executed on a Dell OPTIPLEX GX260 PC, 512 MB of RAM, with a Pentium 4 processor clocked at 2.4 GHz and Windows XP) detailed in columns 9–11. Even if it is not dedicated to the LRP, our method is competitive with the algorithm of Prins et al. (2006). We choose the latter for two main reasons: (i) it provides very good results and (ii) it allows a fair comparison since it uses also the Clarke and Wright algorithm.

Results are also competitive on IRP instances as shown in Table 6. The number of retailers is shown in column (♯) and the average optimal solution in column z^*. Results of our heuristic for average, minimum, and maximum gap, and total computation time and computation time to best solution (CPU_{best}) for three runs are presented in columns five to eight, showing an average gap of 2.58% computed in 349.36 s. Results for percentage gap to optimal solution of Archetti et al. (2011) and Bertazzi et al. (2002) are presented in columns nine and ten. Corresponding computation times for Archetti et al. (2011) (not available for Bertazzi et al., 2002) are in column 11 (CPU) using an Intel Dual Core 1.86 GHz and 3.2 GB RAM and coded in C++. The approach computes solutions of intermediate quality between Archetti et al. (2011), Bertazzi et al. (2002) within computation times that are similar, even when it is not a dedicated method.

It should, however, be noted that different computers and programming languages were used when comparing algorithms. We have provided this information from benchmark for guidance only since a reliable comparison of computing times is difficult. Even if it were possible to obtain an estimation of the MFlops for each computer (see for example Dongara, 2013) to evaluate each speed factor, it is assumed full exploitation of parallelism. Our algorithms do not satisfy this assumption and we cannot guarantee that benchmark algorithms do it. Further, the used languages differ as well. Nonetheless, the proposed hybrid algorithm executes within reasonable time for benchmark subproblems.

5. Conclusions

We present the combined Inventory-Location-Routing Problem (ILRP) as an approach to supply chain design considering inventory management and routing cost in order to overcome the fact that traditional approaches decompose decisions and often provide suboptimal solutions. We consider a discrete and finite planning horizon and a two-echelon supply chain. Assumptions include a homogeneous fleet of vehicles, and deterministic, non-constant demand. Seasonal holding costs and obsolescence penalty costs are additional (but not restrictive) features of our model. Decisions that must be taken simultaneously are: (1) location decisions of depots, (2) inventory decisions at both echelons of the supply chain, (3) allocation decisions of retailers to depots, and (4) multi-period routing decisions.

We propose a hybrid approach to solve a supply chain design problem with estimated distribution costs using exact methods while the remaining routing decisions are computed by heuristic procedures. By alternating between decisions spaces and information sharing, the algorithm manages to optimize globally the components of the problem without oversimplifying it. Results for randomly generated instances show significant cost savings over the traditional approach and efficient computation when compared to commercial solvers. The ILRP reduces to the LRP and the IRP under certain conditions. Our tests show a robust performance over larger benchmark instances for both the LRP and the IRP.

Future research comprises the ILRP with two routing decision levels as in the 2E-LRP and the ILRP considering maritime transportation constraints.

Acknowledgments

This research is partially supported by Champagne-Ardenne Regional Council (France) and Centro de Estudios Interdisciplinarios Básicos y Aplicados - CEIBA (Colombia). We thank two anonymous reviewers for their valuable and constructive suggestions to this paper.

Appendix A. Notation for mathematical formulation

Sets

- I Set of candidate depots
- J Set of retailers
- $V=\{I \cup J\}$ Set of facilities
- H Set of periods in the planning horizon
- $H_0=H \cup \{0\}$ Planning horizon including initial conditions
- $H'=H \cup \{p+1\}$ Planning horizon including final conditions
- K Set of available vehicles
- A Set of arcs connecting facilities

Parameters

- n Number of retailers
- p Number of periods
- m Number of candidate depots
- r Number of available vehicles
- G Graph defining the ILRP. $G=(V, A, C)$
- C Cost matrix where c_{ij} is the traveling cost for arc $(i,j) \in A$ [$]
- W_i Storage capacity at facility $i \in V$ [units]
- O_i Opening cost [$]
- s_i Ordering cost for depot [$]
- Q Vehicle capacity [units]
- F Cost for using a vehicle at least once over H [$]
- B_i Initial inventory [units]
- q_{itl} Unitary holding cost at facility i from period t up to period l [$/unit]
- d_{jt} Demand at retailer $j \in J$ in period $t \in H$ [units]

Decision variables

- y_i Binary variable. It is equal to 1 iff depot $i \in I$ is opened
- f_{ij} Binary variable. It is equal to 1 iff retailer $j \in J$ is allocated to depot $i \in I$
- x_{ijkt} Binary variable. Equal to 1 iff the arc $(i,j) \in A$ is crossed from i to j by vehicle $k \in K$ on period $t \in H$
- T_i Maximum number of vehicles allocated to depot i
- w^1_{itl} Quantity held in stock at depot i(1st echelon) from period t up to period l [unit]
- w^2_{j0t} Quantity held in stock at retailer i(2nd echelon) from initial period 0 up to period l [unit]
- w^2_{ijtlk} Quantity delivered by depot i with vehicle k held in stock at retailer j(2nd echelon) from period t up to period l [unit]
- z_{ti} Binary variable. It is equal to 1 iff depot i is replenished at period t

Appendix B. Notation for heuristic procedure

- $S=(I,F,W)$ Supply chain design
- I set of selected depots to open
- F Depot assignment for each retailer
- W Matrix of dimensions $(m+n)\times(p+1)\times(p+1)$ in which each element w_{itl} indicates, for each facility $i \in V$, the quantity of product arriving in period $t \in H_0$ that will remain in stock until period $l \in H'$ [unit]
- $R_{(S)}$ Set of selected routes in the solution
- C^* Matrix with estimated delivery cost from depots to retailers (assignment costs) [$]
- $\hat{x}_{ijt}$ Binary Decision variable indicating whether depot i supplies retailer j at period t
- $\hat{w}^2_{ijtl}$ Decision variable indicating the quantity supplied by depot i, held in stock at retailer j from period t to period l [unit]
- ξ_1, ξ_2 Uniform random variables
- α Parameter to estimate initial distribution costs
- π_{LS} Probability of performing a local search procedure in the first phase of the algorithm
- N_1 Parameter of maximum number of improving iterations
- N_2 Parameter of maximum number of iterations without improvement
- MAX_{it} Parameter to limit on the iterations performed in the first phase of the algorithm
- $\mathcal{L}, \mathcal{L}'$ tabu lists to synchronize retailer visits and diversify initial S

References

Ambrosino, D., Scutellà, M.-G., 2005. Distribution network design: new problems and related models. European Journal of Operational Research 165 (3), 610–624.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A., 2010. Invited review: industrial aspects and literature survey: combined inventory management and routing. Computers & Operations Research 37, 1515–1536.

Archetti, C., Bertazzi, L., Hertz, A., Speranza, M., 2011. A hybrid heuristic for an inventory routing problem. INFORMS Journal on Computing, Articles in Advance, 1–16.

Balcik, B., Beamon, B., Krejci, C., Muramatsu, K., Ramirez, M., 2010. Coordination in humanitarian relief chains: practices, challenges and opportunities. International Journal of Production Economics 126, 22–34.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., Wolfler-Calvo, R., 2011. A branch-and-cut method for the capacitated location-routing problem. Computers & Operations Research 38 (6), 931–941.

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

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

Brandão, J., Mercer, A., 1997. A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. European Journal of Operational Research 100, 180–191.

Clarke, G., Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 12 (4), 568–581.

Cobb, G.W., 1998. Introduction to Design and Analysis of Experiments (Textbooks in Mathematical Sciences). Springer.

Dongara, J.J., 2013. Performance of Various Computers Using Standard Linear Equations Software. Technical Report CS-89-85, Electrical Engineering and Computer Science Dept., University of Tennessee., Computer Science and Mathematics Division, University of Manchester. ⟨http://www.netlib.org/benchmark/performance.pdf⟩.

Gebennini, E., Gamberini, R., Manzini, R., 2009. An integrated production–distribution model for the dynamic location and allocation problem with safety stock optimization. International Journal of Production Economics 122 (1), 286–304.

Golden, B., Raghavan, S., Wasil, E., 2008. The Vehicle Routing Problem: Latest Advances and New Challenges. Operations Research/Computer Science Interfaces, vol. 43. Springer, US.

Hansen, P., Mladenovic, N., 2003. Variable neighborhood search. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 57. Springer, New York, pp. 145–184.

Lourenço, H., Martin, O., Stützle, T., 2003. Iterated local search. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 57. Springer, New York, pp. 320–353.

Ma, H., Davidrajuh, R., 2005. An iterative approach for distribution chain design in agile virtual environment. Industrial Management & Data Systems 105 (6), 815–834.

Melo, M., Nickel, S., da Gama, F.S., 2009. Facility location and supply chain management: a review. European Journal of Operational Research 196 (2), 401–412.

Mendoza, J.E., Medaglia, A.L., Velasco, N., 2009. An evolutionary-based decision support system for vehicle routing: the case of a public utility. Decision Support Systems 46 (3), 730–742.

Mete, H.O., Zabinsky, Z.B., 2010. Stochastic optimization of medical supply location and distribution in disaster management. International Journal of Production Economics 126 (1), 76–84.

Miranda, P., Garrido, R., 2009. Inventory service-level optimization within distribution network design problem. International Journal of Production Economics 122 (1), 276–285.

Nagy, G., Salhi, S., 2007. Location-routing: issues, models and methods. European Journal of Operational Research 177 (2), 649–672.

Pisinger, D., Ropke, S., 2010. Large neighborhood search. In: Gendreau, M., Potvin, J.-Y. (Eds.), Handbook of Metaheuristics. International Series in Operations Research & Management Science, vol. 146. Springer, US, pp. 399–419.

Prins, C., Prodhon, C., Ruiz, A., Soriano, P., Wolfler-Calvo, R., 2007. Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic. Transportation Science 41 (4), 470–483.

Prins, C., Prodhon, C., Wolfler-Calvo, R., 2006. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR: A Quarterly Journal of Operations Research 4, 221–238.

Prodhon, C., 2011. A hybrid evolutionary algorithm for the periodic location-routing problem. European Journal of Operational Research 210 (April (2)), 204–212.

Raidl, G., Puchinger, J., 2008. Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In: Blum, C., Aguilera, M., Roli, A., Sampels, M. (Eds.), Hybrid Metaheuristics Studies in Computational Intelligence, vol. 114. Springer, Berlin/Heidelberg, pp. 31–62.

Salhi, S., Rand, G., 1989. The effect of ignoring routes when locating depots. European Journal of Operational Research 39 (2), 150–156.

Shen, Z.-J., Qi, L., 2007. Incorporating inventory and routing costs in strategic location models. European Journal of Operational Research 179 (2), 372–389.

Toth, P., Vigo, D. (Eds.), 2001. The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Wagner, H., Whitin, T., 1958. Dynamic version of the economic lot size model. Management Science 5 (1), 89–96.

Whybark, D.C., 2007. Issues in managing disaster relief inventories. International Journal of Production Economics 108 (12), 228–235.