

Research Article

Optimization of a Two-Echelon Location Lot-Sizing Routing Problem with Deterministic Demand

Laila Kechmane ¹, Benayad Nsiri ², and Azeddine Baalal¹

¹Department of Mathematics and Computing, MACS Laboratory, Faculty of Sciences, Hassan II University, Casablanca, Morocco

²Department of Applied Mathematics and Informatics, Ecole Normale Supérieure de l'Enseignement Technique, Mohamed V University, Rabat, Morocco

Correspondence should be addressed to Laila Kechmane; kechmanelaila@gmail.com

Received 12 October 2017; Revised 13 April 2018; Accepted 20 May 2018; Published 14 June 2018

Academic Editor: Josefa Mula

Copyright © 2018 Laila Kechmane et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper aims to solve a multiperiod location lot-sizing routing problem with deterministic demand in a two-echelon network composed of a single factory, a set of potential depots, and a set of customers. Solving this problem involves making strategic decisions such as location of depots as well as operational and tactical decisions which include customers' assignment to the open depots, vehicle routing organization, and inventory management. A mathematical model is presented to describe the problem and a genetic algorithm combined with a local search procedure is proposed to solve it and is tested over three sets of instances.

1. Introduction

The design of a distribution network comprises three major problems: facility location, vehicle routing, and inventory control, these problems belong to different categories of decisions, and researchers classify the supply chain decisions into strategic, tactical, and operational: strategic level decisions are fixed for long periods time, which could be years, such as location of facilities, tactical decisions are made for medium periods of time such as inventory management, and operational level concerns short-term decisions which includes distribution decisions.

Recently, many researchers have studied the benefit of combining two decision levels into a single model; the inventory location problem (ILP) combines the facility location problem (FLP) and inventory management; many studies have been done on the different variants of ILP such as Daskin et al. [1], Shen [2], Ozsen et al. [3], Shen and Qi [4], Miranda and Garrido [5], and Melo et al. [6].

Many papers in the literature studied the location routing problem (LRP) which combines the facility location problem and the vehicle routing problem (VRP); the importance of tackling them simultaneously is studied by Salhi and Rand [7]. The LRP consists in determining the optimal number of facilities, their location, allocation of customers to the open

facilities, and the vehicle routing organization. This problem has been studied by many researchers such as Prins et al. [8], Prins et al. [9], Duhamel et al. [10], Benlenger [11], Baldacci et al. [12], Ting et al. [13], Contardo et al. [14], and Kechmane et al. [15]. A recent survey on this problem is presented by Prodhon [16].

The inventory-routing problem (IRP) integrates inventory management and the vehicle routing problem, and the deterministic version of this problem has been studied by Archetti et al. [17], Oppen [18], Archetti et al. [19], Moin et al. [20], Coelho et al. [21], and Coelho and Laporte [22]. Adelman [23], Hvattum and Lokketangen [24], Hvattum et al. [25], and Mete and Zabinsky [26] studied the stochastic version of the problem. A survey on IRP is presented by Vidovic et al. [27]. Other papers in the literature integrate inventory, lot-sizing, and vehicle routing decisions such as Nananukul [28]; the author developed a reactive Tabu search procedure for solving a production-distribution supply chain and improved results using a path relinking strategy; Cetinkaya et al. [29] studied an integrated multiproduct inventory lot-sizing and vehicle routing problem and presented an iterative solution approach in which the problem is decomposed into inventory and routing subproblems; Chen and Sarker [30] established an integrated optimal model of inventory lot-sizing vehicle routing of multisupplier single-manufacturer

with milk-run JIT delivery and solved it using an ant colony optimization algorithm; and Kechmane et al. [31] proposed a mathematical model to optimize transport cost and delivery lot-sizing in a single level logistic network and solved it using a MILP solver.

Few studies in the literature consider the strategic, operational, and tactical level decisions simultaneously, yet the vehicle routing organization depends on inventory control problem, which in turn depends on decisions taken in the location-allocation phase. The problem with the aim of facilities location, vehicle routing organization, and inventory management is known as location-inventory-routing problem (LIRP); inventory may be considered at plants, at final customers, or at both of them. Miranda and Garrido [32] studied the effect of ignoring inventory decisions while designing a supply chain network and the importance of managing inventories to reduce the total costs.

Liu and Lee [33] proposed a single product, multidepot location routing problem taking inventory control decisions into consideration where customers have stochastic demand and applied a two-phase heuristic method to solve the problem. Liu and Lin [34] divided the proposed problem by Liu and Lee [33] into a location-allocation problem and inventory-routing problem; they proposed a hybrid heuristic combining Tabu search with simulated annealing to solve the problem. Ambrossino and Scutella [35] proposed a linear programming model for the LIRP with deterministic demand where inventories must be managed at both plants and customers.

Shen and Qi [4] modified a LIRP model proposed by Daskin et al. [1] and used Lagrangian relaxation to solve sub-problems. Ahmadi-Javid and Azad [36] proposed a stochastic model to simultaneously optimize location, allocation, inventory, and routing decisions; they proposed a hybrid algorithm of Tabu search and simulated annealing. Hiassat and Diabat [37] proposed a multiperiod model formulated as Mixed Integer Program for the LIRP with deterministic demand for perishable product. A multiphase heuristic algorithm based on simulated annealing and ant colony system (ACS) is proposed by Ahmadi-Javid and Seddighi [38] to solve a location-routing-inventory model with multisource distribution networks.

Guerrero et al. [39] proposed an algorithm for multiperiod LIRP with deterministic demand in a three-layer supply chain composed of a single supplier, potential depots, and customers; a hybrid heuristic and exact solution method are developed to solve the problem as well as an intensification phase to investigate the inventory and routing decision where the location and customers assignments are fixed and a dynamic lot-sizing problem is solved using a MIP solver. Nekooghadirli et al. [40] studied a biobjective location-routing-inventory model with multiperiod, multiproduct, stochastic demand and probabilistic travelling time among customers; their model aims to minimize the total cost and maximize the average time for delivering commodities to customers; and Zhang et al. [41] presented a hybrid metaheuristic combined with simulated annealing method to solve a LIRP in a two-echelon network composed of multiple capacitated depots and final customers.

Genetic algorithm (GA) is a stochastic optimization method proposed by J. Holland (1962) and then developed by D. Goldberg (1989); it is inspired by the natural evolution in genetics where a population of individuals each represented as a chromosome and representing a solution is subject to selection, crossover, and mutation operators at each generation to breed a new generation.

GA have been successfully applied to a variety of optimization problems including transport problems combining two decision levels; Abdelmaguid and Dessouky [42] developed a genetic algorithm approach to the integrated inventory distribution problem. A biased random key genetic algorithm approach is developed by Chan et al. [43] to solve the inventory-based multi-item lot-sizing problem. Rungreunganaun and Woarawichai [44] applied genetic algorithms to solve an inventory lot-sizing problem with supplier selection under storage space; the objective of their research is to calculate the optimal inventory lot-sizing suppliers and minimize the inventory cost. Prins et al. [8] proposed a memetic algorithm combined with population management, which is a technique developed by Sorensen and Sevaux [45], to solve a capacitated location routing problem; Park et al. [46] presented a GA for the vendor-managed inventory-routing problem with lost sales; Marinakis and Marinaki [47] presented a bilevel genetic algorithm and applied it to a real life location routing problem; a hybrid GA for the multiproduct multiperiod inventory-routing problem is presented by Moin et al. [20]; Liao et al. [48] used a nondominated sorting genetic algorithm (NSGA-II) to optimize a multiobjective location-inventory problem; and Kechmane et al. [49] developed a memetic algorithm to solve a capacitated location routing problem.

Recently, some genetic algorithms have been developed to solve problems combining location, inventory and routing decisions; Forouzanfar et al. [50] used a GA to optimize the total cost for a LIRP in a supply chain with risk pooling, a hybrid genetic-simulated annealing algorithm is proposed by Li et al. [51] for the LIRP considering returns under e-supply chain environment, and Hiassat et al. [52] developed a GA approach for a LIRP with perishable products.

This paper deals with a multiperiod Location lot-sizing routing problem (LLSRP) with deterministic demand that aims to design a two-echelon supply chain network consisting of potential depots supplied by a factory and facing deterministic nonconstant demands of customers. The objective is to determine the location of depots to open, to assign customers to the opened depots, to organize the vehicles routing, and to manage inventories at depots and customers by determining quantities to send to them during each period of the planning horizon so that the overall cost is minimized and customers' demands are satisfied, such problem arises in many domains such as food distribution for supermarket chains, military operations, and pharmaceutical industry, to solve the LLSRP, and a hybrid genetic algorithm is developed. This paper is organized as follows: the description of the problem and a mathematical model are presented in Section 2, in Section 3 components of the proposed algorithm are presented in detail, computational results are given in Section 4, and a conclusion is given in the last section.

2. Problem Description and Mathematical Formulation

Let $G = (V, E, C)$ be a complete, weighted, and directed graph, $V = I \cup J$ is the set of nodes in the graph where I is the set of potential depots and J is the set of customers. $E = \{(i, j) : i \in V, j \in V, i \neq j\}$ is the set of arcs linking the different nodes in the graph and $C = \{c_{ij} / i \in V, j \in V\}$ with c_{ij} being the cost of the trip from node i to node j . T is a discrete and finite planning horizon. Each depot $i \in I$ has an opening cost O_i and an ordering cost ord_i . Each node $i \in I \cup J$ has a storage capacity cap_i . Each customer is assigned to a single depot. A homogenous unlimited fleet of vehicles K is available. A vehicle might serve multiple customers during its tour but a customer is served by only one vehicle. Vehicles have a limited capacity Q and a cost of use F . K_i^t is the number of vehicles used from depot i fleet during the period t . Each vehicle ends its tour at the depot it starts from.

Before presenting the mathematical model for the LLSRP, let the decision variables be $p_i = 1$ if the depot i is opened, $l_{ijtk} = 1$ if the vehicle k crosses the arc $(i, j) \in E$ on period t , and $l'_{it} = 1$ if the depot i is replenished on period t . $l''_{ij} = 1$ if customer j is affected to depot i . x_{it} is the delivery quantity received by depot i on period t . y_{ijtk} is the delivery quantity received by customer j from depot i in period t via the vehicle k . b_i^t is the quantity needed by node $i \in I \cup J$ in period t . stk_i^t is the available stock at node $i \in I \cup J$ at the end of time period t , stk_i^0 represents the initial stock at node $i \in I \cup J$. q_{it} is the cost of holding a unit from period t to period l at node $i \in I \cup J$.

$$\begin{aligned} \min \quad & \sum_{i \in I} O_i p_i + \sum_{i \in I} \sum_{t \in T} F K_i^t + \sum_{i \in I} \sum_{j \in I \cup J} \sum_{k \in K} \sum_{t \in T} l_{ijtk} c_{ij} \\ & + \sum_{i \in I} \sum_{t \in T} ord_i l'_{it} \\ & + \sum_{i \in I} \sum_{j \in J} \sum_{t \in T} \sum_{k \in K} (y_{ijtk} - b_j^t + stk_j^{t-1}) q_{jt(t+1)} \\ & + \sum_{i \in I} \sum_{t \in T} (x_{it} - b_i^t + stk_i^{t-1}) q_{it(t+1)} \end{aligned} \quad (1)$$

subject to

$$b_i^t = \sum_{j \in I} \sum_{k \in K} y_{ijtk} \quad \forall i \in I \quad \forall t \in T \quad (2)$$

$$stk_j^t = \sum_{i \in I} \sum_{k \in K} y_{ijtk} l''_{ij} - b_j^t + stk_j^{t-1} \quad \forall j \in J \quad \forall t \in T \quad (3)$$

$$stk_i^t = x_{it} l'_{it} - b_i^t + stk_i^{t-1} \quad \forall i \in I \quad \forall t \in T \quad (4)$$

$$b_j^t \leq stk_j^{t-1} + \sum_{i \in I} \sum_{k \in K} y_{ijtk} \quad \forall j \in J \quad \forall t \in T \quad (5)$$

$$b_i^t \leq stk_i^{t-1} + x_{it} \quad \forall i \in I \quad \forall t \in T \quad (6)$$

$$stk_i^t \leq p_i cap_i \quad \forall i \in I \quad \forall t \in T \quad (7)$$

$$stk_j^t \leq cap_j \quad \forall j \in J \quad \forall t \in T \quad (8)$$

$$\sum_{j \in J} y_{ijtk} \leq Q \quad \forall i \in I \quad \forall k \in K \quad \forall t \in T \quad (9)$$

$$\sum_{i \in I} l''_{ij} = 1 \quad \forall j \in J \quad (10)$$

$$l''_{ij} \leq p_i \quad \forall i \in I \quad \forall j \in J \quad (11)$$

$$\sum_{j \in I \cup J} l_{ijtk} - \sum_{j \in I \cup J} l_{jtk} = 0 \quad (12)$$

$$\forall i \in I \cup J \quad \forall t \in T \quad \forall k \in K$$

$$\sum_{i \in I \cup J} \sum_{k \in K} l_{ijtk} \leq 1 \quad \forall j \in J \quad \forall t \in T \quad (13)$$

$$\sum_{i \in I \cup J} \sum_{k \in K} l_{jtk} \leq 1 \quad \forall j \in J \quad \forall t \in T \quad (14)$$

$$\sum_{i \in I} \sum_{j \in J} l_{ijtk} \leq 1 \quad \forall t \in T \quad \forall k \in K \quad (15)$$

$$\sum_{i \in S} \sum_{j \in S} l_{ijtk} \leq |S| - 1 \quad \forall k \in K \quad \forall t \in T \quad \forall S \subseteq J \quad (16)$$

$$\sum_{h \in J} l_{ihkt} - \sum_{\substack{h \in I \cup J \\ h \neq j}} l_{hjt} \leq 1 + l''_{ij} \quad (17)$$

$$\forall i \in I \quad \forall j \in J \quad \forall k \in K \quad \forall t \in T$$

$$l_{ijtk} = 0 \quad \forall i, j \in I \quad \forall k \in K \quad \forall t \in T \quad i \neq j \quad (18)$$

$$\sum_{j \in J} \sum_{k \in K} l_{ijtk} = K_i^t \quad \forall i \in I \quad \forall t \in T \quad (19)$$

$$K_i^t \in \mathbb{N} \quad \forall i \in I \quad \forall t \in T \quad (20)$$

$$p_i \in \{0, 1\} \quad \forall i \in I \quad (21)$$

$$l_{ijtk} \in \{0, 1\} \quad \forall i \in I \cup J \quad \forall j \in I \cup J \quad \forall k \in K \quad \forall t \in T \quad (22)$$

$$l'_{it} \in \{0, 1\} \quad \forall i \in I \quad \forall t \in T \quad (23)$$

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

$$x_{it} \in \mathbb{R}^+ \quad \forall i \in I \quad \forall t \in T \quad (25)$$

$$y_{ijtk} \in \mathbb{R}^+ \quad \forall i \in I \quad \forall j \in J \quad \forall t \in T \quad \forall k \in K \quad (26)$$

$$stk_j^t \in \mathbb{R}^+ \quad \forall j \in I \cup J \quad \forall t \in T \quad (27)$$

$$b_j^t \in \mathbb{R}^+ \quad \forall j \in I \cup J \quad \forall t \in T \quad (28)$$

The objective function (1) sums the depots opening costs, the vehicles' cost of use, the total routing costs, the ordering costs for every depot, and the holding costs at customers and depots over the planning horizon.

Constraints (2) define the depots' need for every period. Inventory balance equations are presented by (3) and (4). Constraints (5) and (6) guarantee the satisfaction of customers and depots demand for every period. Equations (7) and (8) ensure that opened depots and customers storage

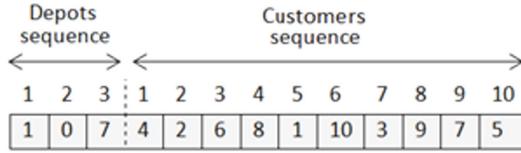


FIGURE 1: A chromosome representation.

capacities are respected. Equations (9) force the respect of vehicles capacity.

Constraints (10) ensure that each customer is assigned to a single depot and constraints (11) ensure that if a customer is assigned to a depot, the latter is opened. Constraints (12) force vehicles to leave every node they visit. Constraints (13) and (14) guarantee that each customer is visited by one vehicle at most per period. Constraints (15) ensure that each vehicle performs one route per period at the most.

Equations (16) are subtour elimination constraints, they guarantee the presence of a depot in every route. Constraint (17) links the vehicle routing variables and the assignment variables; it states that if a vehicle leaves a depot and visits a customer, then this customer must be assigned to the depot. Constraints (18) ensure that travels from a depot to another one are not allowed. Equations (19) provide the number of vehicles used from every depot fleet in every period. Finally (20)–(28) are standard integrality and nonnegativity constraints.

3. A Hybrid Genetic Algorithm for the Location Lot-Sizing Routing Problem

In order to solve the Location lot-sizing routing problem with deterministic demand, a hybrid genetic algorithm is proposed, and the different components of the algorithm are presented in the sequel.

3.1. Solution Representation. As mentioned earlier, in genetic algorithms, solutions are represented as chromosomes, this representation must be clearly defined; in the proposed algorithm, the encoding method for the location routing problem proposed by Prins et al. [8] is adopted; a chromosome consists of two parts; the first one relates to the depots; it indicates their status; if the gene corresponding to a depot is zero, it means that the depot is closed, else it contains the index of the first customer assigned to it. The second part of the chromosome indicates the set of customers assigned to each open depot. Figure 1 represents an example of a chromosome, the gene corresponding to depot 2 is zero, and this means that depot 2 is closed. The genes corresponding to depots 1 and 3 contain, respectively, the values 1 and 7, this means that both of them are open, and the sets of customers assigned to depots 1 and 3 are, respectively $\{4, 2, 6, 8, 1, 10\}$ and $\{3, 9, 7, 5\}$.

3.2. Initial Population. Initial population is composed of solutions obtained by InitialSol procedure presented in Algorithm 1 as well as random solutions in order to better explore the solution space.

```

Procedure: InitialSol
Input: Problem' data values
Output: Depots to open and customers' assignment
 $A_{ret} \leftarrow \emptyset$ 
repeat
  Select randomly an available depot  $d$ 
   $C_d \leftarrow \emptyset$ 
   $V_d \leftarrow J$ 
   $p_d \leftarrow 1$ 
  repeat
     $r \leftarrow \arg \min_{j \in V_d} \{d_{dj} / \forall i \in I : l''_{ij} = 0\}$ 
    if ( $Moy(C_d \cup \{r\}) \leq cap_d$ ) then
       $C_d \leftarrow C_d \cup \{r\}$ 
       $l''_{dr} \leftarrow 1$ 
       $A_{ret} \leftarrow A_{ret} \cup \{r\}$ 
       $V_d \leftarrow V_d - \{r\}$ 
    else
       $V_d \leftarrow V_d - \{r\}$ 
    end if
  until ( $V_d = \emptyset$ )
until ( $A_{ret} = J$ )

```

ALGORITHM 1: Depots to open and customers' assignment.

As well as the adopted chromosome representation, InitialSol procedure allows determining depots to open and customers' assignment to the open depots. Algorithm 1 represents the steps of this procedure; as long as there are customers not yet affected to the open depots, the procedure randomly selects and opens a depot; the closest customer not yet assigned to a depot is selected. If the average demand of this customer and those assigned to the depot during the planning horizon does not exceed the storage capacity of the depot, he is assigned to the depot; else, this procedure is repeated until the possibility of adding a customer is verified for all the customers not yet assigned to a depot.

Function *Moy* is defined as follows:

$$Moy(C_d) = \frac{\sum_{t \in T} \sum_{j \in C_d} b_j^t}{Card(T)} \quad (29)$$

where C_d represents the set of customers assigned to depot d .

3.3. Chromosome Decoding. Chromosomes, as well as InitialSol procedure, give the status of each depot and the set of customers assigned to it if it is open. In order to organize the vehicle routing, to manage the inventories at depots and customers over the planning horizon and to obtain a solution cost, two complementary procedures are proposed in the sequel.

Once customers are assigned to the open depots, procedure SolOrg1 (Algorithm 2) allows organizing the vehicles' routing and determining quantities to send to each customer over the planning horizon. For each open depot, an available vehicle from its fleet starts its route by visiting the closest customer affected to the depot who needs to be replenished, i.e., his stock level does not respond to the need of the day; as long as the vehicle can serve, the closest customer to the

```

Procedure: SolOrg1
Input: Problem' data values
Output: Routing organization and customers'
inventory management
S ← 0
for each  $i \in I/p_i = 1$  do
  S ← S +  $O_i + stk_i^0 * q_{i(0)1}$ 
  t ← 1
  while ( $t < t_{\max}$ ) do
     $b_i^t \leftarrow 0$ 
     $C'_i \leftarrow C_i$ 
    repeat
      Select an available vehicle  $k$ 
      S ← S + F
       $D_k^t \leftarrow 0$ 
       $C''_i \leftarrow C'_i$ 
      r ← i
      repeat
         $r' \leftarrow \arg \min_{j \in C''_i} \{d_{rj}/stk_j^{t-1} \leq b_j^t\}$ 
        if ( $cap_k \geq D_k^t + b_{r'}^t - stk_{r'}^{t-1}$ ) then
           $D_k^t \leftarrow D_k^t + b_{r'}^t - stk_{r'}^{t-1}$ 
           $l_{r'r'ik} \leftarrow 1$ 
          S ← S +  $c_{r'r'}$ 
          r ← r'
           $C'_i \leftarrow C'_i - \{r'\}$ 
           $C''_i \leftarrow C''_i - \{r'\}$ 
        else
           $C''_i \leftarrow C''_i - \{r'\}$ 
        end if
      until ( $C''_i = \emptyset$ )
      S ← S +  $c_{ri}$ 
       $b_i^t \leftarrow b_i^t + D_k^t$ 
      if ( $D_k^t < Q$ ) then
         $j \leftarrow \arg \min_{j \in C_i} \{q_{jt(t+1)}/l_{djtk} = 1,$ 
         $d \in J \cup \{i\} \text{ and } stk_j^t + Q - D_k^t \leq cap_j\}$ 
        S ← S +  $(Q - D_k^t) * q_{j't(t+1)}$ 
         $b_i^t \leftarrow b_i^t + Q - D_k^t$ 
      end if
    until ( $C'_i = \emptyset$ )
    t ← t + 1
  end while
end for
for each  $j \in J$  do
  S ← S +  $stk_j^0 q_{j(0)1} + stk_j^{t_{\max}} q_{j(t_{\max})t_{\max}+1}$ 
end for
return S

```

ALGORITHM 2: Routing organization and customers' inventories management.

last visited one, whose demand does not exceed the vehicle's capacity, is added to the vehicle tour; once the vehicle's capacity cannot serve more customers, it returns to the depot and another tour via another vehicle is organized in the same way. t_{\max} represents the last day of the planning period and D_k^t is the quantity that the vehicle k carries on day t , if this quantity is less than the vehicle's capacity, in order to reduce the number of tours and the vehicles used, the vehicle left its departure depot fully loaded and the added quantity is

```

Procedure: SolOrg2
Input: Problem' data values, solution cost S
Output: Depots' inventory management
for each  $i \in I/p_i = 1$  do
  t ← 2
  repeat
    S' ← 0
    L ← 0
    t' ← t - 1
    while ( $L < b_i^t$  and  $t' \neq 0$ ) do
      L ← L +  $cap_i - stk_i^{t'}$ 
      S' ← S' +  $(cap_i - stk_i^{t'})q_{it't}$ 
      t' ← t' - 1
    end while
    if ( $L \geq b_i^t$  and  $ord_i > S'$ ) then
      S ← S + S'
    end if
    t ← t + 1
  until ( $t = t_{\max}$ )
end for
for each  $i \in I/p_i = 1$  do
  t ← 1
  while ( $t < t_{\max}$ ) do
    if ( $b_i^t > stk_i^{t-1}$ ) then
       $l'_{it} \leftarrow 1$ 
      S ← S +  $ord_i$ 
    end if
    t ← t + 1
  end while
end for
return the total cost of the solution, S

```

ALGORITHM 3: Depots' inventory management.

sent to the customers, visited during the vehicle tour, whose the storage cost is the lowest and the storage capacity allows receiving this quantity.

As mentioned in Section 2, every time a depot needs to be replenished there is an ordering cost to consider. Procedure SolOrg2 (Algorithm 3) aims to reduce this cost by checking if the ordering of a depot for every period could be distributed over the periods before; if it is feasible and the storage cost is less than the ordering cost, the total cost of the solution would be reduced. In addition to organizing the vehicle routing and the replenishment schedule, these procedures try to reduce the depots' ordering costs.

3.4. Selection and Crossover. Parents are selected using the binary tournament method; the first one is obtained by selecting randomly two chromosomes among the individuals of the best half of the population and the best one is kept. The second parent is selected with the same method but from the whole population except the first selected parent.

A new chromosome "child" is obtained by crossing the selected parents; crossover for the depot sequence works as on a sequence of binary; a positive cut-off point less than the sequence length is generated randomly. The depot sequence of the child receives the part before the cut point of the first

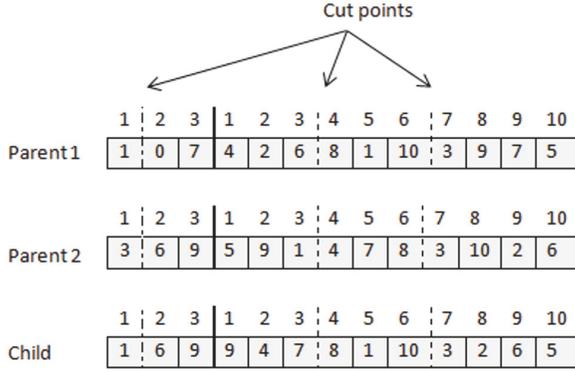


FIGURE 2: Example of chromosomes crossover.

parent and the part after the cut point of the second parent. The classical order crossover is used to obtain the customers sequence; two cut-off points less than the sequence length are generated randomly; the part between the two cut points in the first parent is copied in the child in the same order and in the same position; then the second parent is scanned from the second cut point to the end and from the beginning to the second cut point while copying each element not yet in the child. Figure 2 illustrates this operator.

3.5. Chromosome Repair. In order to check the validity of the obtained chromosome, the first step is to verify the existence of 1 in the depot sequence; this means that in the worst case all customers have been assigned to the same depot; then it shall be ensured that the average demand of customers assigned to a depot over the planning horizon does not exceed its storage capacity; this is verified by means of function Moy defined earlier.

If the average demand of customers assigned to a depot exceeds its capacity, the last assigned customer to the depot is reassigned to another depot of which capacity allows serving him. Algorithm 4 describes steps of this procedure.

3.6. Local Search. In each generation of the hybrid genetic algorithm, a local search procedure is applied with a probability p_{ls} to improve the solution obtained by the crossover operator by exploring its neighborhood; Algorithm 5 describes steps of the proposed procedure, six moves are used, three of them operate on depots, and the others operate on customers. The proposed moves are presented in the sequel.

CloseDepot. This move consists in closing a depot and reallocates its customers according to their proximity to other depots. Figure 3 illustrates this move. The steps of CloseDepot are as follows:

- Randomly select a depot i .
- Randomly select a customer j from C_i .
- Let D_j be the set of open depots able to satisfy customer j need for the planning horizon.
- Find in the set D_j the closest depot i' to j .

```

Procedure: Chromrepair
Input: A chromosome
Output: Valid chromosome
Find  $i \in I/ds(i) = 1$ , if it fails, find the first closed depot  $i'$  and  $ds(i') \leftarrow 1$ 
for each  $i \in I/ds(i) \neq 0$ 
  if ( $Moy(C_i) > cap_i$ ) then
    repeat
       $j \leftarrow$  last customer assigned to  $i$ 
       $i' \leftarrow \arg \min_{d \in I} \{d_{id} / Moy(C_d \cup \{j\}) < cap_d\}$ 
       $C_i \leftarrow C_i - \{j\}$ 
       $C_{i'} \leftarrow C_{i'} + \{j\}$ 
    if ( $\{i'\} = \emptyset$ ) then
      Find  $i'' / ds(i'') = 0$ 
       $C_i \leftarrow C_i - \{j\}$ 
       $C_{i''} \leftarrow C_{i''} + \{j\}$ 
    end if
  until ( $Moy(C_i) \leq cap_i$ )
end if
end for

```

ALGORITHM 4: Chromosome repairing procedure.

```

Procedure: LocalSearch
Input: A solution  $S$ 
Output:  $S_{best}$ 
 $S_{best} \leftarrow S$ ;  $S' \leftarrow S$ ;  $S'' \leftarrow S$ 
 $N_{CD} \leftarrow S$  neighborhood generated using CloseDepot move
 $N_{RE} \leftarrow S$  neighborhood generated using RouteExchange move
 $N_{RT} \leftarrow S$  neighborhood generated using RouteTrans move
 $S' \leftarrow \arg \min \{Cost(S) / S \in N_{CD} \cup N_{RE} \cup N_{RT}\}$ 
if ( $Cost(S') < Cost(S_{best})$ ) then
   $S'' \leftarrow S'$ 
else
   $S'' \leftarrow S_{best}$ 
end if
 $N_{RTE} \leftarrow S''$  neighborhood generated using CustExchange move
 $N_{RTT} \leftarrow S''$  neighborhood generated using CustTrans move
 $N_{RR} \leftarrow S''$  neighborhood generated using CustRev move
 $S_{min} \leftarrow \arg \min \{Cost(S) / S \in N_{RTE} \cup N_{RTT} \cup N_{RR}\}$ 
if ( $Cost(S_{min}) < Cost(S'')$ ) then
   $S_{best} \leftarrow S_{min}$ 
else
   $S_{best} \leftarrow S''$ 
end if
Return  $S_{best}$  and  $Cost(S_{best})$ 

```

ALGORITHM 5: The local search procedure.

- Assign customer j to depot i' .
- If D_j is empty, open a closed depot and assign j to it.

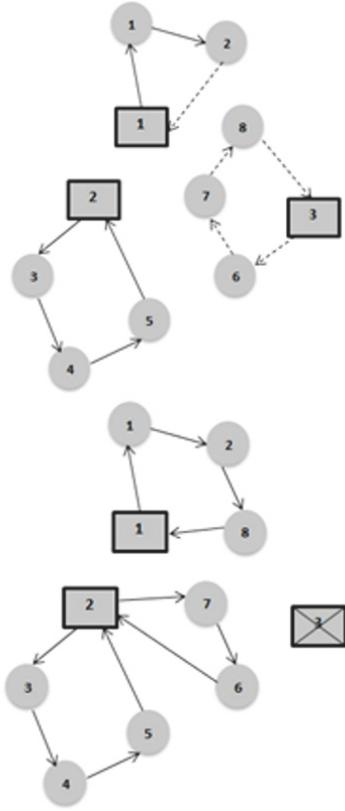


FIGURE 3: Example of a CloseDepot move.

- (g) Delete j from C_i .
- (h) If C_i is empty close depot i , else go to (b).
- (i) Build routes allocated to the concerned depots and determine the suitable inventory for both depot and customers using procedures SolOrg1 and SolOrg2.

RouteExchange. Two depots exchange customers of two routes and tours and inventories are reorganized. An example of this move is given in Figure 4; the steps of RouteExchange are as follows:

- (a) Randomly select two open depots.
- (b) Randomly select two routes, one from each selected depot.
- (c) Using function *Moy*, check if changing customers of selected routes between depots is feasible, if yes change assignation of those customers, else exit from this move.
- (d) Build routes allocated to the two depots and determine suitable inventories for both depots and customers using procedures SolOrg1 and SolOrg2.

RouteTrans. A route's customers are transferred to another depot that may be open or closed. Figure 5 presents an example of this move. The procedure of RouteTrans is as follows:

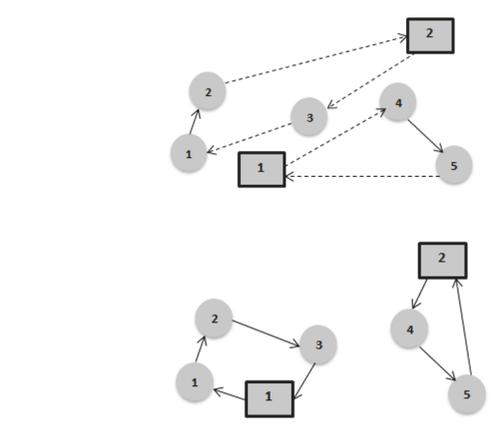


FIGURE 4: Example of a RouteExchange move.

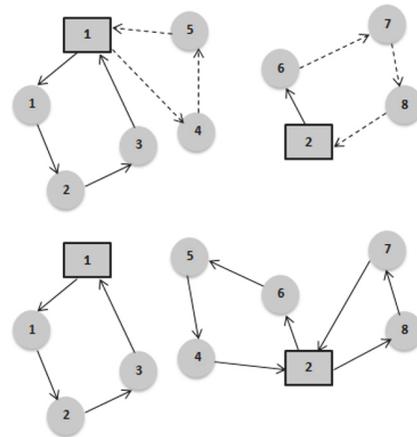


FIGURE 5: Example of a RouteTrans move.

- (a) Randomly select an opened depot.
- (b) Randomly select a route from the selected depot.
- (c) Randomly select another depot.
- (d) Assign customers of the route selected in (b) to the depot selected in (c); if the depot is not able to serve the customers, go to (c).
- (e) Build routes allocated to the two depots and determine suitable inventories for both depots and customers using procedures SolOrg1 and SolOrg2.

Remark. In the proposed moves, to check the ability of a depot to serve a set of customers, function *Moy* defined in Section 3.2 is used.

CustExchange. Two routes starting from the same depot over the same period exchange two customers. This move is shown in Figure 6; the procedure of CustExchange is as follows:

- (a) Randomly choose a period.
- (b) Randomly choose an open depot.

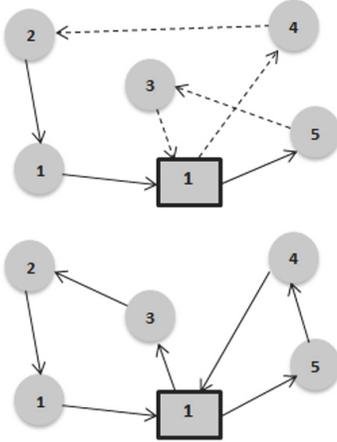


FIGURE 6: Example of a CustExchange move.

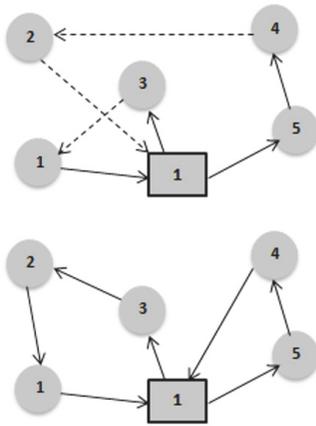


FIGURE 7: Example of a CustTrans move.

- (c) Randomly choose two routes from the selected depot.
- (d) Randomly select two customers, one in each route.
- (e) If both selected customers can be exchanged without violating capacity of each route, exchange positions of the customers; else go to (d).

CustTrans. A customer is transferred from the route to which he belongs to another one starting from the same depot over the same period. This move is illustrated in Figure 7 and its steps are as follows:

- (a) Select a period randomly.
- (b) Select an open depot randomly.
- (c) Choose randomly two routes 1 and 2 from the depot selected in (b).
- (d) Select one customer on route 1.
- (e) If capacity of route 2 allows, transfer the selected customer to it; else go to (c).

CustRev. A subsequence of customers in a route is inverted. Figure 8 illustrates this move and its steps are as follows:

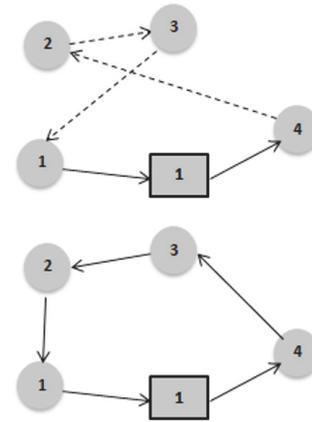


FIGURE 8: Example of a CustRev move.

- (a) Choose a period randomly.
- (b) Choose an open depot randomly.
- (c) Choose randomly a route from the selected depot in (b).
- (d) Choose randomly a subsequence of the route selected in (c) and invert the order in which customers are visited.

Function *Cost* returns the cost of a solution, i.e., the value returned by procedures *SolOrg1* and *SolOrg2*, when moves operating on depots are used. When moves operating on customers are used, the cost of a solution is modified by considering the applied change.

3.7. The Hybrid Genetic Algorithm. The overall hybrid genetic algorithm is presented in Algorithm 6, the algorithm is obtained by gathering the components previously presented, and it starts by generating an initial population which includes solutions obtained by *InitialSol* procedure as well as random solutions in order to better explore the solution space.

At each iteration, parents selected using binary tournament method are crossed to obtain a chromosome child which is repaired using *Chromrepair* procedure; if the current population already contains the obtained chromosome, the selection and the crossover are repeated; else, *LocalSearch* procedure is applied to the chromosome child to improve it; it is applied once to each offspring with a probability p_{ls} in order to avoid being trapped in a local optimum; if the obtained chromosome does not belong to the population, it replaces the worst chromosome of it. Once the maximum number of iterations is reached, the algorithm stops and returns the best solution of the last obtained population.

4. Computational Experiments

In order to test the proposed hybrid genetic algorithm, three sets of instances for the LLSRP are created, the number of customers in instances of the first and second sets is,

TABLE 1: Comparison of HGA results on LLSRP instances.

<i>Instan ces</i>	<i>BKS</i>	<i>Cost</i> _{0.3}	<i>Gap</i> (%)	<i>CPU</i> (s)	<i>Cost</i> _{0.5}	<i>Gap</i> (%)	<i>CPU</i> (s)
3 – 5 – 3	68734.4	68734.4	0	32.4	68734.4	0	38.4
3 – 5 – 5	78976.8	78976.8	0	30	78976.8	0	44.9
3 – 5 – 7	78905	78905	0	39.6	79045.8	0.18	56.3
3 – 8 – 3	85674.7	87163.4	1.74	50.4	85674.7	0	59.9
3 – 8 – 5	86785.9	86943.3	0.18	68.3	86785.9	0	70.5
3 – 8 – 7	86889.6	86889.6	0	49.4	87004.8	0.13	81.9
5 – 8 – 3	85543.1	86065.8	0.61	70.1	85543.1	0	96.8
5 – 8 – 5	86675.7	87529.5	0.98	88.2	86675.7	0	105.4
5 – 8 – 7	85548.9	86170.8	0.73	83.3	85548.9	0	140.8
5 – 12 – 3	97249.7	97249.7	0	127.5	97496	0.25	198.5
5 – 12 – 5	99082.7	99082.7	0	180.7	100165.3	1.09	210.6
5 – 12 – 7	101536.6	101734.4	0.19	178.3	101536.6	0	201.7
5 – 16 – 3	110764.1	110764.1	0	207.7	110928.3	0.15	309.6
5 – 16 – 5	130785.3	131707.1	0.7	253.6	130785.3	0	283
5 – 16 – 7	131786.9	131929.8	0.11	361.9	131786.9	0	317.4
<i>Average</i>	94329.29	94656.43	0.35	121.43	94445.9	0.12	147.71

```

Procedure: HybridGeneticAlgorithm
Input: Problem Data
Output: Best solution  $P_{best}$ 
 $t \leftarrow 0$ 
Generate initial population  $P(0)$  using InitialSol
procedure and random chromosomes
while ( $t < \text{max gen}$ ) do
  Select parents  $p_1$  and  $p_2$  using binary tournament
  from  $P(t)$ 
  Cross  $p_1$  and  $p_2$  to obtain child  $c$ 
  Repair  $c$  using Chromrepair procedure
  if ( $c \in P(t)$ ) then
    Repeat selection and crossover
  else
    Apply LocalSearch Procedure to  $c$  with a
    probability  $p_{ls}$  to obtain  $c'$ 
  end if
  if ( $c' \notin P(t)$ ) then
     $c'$  replaces the worst solution in  $P(t)$ 
  end if
   $P(t+1) \leftarrow P(t)$ 
   $t \leftarrow t+1$ 
end while
Return the best sol  $P_{best}$  in  $P(\text{max gen})$ 

```

ALGORITHM 6: The overall hybrid genetic algorithm.

respectively, 5 and 8 and the last set contains instances with 12 customers and others with 16 customers. This section reports the results obtained by the proposed algorithm. The number of potential depots $m \in \{3, 5\}$, the number of customers $n \in \{5, 8, 12, 16\}$, and the number of periods $p \in \{3, 5, 7\}$; a period refers to a week; orders are made once a week by depots and

customers. Depots opening costs and replenishment costs vary for each depot. All vehicles have the same capacity and the same cost of use. Depots, as well as customers, have different storage costs and different capacities.

After preliminary testing max gen is fixed to 500, and the population size is 100. Procedures are implemented in C language on an Acer Aspire ONE D255 1.00 GHz machine, running Windows 7 Starter Edition.

To investigate the effect of local search probability on solutions' quality and the computational time, the algorithm is tested with two local search probability values; Table 1 provides comparison between the obtained results; the first column corresponds to the names of the instances; they are labeled as n-m-p; the name of each one indicates its size. The second column corresponds, for each instance, to the best solution obtained by the hybrid genetic algorithm using two local search probability values: 0.3 and 0.5.

Columns three to eight indicate, for each of the probability values and for each instance, the obtained solution cost, the gap with BKS, and the computation time in seconds. The gap is obtained by calculating $\text{Gap} = [(Cost - BKS)/BKS] * 100$.

The algorithm found the same solutions for two instances with 3 potential depots and 5 customers. The average gap for HGA when $p_{ls} = 0.3$ is 0.35%, the average computational time is 121.43 seconds and the algorithm found 7 of the best solutions with this local search probability value. Better solutions are found when $p_{ls} = 0.5$; the average gap to BKS is 0.12%, the average computational time is 147.71 seconds, and 10 of the best solutions are found with this probability value, this can be explained by the fact that locale search procedure is applied to more chromosomes, consequently more solutions have been improved, and this also explains why the computational time is higher when $p_{ls} = 0.5$.

TABLE 2: Comparison of HGA results on each LLSRP instance size.

Instances size	$p_{ls} = 0.3$			$p_{ls} = 0.5$		
	Average cost	Average Gap (%)	Average CPU (s)	Average cost	Average Gap (%)	Average CPU (s)
1st set	75538.73	0	34.03	75585.67	0.06	46.53
2nd set	86793.73	0.71	68.28	86205.52	0.02	92.55
3rd set	112077.97	0.17	218.28	112116.4	0.25	253.47

Table 2 provides a comparison of solutions' quality for each set of instances as a well as the computational time for both local search probability values.

The algorithm found the best solutions for the first and third sets of instances with $p_{ls} = 0.3$; however the difference in the average gaps is small; their respective gaps are 0.06% and 0.8%. For the second set, the best solutions are found when $p_{ls} = 0.5$ and the difference in the average gaps is 0.69%. On average, the algorithm provides better results when $p_{ls} = 0.5$ and for all sets of instances, the average computational time when $p_{ls} = 0.5$ is higher than $p_{ls} = 0.3$.

5. Conclusions

In this paper, a mixed integer linear programming model is proposed for a multiperiod location lot-sizing routing problem in a two-echelon network; the objective is to determine the set of depots to open, allocation of customers to open depots, the vehicle routing organization, quantities to ship to depots, and those to ship from depots to customers in every period in a way that minimizes the total distribution cost and satisfies customers need. To solve this problem, a genetic algorithm combined with a local search procedure is proposed; it is tested over three sets of instances with two local search probability values. In the proposed mathematical model, customers have a deterministic demand, for future research, it would be interesting to consider uncertain demand of customers, and also, some other constraints can be considered such as time window to visit the final customers and a heterogeneous fleet of vehicles at depots.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] M. S. Daskin, C. R. Coullard, and Z. J. Shen, "An inventory-location model: formulation, solution algorithm and computational results," *Annals of Operations Research*, vol. 110, pp. 83–106, 2002.
- [2] Z.-J. M. Shen, "A multi-commodity supply chain design problem," *IIE Transactions*, vol. 37, no. 8, pp. 753–762, 2005.
- [3] L. Ozsen, C. R. Coullard, and M. S. Daskin, "Capacitated warehouse location model with risk pooling," *Naval Research Logistics (NRL)*, vol. 55, no. 4, pp. 295–312, 2008.
- [4] Z. J. Shen and L. Qi, "Incorporating inventory and routing costs in strategic location models," *European Journal of Operational Research*, vol. 179, no. 2, pp. 372–389, 2007.
- [5] P. A. Miranda and R. A. Garrido, "Valid inequalities for Lagrangian relaxation in an inventory location problem with stochastic capacity," *Transportation Research Part E: Logistics and Transportation Review*, vol. 44, no. 1, pp. 47–65, 2008.
- [6] M. T. Melo, S. Nickel, and F. Saldanha-da-Gama, "Facility location and supply chain management—a review," *European Journal of Operational Research*, vol. 196, no. 2, pp. 401–412, 2009.
- [7] S. Salhi and G. K. Rand, "The effect of ignoring routes when locating depots," *European Journal of Operational Research*, vol. 39, no. 2, pp. 150–156, 1989.
- [8] C. Prins, C. Prodhon, and R. Wolfer-Calvo, "A memetic algorithm with population management (MA/PM) for the capacitated location-routing problem," in *Evolutionary Computation in Combinatorial Optimization*, J. Gottlieb and G. Raidl, Eds., vol. 3906 of *Lecture Notes in Computer Science*, pp. 183–194, Springer, Berlin, Germany, 2006.
- [9] C. Prins, C. Prodhon, and R. Wolfer-Calvo, "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*, vol. 4, no. 3, pp. 221–238, 2006.
- [10] C. Duhamel, P. Lacomme, C. Prins, and C. Prodhon, "A GRASP×ELS approach for the capacitated location-routing problem," *Computers & Operations Research*, vol. 37, no. 11, pp. 1912–1923, 2010.
- [11] J.-M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. W. Calvo, "A branch-and-cut method for the capacitated location-routing problem," *Computers & Operations Research*, vol. 38, no. 6, pp. 931–941, 2011.
- [12] R. Baldacci, A. Mingozzi, and R. Wolfer Calvo, "An exact method for the capacitated location-routing problem," *Operations Research*, vol. 59, no. 5, pp. 1284–1296, 2011.
- [13] C.-J. Ting and C.-H. Chen, "A multiple ant colony optimization algorithm for the capacitated location routing problem," *International Journal of Production Economics*, vol. 141, no. 1, pp. 34–44, 2013.
- [14] C. Contardo, J.-F. Cordeau, and B. Gendron, "An exact algorithm based on cut-and-column generation for the capacitated location-routing problem," *INFORMS Journal on Computing*, vol. 26, no. 1, pp. 88–102, 2014.
- [15] L. Kechmane, B. Nsiri, and A. Baalal, "A hybrid particle swarm optimization algorithm for the capacitated location routing problem," *International Journal of Intelligent Computing and Cybernetics*, vol. 11, no. 1, pp. 106–120, 2018.
- [16] C. Prodhon and C. Prins, "A survey of recent research on location-routing problems," *European Journal of Operational Research*, vol. 238, no. 1, pp. 1–17, 2014.
- [17] C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza, "A branch-and-cut algorithm for a vendor-managed inventory-routing problem," *Transportation Science*, vol. 41, no. 3, pp. 382–391, 2007.

- [18] J. Oppen, A. Løkketangen, and J. Desrosiers, "Solving a rich vehicle routing and inventory problem using column generation," *Computers & Operations Research*, vol. 37, no. 7, pp. 1308–1317, 2010.
- [19] C. Archetti, L. Bertazzi, A. Hertz, and M. G. Speranza, "A hybrid heuristic for an inventory routing problem," *INFORMS Journal on Computing*, vol. 24, no. 1, pp. 101–116, 2012.
- [20] N. H. Moin, S. Salhi, and N. A. B. Aziz, "An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem," *International Journal of Production Economics*, vol. 133, no. 1, pp. 334–343, 2011.
- [21] L. C. Coelho, J.-F. Cordeau, and G. Laporte, "Consistency in multi-vehicle inventory-routing," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 270–287, 2012.
- [22] L. C. Coelho and G. Laporte, "The exact solution of several classes of inventory-routing problems," *Computers & Operations Research*, vol. 40, no. 2, pp. 558–565, 2013.
- [23] D. Adelman, "A price-directed approach to stochastic inventory/routing," *Operations Research*, vol. 52, no. 4, pp. 499–514, 2004.
- [24] L. M. Hvattum and A. Løkketangen, "Using scenario trees and progressive hedging for stochastic inventory routing problems," *Journal of Heuristics*, vol. 15, no. 6, pp. 527–557, 2009.
- [25] L. M. Hvattum, A. Løkketangen, and G. Laporte, "Scenario tree-based heuristics for stochastic inventory-routing problems," *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 268–285, 2009.
- [26] H. O. Mete and Z. B. Zabinsky, "Stochastic optimization of medical supply location and distribution in disaster management," *International Journal of Production Economics*, vol. 126, no. 1, pp. 76–84, 2010.
- [27] M. Vidovic, D. Popović, and B. Ratković, "Mixed integer and heuristics model for the inventory routing problem in fuel delivery," *International Journal of Production Economics*, vol. 147, no. C, pp. 593–604, 2014.
- [28] N. Nananukul, *Lot-sizing and inventory routing for a production-distribution supply chain [Ph.D. dissertation]*, The University of Texas, Austin, Tex, USA, 2008.
- [29] S. Cetinkaya, H. Üster, G. Easwaran, and B. B. Keskin, "An integrated outbound logistics model for frito-lay: coordinating aggregate-level production and distribution decisions," *Interfaces*, vol. 39, no. 5, pp. 460–475, 2009.
- [30] Z. Chen and B. R. Sarker, "An integrated optimal inventory lot-sizing and vehicle-routing model for a multisupplier single-assembler system with JIT delivery," *International Journal of Production Research*, vol. 52, no. 17, pp. 5086–5114, 2014.
- [31] L. Kechmane, B. Nsiri, and A. Baalal, "A mathematical model to optimize transport cost and inventory level in a single level logistic network," in *Computational Problems in Science and Engineering*, vol. 343, pp. 271–281, Springer, 2016.
- [32] P. A. Miranda and R. A. Garrido, "Inventory service-level optimization within distribution network design problem," *International Journal of Production Economics*, vol. 122, no. 1, pp. 276–285, 2009.
- [33] S. C. Liu and S. B. Lee, "A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into consideration," *The International Journal of Advanced Manufacturing Technology*, vol. 22, no. 11-12, pp. 941–950, 2003.
- [34] S. C. Liu and C. C. Lin, "A heuristic method for the combined location routing and inventory problem," *The International Journal of Advanced Manufacturing Technology*, vol. 26, no. 4, pp. 372–381, 2005.
- [35] D. Ambrosino and M. G. Scutella, "Distribution network design: new problems and related models," *European Journal of Operational Research*, vol. 165, no. 3, pp. 610–624, 2005.
- [36] A. Ahmadi-Javid and N. Azad, "Incorporating location, routing and inventory decisions in supply chain network design," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 5, pp. 582–597, 2010.
- [37] A. Hiassat and A. Diabat, "Location-inventory-routing problem with perishable products," in *Proceedings of the 41st International Conference on Computers and Industrial Engineering*, pp. 130–135, Los Angeles, Calif, USA, October 2011.
- [38] A. Ahmadi-Javid and A. H. Seddighi, "A location-routing-inventory model for designing multisource distribution networks," *Engineering Optimization*, vol. 44, no. 6, pp. 637–656, 2012.
- [39] W. J. Guerrero, C. Prodhon, N. Velasco, and C. A. Amaya, "Hybrid heuristic for the inventory location-routing problem with deterministic demand," *International Journal of Production Economics*, vol. 146, no. 1, pp. 359–370, 2013.
- [40] N. Nekooghdirli, R. Tavakkoli Moghaddam, V. R. Ghezavati, and S. Javanmard, "Solving a new bi-objective location-routing-inventory problem in a distribution network by meta-heuristics," *Computers & Industrial Engineering*, vol. 76, no. 1, pp. 204–221, 2014.
- [41] Y. Zhang, M. Y. Qi, L. X. Miao, and E. Liu, "Hybrid meta-heuristic solutions to inventory location routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 70, no. 1, pp. 305–323, 2014.
- [42] T. F. Abdelmaguid and M. M. Dessouky, "A genetic algorithm approach to the integrated inventory-distribution problem," *International Journal of Production Research*, vol. 44, no. 21, pp. 4445–4464, 2006.
- [43] F. T. S. Chan, R. K. Tibrewal, A. Prakash, and M. K. Tiwari, "A biased random key genetic algorithm approach for inventory-based multi-item lot-sizing problem," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 229, no. 1, pp. 157–171, 2015.
- [44] V. Rungtungeanaun and C. Woarawichai, "Applying genetic algorithms for inventory lot-sizing problem with supplier selection under storage space," *International Journal of Industrial and Manufacturing Engineering*, vol. 7, no. 2, pp. 213–218, 2013.
- [45] K. Sorensen and M. Sevaux, "MA/PM: memetic algorithms with population management," *Computers & Operations Research*, vol. 33, no. 5, pp. 1214–1225, 2006.
- [46] Y.-B. Park, J.-S. Yoo, and H.-S. Park, "A genetic algorithm for the vendor-managed inventory routing problem with lost sales," *Expert Systems with Applications*, vol. 53, no. 1, pp. 149–159, 2016.
- [47] Y. Marinakis and M. Marinaki, "A Bilevel Genetic Algorithm for a real life location routing problem," *International Journal of Logistics Research and Applications*, vol. 11, no. 1, pp. 49–65, 2008.
- [48] S.-H. Liao, C.-L. Hsieh, and Y.-S. Lin, "A multi-objective evolutionary optimization approach for an integrated location-inventory distribution network problem under vendor-managed inventory systems," *Annals of Operations Research*, vol. 186, pp. 213–229, 2011.
- [49] L. Kechmane, B. Nsiri, and A. Baalal, "A memetic algorithm for the capacitated location-routing problem," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, pp. 219–226, 2016.

- [50] F. Forouzanfar and R. Tavakkoli-Moghaddam, "Using a genetic algorithm to optimize the total cost for a location-routing-inventory problem in a supply chain with risk pooling," *Journal of Applied Operational Research*, vol. 4, no. 1, pp. 2–13, 2012.
- [51] Y. H. Li, H. Guo, L. Wang, and J. Fu, "A hybrid geneticsimulated annealing algorithm for the location-inventory-routing problem considering returns under esupply chain environment," *The Scientific World Journal*, vol. 2013, Article ID 125893, 10 pages, 2013.
- [52] A. Hiassat, A. Diabat, and I. Rahwan, "A genetic algorithm approach for location-inventory-routing problem with perishable products," *Journal of Manufacturing Systems*, vol. 42, pp. 93–103, 2017.

