Engineering Optimization

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/geno20

A location-routing-inventory model for designing multisource distribution networks

Amir Ahmadi-Javid [a] & Amir Hossein Seddighi [a]

[a] Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

Version of record first published: 10 Oct 2011.

Engineering Optimization
Vol. 44, No. 6, June 2012, 637–656

A location-routing-inventory model for designing multisource distribution networks

Amir Ahmadi-Javid* and Amir Hossein Seddighi

Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran

(*Received 25 March 2010; final version received 16 June 2011*)

This article studies a ternary-integration problem that incorporates location, inventory and routing decisions in designing a multisource distribution network. The objective of the problem is to minimize the total cost of location, routing and inventory. A mixed-integer programming formulation is first presented, and then a three-phase heuristic is developed to solve large-sized instances of the problem. The numerical study indicates that the proposed heuristic is both effective and efficient.

Keywords: multisource distribution networks; supply chain design; facility location; inventory control; vehicle routing; transportation; ternary integration

1. Introduction

In the last two decades, several integrated distribution network design problems have been proposed in the literature. The design of a distribution network comprises three major problems: location-allocation, vehicle routing and inventory control. Most of the research in the literature so far has been focused on the integration of two of the above problems; *i.e.* location-routing, location-inventory and inventory-routing problems. In the following, articles studying these binary-integration problems are briefly reviewed.

Location-routing problems (LRP) are among the oldest integration issues, and have been studied since the middle of the 1970s. Min *et al.* (1998) and Nagy and Salhi (2007) provided surveys on LRPs.

Inventory-routing problems (IRP) are concerned with the storage and distribution of products from facilities to a set of customers over a given planning horizon. This type of integration problem has been looked at since the early 1980s, in articles such as those surveyed by Federgruen and Simchi–Levi (1995) and Bertazzi *et al.* (2008). More recent publications on this topic include Raa and Aghezzaf (2008), Zhao *et al.* (2007), Yu *et al.* (2008) and Raa and Aghezzaf (2009).

In a location-inventory problem (LIP), decisions about stocks are incorporated into facility location. Interest in LIPs has increased in the past decade, with major articles coming from Daskin *et al.* (2002), Shen (2005), Ozsen *et al.* (2006), Shen and Qi (2007) and Miranda and Garrido (2008). It should be noted that, although inventory-routing and location-routing problems have

*Corresponding author. Email: ahmadi_javid@aut.ac.ir

ISSN 0305-215X print/ISSN 1029-0273 online

http://dx.doi.org/10.1080/0305215X.2011.600756
http://www.tandfonline.com

been examined extensively for more than 30 years and 20 years, respectively, location-inventory integration issues have been taken into account only in the last 10 years. The reason may be that the mathematical models of location-inventory are mixed-integer nonlinear programs while almost all of those developed for inventory-routing and location-routing problems are mixed-integer linear programs. In fact, due to recent advances in developing heuristic algorithms for large-scale optimization problems, complex supply-chain integration problems are now practically solvable.

One key shortcoming in most of the past research on integration is that only two of the three issues of location, routing and inventory are combined. As can be expected, however, because of the interrelation between these subjects, there is also an interest in investigating ternary integration, *i.e.* location-routing-inventory problems (LRIPs).

The first article incorporated routing costs, but not decisions, in a location-inventory problem is Shen and Qi (2007). They added an approximation of the routing cost, which depends only on location-allocation decisions, in the objective; however, their model does not determine routing decisions. Recently, for the first time, Ahmadi-Javid and Azad (2010) presented an integrated location-routing-inventory problem in a single-source supply chain. They integrate location-allocation, routing and inventory decisions, and show that their model provides a significant advantage over Shen and Qi's model, which only incorporates approximate routing cost instead of routing decisions. This article considers a ternary-integration, multisource-distribution-network design problem that optimizes location, inventory and routing decisions simultaneously. The objective is to minimize the total yearly cost associated with these decisions. The proposed problem features a multisource distribution network and a complex but more realistic routing system, neither of which are considered in Ahmadi-Javid and Azad (2010).

Since LRPs were shown by Perl and Daskin (1985) to be NP-hard, LRIPs also belong to the class of NP-hard problems, so a common way to solve such complex problems is to develop a heuristic method. To construct a heuristic, the problem is first formulated as a mixed-integer programming model. Then, a multiphase heuristic algorithm based on simulated annealing (SA) and ant colony system (ACS) is designed. The heuristic method is decomposed into three phases: location, routing-1 and routing-2. After determining a suitable initial solution, an SA algorithm and a hybrid ACS algorithm are implemented repeatedly to improve the solution in the first two phases and in the third phase, respectively.

The article is organized as follows. In Section 2, the problem is described and formulated. Section 3 presents the proposed heuristic solution method for solving the problem, and Section 4 provides wide-ranging computational results to examine its effectiveness. Finally, Section 5 concludes the article.

2. Problem description and formulation

Distribution refers to the steps taken to procure the products needed by customers, move these items from the suppliers to the distribution centres, store them and, finally, distribute them to customers. The design of a distribution network is made up of three major problems: location-allocation, vehicle routing and inventory control. This article considers a LRIP that integrates all three of the above problems to design a multisource distribution network. The assumptions, notation and formulation are presented below, and then a mixed-integer-nonlinear program is given to mathematically address the problem.

Assumptions

- Each open distribution centre can be assigned to only one supplier, and each customer's demand level is constant over one year.

- The possible capacity levels for each distribution centre are known, and the company pays a fixed location cost to open a distribution centre with a specific capacity. Moreover, the sum of distribution capacities is sufficient to fully meet customer demands.
- Each new order placed at an open distribution centre is delivered in full as the inventory reaches zero. There is a fixed cost charged for each order placed. There is also a holding cost for the inventory at each distribution centre, which maintains a cyclic inventory.
- The vehicle capacities are the same and fleet type is homogeneous. Additionally, a sufficient number of vehicles are available. Each customer is placed on exactly one vehicle route.
- The distribution system is based on a cyclic-routing pattern. The amount that each vehicle delivers to each customer is equal to the customer's demand over the time required by the vehicle to do its tour and return to the customer.
- The objective is to minimize the total yearly cost of location, purchasing the commodity, shipment, inventory and routing.

In a supply chain satisfying the above assumptions, the location-allocation decisions are interrelated with inventory and vehicle-routing decisions. Location decisions affect the cost of shipping which must be paid each time an order is placed at each distribution centre, as well as the cost of establishing the routes needed to serve the customers. Moreover, the allocation of customers to open distribution centres significantly impacts the optimal order sizes placed at the open distribution centres, as well as each centre's routing system. The dependence among location-allocation, inventory and routing leads one to establish a model that integrates the three decisions.

Index sets

K The set of customers.
J The set of potential distribution centres.
S The set of suppliers.
N The set of capacity levels available to the potential distribution centres.
V The set of vehicles.
M The merged set of customers and potential distribution centres, *i.e.* $K \cup J$.

Parameters

d_k The yearly demand of customer k ($\forall k \in K$).
f_j^n The yearly fixed cost for opening and operating distribution centre j with capacity level n ($\forall j \in J, \forall n \in N$).
c_j^n The yearly capacity associated with capacity level n for potential distribution centre j ($\forall j \in J, \forall n \in N$).
tc_{kl} The transportation cost between node k and node l ($\forall k, l \in M$).
tt_{kl} The transportation time (in years) between node k and node l ($\forall k, l \in M$).
vc The capacity of each vehicle for each tour.
h_j The inventory holding cost per product unit per year at distribution centre j ($\forall j \in J$).
p_{sj} The fixed cost for placing an order from distribution centre j to supplier s ($\forall j \in J, \forall s \in S$).
a_{sj} The cost of shipping one unit from supplier s to distribution centre j ($\forall j \in J, \forall s \in S$).
e_{sj} The cost of purchasing one unit from supplier s via distribution centre j ($\forall j \in J, \forall s \in S$).
g_s The yearly capacity of supplier s ($\forall s \in S$).

Decision variables

$$R_{klv} = \begin{cases} 1 & \text{if } k \text{ precedes } l \text{ in the route of vehicle } v \\ 0 & \text{otherwise} \end{cases} \quad (\forall k, l \in M, \forall v \in V).$$

$$Y_{jk} = \begin{cases} 1 & \text{if customer } k \text{ is assigned to distribution centre } j \\ 0 & \text{otherwise} \end{cases} \quad (\forall k \in K, \forall j \in J).$$

$$Z_{sj} = \begin{cases} 1 & \text{if distribution centre } j \text{ is assigned to supplier } s \\ 0 & \text{otherwise} \end{cases} \quad (\forall j \in J, \forall s \in S).$$

$$U_j^n = \begin{cases} 1 & \text{if distribution centre } j \text{ is opened with capacity level } n \\ 0 & \text{otherwise} \end{cases} \quad (\forall j \in J, \forall n \in N).$$

Q_j A positive real variable determining the order size at distribution centre j ($\forall j \in J$).
M_{kv} An auxiliary non-negative variable used for sub-tour elimination in the route of vehicle v ($\forall k \in K, \forall v \in V$).

Objective function

The objective function includes the following yearly costs:

(1) *Location cost*. The yearly fixed cost of locating the open distribution centres:

$$\sum_{j \in J} \sum_{n \in N} f_j^n U_j^n$$

(2) *Routing cost*. The annual cost of routing from the open distribution centres to the customers is given in the formula

$$\sum_{v \in V} \left\{ \frac{1}{\sum_{k \in M} \sum_{l \in M} tt_{kl} R_{klv}} \left(\sum_{k \in M} \sum_{l \in M} tc_{kl} R_{klv} \right) \right\},$$

in which the convention $0/0 = 0$ is considered. The term $\sum_{k \in M} \sum_{l \in M} tt_{kl} R_{klv}$ indicates the amount of time spent on the route by vehicle v, and $\sum_{k \in M} \sum_{l \in M} tc_{kl} R_{klv}$ is the total cost incurred by vehicle v for doing its route once. The term

$$\frac{1}{\sum_{k \in M} \sum_{l \in M} tt_{kl} R_{klv}} \left(\sum_{k \in M} \sum_{l \in M} tc_{kl} R_{klv} \right)$$

is therefore the yearly routing cost incurred by vehicle v.

(3) *Inventory cost*. The yearly inventory cost includes the fixed costs of purchasing, shipment, placing orders and the holding cost of cycle inventories. Thus, at distribution centre j, this cost is calculated as

$$\sum_{s \in S} e_{sj} Z_{sj} D_j + \sum_{s \in S} a_{sj} Z_{sj} D_j + \sum_{s \in S} p_{sj} Z_{sj} \frac{D_j}{Q_j} + h_j \frac{Q_j}{2},$$

where D_j denotes the total annual demand going through distribution centre j, *i.e.* $D_j = \sum_k d_k Y_{jk}$.

Formulation

The problem formulation is as follows:

$$\min: \quad \sum_{j\in J}\sum_{n\in N} f_j^n U_j^n + \sum_{v\in V}\left\{\frac{\sum_{k\in M}\sum_{l\in M} tc_{kl}R_{klv}}{\sum_{k\in M}\sum_{l\in M} tt_{kl}R_{klv}}\right\}$$
$$+\sum_{j\in J}\left[\sum_{s\in S} p_{sj}Z_{sj}\frac{\sum_{k\in K} d_k Y_{jk}}{Q_j} + \sum_{s\in S}(a_{sj}+e_{sj})Z_{sj}\sum_{k\in K} d_k Y_{jk} + \frac{h_j Q_j}{2}\right] \tag{1}$$

subject to

$$\sum_{v\in V}\sum_{l\in M} R_{klv} = 1 \quad \forall k \in K \tag{2}$$

$$\left(\sum_{k\in M}\sum_{l\in M} tt_{kl}R_{klv}\right)\times\left(\sum_{l\in K} d_l \sum_{k\in M} R_{klv}\right) \le vc \quad \forall v \in V \tag{3}$$

$$M_{kv} - M_{lv} + |K| \times R_{klv} \le |K| - 1 \quad \forall k, l \in K, \forall v \in V \tag{4}$$

$$\sum_{l\in M} R_{klv} - \sum_{l\in M} R_{lkv} = 0 \quad \forall k \in M, \forall v \in V \tag{5}$$

$$\sum_{j\in J}\sum_{k\in K} R_{jkv} \le 1 \quad \forall v \in V \tag{6}$$

$$\sum_{l\in M} R_{klv} + \sum_{l\in M} R_{jlv} - Y_{jk} \le 1 \quad \forall j \in J, \forall k \in K, \forall v \in V \tag{7}$$

$$\sum_{n\in N} U_j^n \le 1, \quad \forall j \in J \tag{8}$$

$$\sum_{k\in K} d_k Y_{jk} \le \sum_{n\in N} c_j^n U_j^n \quad \forall j \in J \tag{9}$$

$$\sum_{s\in S} Z_{sj} = \sum_{n\in N} U_j^n \; \forall j \in J \tag{10}$$

$$\sum_{j\in J}\sum_{k\in K} d_k Y_{jk} Z_{sj} \le g_s \quad \forall s \in S \tag{11}$$

$$\begin{aligned} &Y_{jk} \in \{0,1\} \quad \forall j \in J, \forall k \in K \\ &U_j^n \in \{0,1\} \quad \forall j \in J, \forall n \in N \\ &R_{klv} \in \{0,1\} \quad \forall k, l \in M, \forall v \in V \\ &Z_{sj} \in \{0,1\} \quad \forall j \in J, \forall s \in S \end{aligned} \tag{12}$$

$$M_{kv} \ge 0 \quad \forall k \in K, \forall v \in V \tag{13}$$

$$Q_j > 0 \quad \forall j \in J. \tag{14}$$

The model minimizes the total yearly cost of location, routing and inventory. Constraints (2) make sure that each customer is placed on exactly one vehicle route. Constraints (3) are the vehicle-capacity constraints ensuring that no customer faces any shortage over the time the vehicle needs to go through its tour. Constraints (4) are the sub-tour elimination constraints. Constraints (5) are flow-conservation constraints ensuring that whenever a vehicle enters a customer or a distribution-centre node, it must leave it again, so that the routes remain circular. Constraints (6) imply that only one open distribution centre is included in each route. Constraints (7) link the allocation and routing components: customer k is assigned to distribution centre j if vehicle v, which visits customer k, starts its trip from distribution centre j. Constraints (8) ensure that each distribution centre can be assigned to at most one capacity level. Constraints (9) are the capacity constraints associated with the distribution centres. Constraints (10) ensure that each open distribution centre will be assigned to exactly one supplier. Constraints (11) are the capacity constraints associated with the suppliers.

In Constraint set (3), the term

$$\left(\sum_{k\in M}\sum_{l\in M} tt_{kl}R_{klv}\right)\times\left(\sum_{l\in K} d_l \sum_{k\in M} R_{klv}\right)$$

is equal to the total demand of the route served by vehicle v. Note that if customer l is allocated to the route of vehicle v, that vehicle will then, in each turn, deliver

$$\left(\sum_{k\in M}\sum_{l\in M} tt_{kl}R_{klv}\right)\times\left(d_l \sum_{k\in M} R_{klv}\right)$$

units to customer l. This quantity is exactly equal to the demand of customer during the time it takes for the vehicle to do its tour and once again return to the customer.

Note that Constraints (4) are based on the Miller, Tucker and Zemlin (MTZ) sub-tour-elimination constraints developed for uncapacitated vehicle-routing problems (Miller *et al.* 1960). The MTZ constraints can be adapted for operational capacitated-vehicle-routing problems (see, for example, Kulkarni and Bhave 1985), such that they act simultaneously as both sub-tour-elimination and capacity constraints, but here due to the complexity of capacity constraints (3), designed for a cyclic capacitated-vehicle-routing system, sub-tour-elimination and capacity constraints (3) and (4) need to be considered separately.

3. Solution approach

This section proposes a heuristic algorithm to solve the complex model presented in the previous section. To do this, first the continuous decision variables Q_j are eliminated from the model, and then a heuristic based on the resulting model is developed.

In the model (1)–(14), the decision variable Q_j appears only in the objective function. Moreover, since the objective function is convex in Q_j, the optimal value of Q_j can be obtained by taking the derivative of the objective function with respect to Q_j as

$$Q_j^* = \sqrt{\frac{2(\sum_{s\in S} p_{sj}Z_{sj})(\sum_{k\in K} d_k Y_{jk})}{h_j}}. \tag{15}$$

Note that the quantity Q_j^* in Equation (15) is the same as the economic order quantity (EOQ), which is traditionally used in the classic inventory control theory. By substituting (15) in objective function (1), Q_j is eliminated, and then the new model is obtained as follows:

$$\min: \quad \sum_{j\in J}\sum_{n\in N} f_j^n U_j^n + \sum_{v\in V}\left\{\frac{\sum_{k\in M}\sum_{l\in M} tc_{kl} R_{klv}}{\sum_{k\in M}\sum_{l\in M} tt_{kl} R_{klv}}\right\}$$
$$+\sum_{j\in J}\left[\sqrt{2h_j \sum_{s\in S} p_{sj} Z_{sj} \sum_{k\in K} d_k Y_{jk}} + \sum_{s\in S}\left(a_{sj}+e_{sj}\right) Z_{sj}\sum_{k\in K} d_k Y_{jk}\right] \tag{16}$$

subject to (2)–(13).

After eliminating continuous variables Q_j, a heuristic method can be designed based on simulated annealing (SA) and ant colony system (ACS) to solve the model (16), (2)–(13). The heuristic method consists of three phases: location, routing-1 and routing-2. In the heuristic method, first an initial solution is constructed, and then an SA algorithm is used to improve the current solution iteratively in the first two phases. Additionally, a hybrid ACS algorithm is implemented in the last phase. The heuristic procedure is shown in Figure 1. The stopping criterion terminates the heuristic when the mean of the relative improvements over the last two iterations is less than 2%.

Sections 3.1 and 3.2 present two algorithms: the *nearest-neighbor* and *supplier-distribution-allocation* algorithms, which will be used in the next three subsections. Sections 3.3, 3.4 and 3.5 describe the main components of the proposed heuristic, which are also summarized in Figure 1.

3.1. *Nearest-neighbour algorithm*

The nearest-neighbour algorithm is a heuristic for quickly building a route for the customers assigned to an open distribution centre. The steps of the algorithm for open distribution centre j are as follows:

Step 1: Put all of the customers assigned to distribution centre j into set W_j and set $LV = j$. Consider LV to be the starting point of an idle vehicle's route.

Step 2: While W_j is not empty, repeat the following steps:

- 2.1 Select a customer from W_j, for example k, such that it is the closest to LV .
- 2.2 Suppose that customer k is assigned to the vehicle and is located after LV in the route. Let *RD* and *RT* respectively be the total annual demand of customers assigned to the current vehicle (route) and the amount of time needed for the vehicle to do its route once.

  If $RD \times RT \leq vc$ (here the vehicle-capacity constraint given in (3) is checked), then

  - 2.2.1 put customer k after LV in the route and set $LV = k$,
  - 2.2.2 delete customer k from W_j,

  else,

  - 2.2.3 the current route is complete; therefore, begin to make a new route and set $LV = j$.

This algorithm is a simple and fast heuristic, which is frequently applied during the heuristic's run time. It is chosen due to its speed in constructing primary routes. To further improve the total cost by modifying the routes, a powerful routing heuristic is used in the third phase (see Section 3.5).

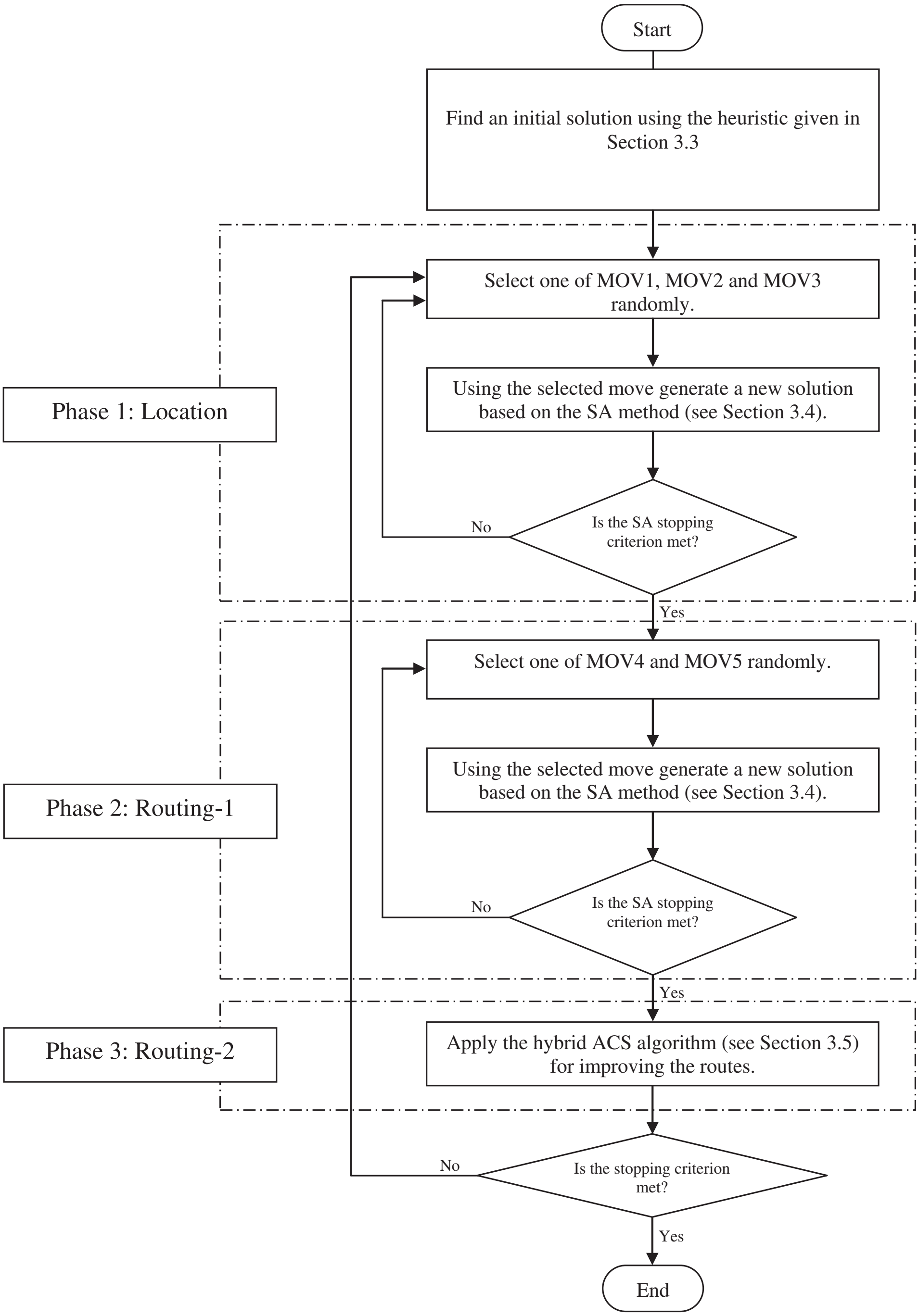

Figure 1. The flowchart of the proposed heuristic.

3.2. *Allocation of distribution centres to suppliers*

To assign each open distribution centre to the suppliers, a heuristic algorithm called *supplier-distribution allocation* (SDA) is used. The steps of this heuristic are as follows:

Step 1:
- 1.1 Put all open distribution centres into set KO.
- 1.2 Put all suppliers into the set S.

Step 2: Randomly select a distribution centre, for example j, from KO, and set $S_j = S$.

Step 3: For each supplier s in S_j, calculate

$$o_{sj} = \sqrt{2h_j p_{sj} D_j} + (a_{sj} + e_{sj})D_j.$$

Step 4: Select a supplier, for example t, from S_j that has the minimum value of o_{tj}.

Step 5:
- 5.1 If the remaining capacity of supplier t is greater than D_j, then assign distribution centre j to supplier t, delete distribution centre j from KO, update the capacity of supplier t, and go to Step 6.
- 5.2 If the remaining capacity of supplier t is equal to D_j, then assign distribution centre j to supplier t, delete distribution centre j from KO and supplier t from S, and then go to Step 6.
- 5.3 If the remaining capacity of supplier t is less than D_j, delete supplier t from S_j and go to Step 4.

Step 6: Is KO empty? If yes, stop; otherwise, go to Step 2.

3.3. *Initial solution construction*

This section presents a heuristic to find a suitable initial solution. This heuristic is based on the one presented in Yu *et al.* (2010). The steps of the heuristic are as follows:

Step 1: Let U be the set of closed distribution centres, and V be the set of unassigned customers.

Step 2: Compute q_j for each distribution centre j in U as follows. Set initially $q_j = 0$. For each unassigned customer in V , for example k, add one unit to q_j if distribution centre j is the nearest distribution centre to customer k in set U.

Step 3: Choose the distribution centre in U with the highest q value. If there are several distribution centres with equal q values, select the distribution centre with the largest minimum capacity.

Step 4: Open the chosen distribution centre and remove it from U, and randomly choose a capacity level for it.

Step 5: Arrange all unassigned customers on the basis of their distance to the chosen distribution centre. Then assign the arranged customers to the chosen distribution centre one by one until the remaining capacity of the open distribution centres is not sufficient to serve unassigned customers. Delete the assigned customers from set V .

Step 6: If there are unassigned customers, *i.e.* if V is not empty, go to Step 2; otherwise, go to Step 7.

Step 7: Build the routes of each open distribution centre using the HACS algorithm (see Section 3.5).

Step 8: Using the SDA heuristic (see Section 3.2), assign each open distribution centre to the suppliers.

Note that the yearly-capacity constraints (9) and the vehicle-capacity constraints (3) are checked in Step 5 and Step 7, respectively.

3.4. *Simulated annealing algorithm: Location and routing-1 phases*

In the first two phases of the proposed heuristic, a simulated annealing (SA) algorithm is used. SA was initially presented by Kirkpatrick *et al.* (1983). This method, like other meta-heuristic methods, attempts to solve hard combinational optimization problems through a controlled randomization. SA is easy to use and provides good solutions to real-world problems, making it one of the most powerful and popular heuristics for solving optimization problems. It uses a stochastic approach to direct the search, which is allowed to proceed to the neighbouring state even if the move causes the value of the objective function to worsen. This important feature prevents the algorithm from falling into local optimum traps. Now let us introduce the parameters used in the SA algorithm as follows:

IT_0	The initial temperature.
T	The current temperature.
K	A predetermined constant (Boltzmann constant).
CS	The decreasing rate of the current temperature (cooling schedule).
FT	The freezing temperature (the temperature at which the desired energy level is reached).
MNT	The maximum number of accepted solutions at each temperature.
nt	The counter for the number of accepted solutions at each temperature.
X_0	The initial solution.
X	The current solution in each iteration.
X_{nh}	A solution which be selected in the neighbourhood of X in each iteration.
X_{best}	The best solution obtained by the algorithm.
$C(X)$	The objective function value for solution X.
ΔC	The difference in the objective values of the current and new solutions.

An SA algorithm typically guides the original local search method in the following way. The algorithm starts with an initial solution for the problem. In the inner cycle of the SA algorithm, which will be repeated while $nt < MNT$, a neighbouring solution X_{nh} of the current solution X is generated by a suitable move. If the move improves the objective function or leaves it unchanged, then the move is always accepted. Moves which worsen the objective function value are accepted with a probability $e^{-\Delta C/KT}$ to allow the search to escape from local optima. The value of the temperature decreases in each iteration of the algorithm's outer cycle. Obviously the probability of accepting worsening solutions decreases as the temperature decreases in each outer cycle.

The pseudo code of the proposed SA algorithm is given in Figure 2. In the two following subsections, moves employed by the SA algorithm to generate neighbouring solutions in the first two phases are described.

3.4.1. *Location phase*

In this phase, the current solution is improved by suitably changing the locations and capacities of open distribution centres, and the allocation of the customers to the open distribution centres. The solution obtained in this phase is used as an input to the next phase. Note that the heuristic algorithm uses objective function (16) that is obtained from the original objective function (1) through the substitution of the best order sizes (15).

In this phase, three different types of moves: Mov1, Mov2 and Mov3, are applied. A candidate move is randomly selected from Mov1, Mov2 and Mov3 to obtain a new solution X_{nh} in the neighbourhood of the current solution X.

```
Take the initial solution X_0
  X_best = X_0, X = X_0, T = IT_0
      While (T > FT) do
         nt = 0
         While (nt < MNT) do
            Select a move randomly from the available moves
            (Mov1, Mov2 and Mov3 for location phase
            and Mov4 and Mov5 for routing-1 phase)
            Using the selected move, generate solution X_nh in the neighbourhood of X
            ΔC = C(X_nh) − C(X)
            If ΔC ≤ 0 then
               X = X_nh
               nt = nt + 1
               If C(X_nh) < C(X_best) then
                  X_best = X_nh
               End If
            Else
               Generate η from U(0,1) randomly
               Set λ = e^(−ΔC/KT)
               If η < λ then
                  X = X_nh
                  nt = nt + 1
               End If
            End If
         End While
         T = CS × T
      End While
```

Figure 2. The SA algorithm applied in the first two phases of the proposed heuristic.

Mov1: In this move, one of the open distribution centres j is closed by the roulette-wheel method (see Goldberg 1989). Then, each one of its customers is re-allocated among the other open distribution centres by the roulette-wheel method. If the remaining capacities of the other open distribution centres are not enough to satisfy the customers of distribution centre j, an open distribution centre is selected by the roulette-wheel method and its capacity level is increased in order to serve all the customers of distribution centre j. Then the nearest-neighbour algorithm is

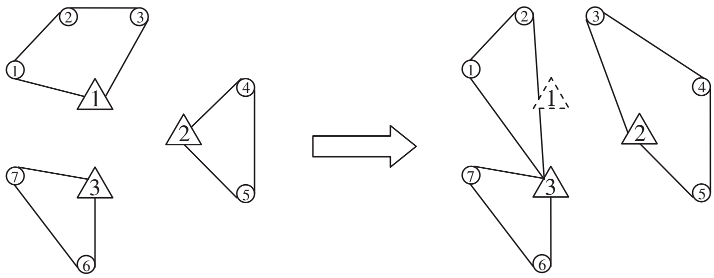

Figure 3. Illustration of Mov1.

used to rebuild the routes for all open distribution centres to which the customers of distribution centre *j* have been assigned. Finally, the SDA heuristic is used to reassign each open distribution centre to a supplier. Figure 3 presents an example of how this move builds a neighbouring solution. Note that, in Mov1 and Mov4, to simplify the description of the steps, it is assumed that, for any distribution centre, the difference between two consecutive capacity levels is larger than the demand of each customer. This assumption can be removed by adding extra steps. The steps of Mov1 are as follows:

Step 1: For each open distribution centre, compute the ratio: capacity/(used capacity).
Step 2: Select an open distribution centre, for example *j*, by the roulette-wheel method, using the ratios obtained in Step 1. Let W_j be the set of customers assigned to distribution centre *j*.
Step 3: Randomly select a customer *k* from W_j.
Step 4: Let O_k be the set of all open distribution centres that have sufficient remaining capacity to serve the demand of customer *k*.
Step 5: If O_k is empty (there is no open distribution centre that has enough unexploited capacity), then go to Step 6; otherwise, go to Step 11.
Step 6: Let T_k be the set of any open distribution centre whose capacity is not set as its highest possible capacity level.
Step 7: If T_k is empty, exit from this move; otherwise, go to Step 8.
Step 8: For each distribution centre in T_k, compute the ratio: 1/(distance between customer *k* and the distribution centre).
Step 9: Select one of the open distribution centres from T_k by the roulette-wheel method, based on the ratios obtained in Step 8.
Step 10: Increase the capacity of the selected open distribution centre to a one higher level, assign customer *k* to this distribution centre and go to Step 13.
Step 11: Compute the inverse ratio of distance between customer *k* and each distribution centre from O_k.
Step 12: Select a distribution centre from O_k by the roulette-wheel method, based on the ratios computed in Step 11, and assign customer *k* to it.
Step 13: Delete customer *k* from W_j.
Step 14: Is W_j empty? If yes, close distribution centre *j* and go to Step 15, or else go to Step 3.
Step 15: Rebuild routes of the open distribution centres to which the customers of distribution centre *j* are assigned, by using the nearest-neighbour algorithm.
Step 16: By using the SDA heuristic, reassign all open distribution centres to which the customers of distribution centre *j* are reassigned to suppliers.

If Mov1 is frequently unsuccessful at generating a neighbourhood solution, it will be not used in the location phase anymore. To this end, the input parameter *max-Mov*1 is considered as the maximum number of times that this move is allowed to fail.

Mov2: The procedure of this move is as follows. Two open distribution centres *i* and *j* are randomly selected and their customers are changed. The nearest-neighbour algorithm is exploited for distribution centres *i* and *j* to build the required routes. Finally, the SDA heuristic is used to reassign each of distribution centres *i* and *j* to suppliers. In this move, the capacities of distribution centres *i* and *j* are adjusted for serving the new sets of assigned customers. This move is illustrated in Figure 4. The steps of Mov2 are as follows:

Step 1: Randomly select two open distribution centres *i* and *j*.
Step 2: Change the customers of distribution centres *i* and *j*, and adjust the capacities of *i* and *j* to serve the new sets of assigned customers. If it is not possible to serve the newly assigned customers, exit from this move.
Step 3: Build the routes of customers allocated to distribution centres *i* and *j*.
Step 4: Using the SDA heuristic, reassign each of the open distribution centres *i* and *j* to a supplier.

Mov3: In this move, one of the open distribution centres is closed, and a closed distribution centre is opened by the roulette-wheel method. An example for this move is given in Figure 5. The steps of Mov3 are as follows:

Step 1: For each open distribution centre, compute the ratio: capacity/(used capacity).
Step 2: By the roulette-wheel method based on ratios computed in Step 1, select an open distribution centre and close it.
Step 3: For each closed distribution centre, compute the ratio: capacity/(cost of opening).
Step 4: Select a closed distribution centre by the roulette-wheel method using the ratios given in Step 3, and open it.
Step 5: Select a suitable capacity level for the newly opened distribution centre such that it satisfies the demands of the customers of the newly closed distribution centre. If it is not possible to serve the newly assigned customers, exit from this move.

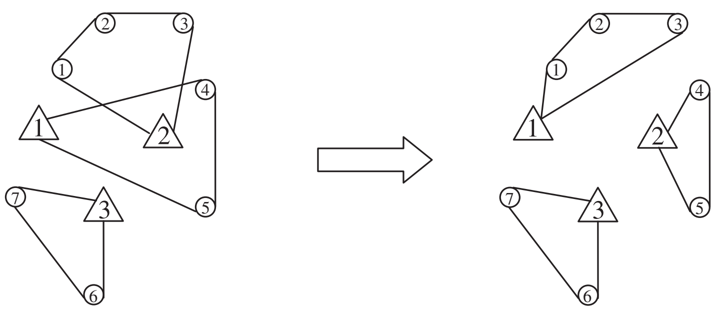

Figure 4. Illustration of Mov2.

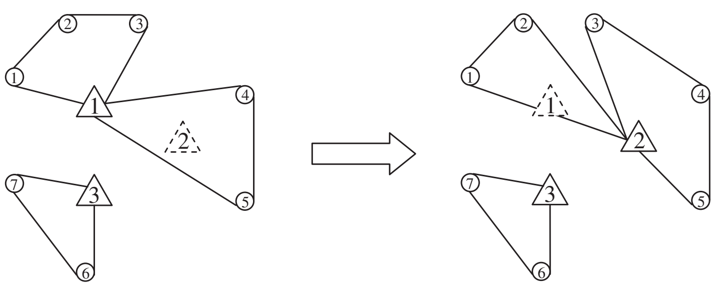

Figure 5. Illustration of Mov3.

Step 6: Assign all of the customers corresponding to the eliminated distribution centre to the newly opened distribution centre.
Step 7: Rebuild routes of the newly opened distribution centre by using the nearest-neighbour algorithm.
Step 8: By using the SDA heuristic, assign the newly opened distribution centre to a supplier.

Note that, in the case that the selected move is unsuccessful to generate a solution, a new move is randomly selected until a new solution is generated. This process is repeated until $T < FT$ at this phase. The solution obtained at the end of this phase is an input to the routing-1 phase.

3.4.2. *Routing-1 phase*

The main role of this phase is to improve the current solution by rebuilding the routes. In this phase, two different types of moves: Mov4 and Mov5, are applied. A candidate move from Mov4 and Mov5 are randomly selected to obtain a new solution X_{nh} in the neighbourhood of the current solution X.

Mov4: In this move, one of the routes is selected randomly and one of its customers is chosen randomly. Then this customer is inserted into another randomly selected route. If the remaining capacity of the associated distribution centre is not enough for the new customer, its capacity level will be increased to satisfy the customer's demand. Afterwards, the best location is determined to insert the new customer in the route without any change in the other customers' order. Finally, the SDA heuristic is used to reassign the distribution centres to suppliers. This move is shown in Figure 6. The procedure of Mov4 is as follows:

Step 1: Select two routes randomly, for example v_i and w_j, then determine the corresponding distribution centres, *i.e.* i and j.
Step 2: Randomly select one customer, for example, k, in route v_i.
Step 3: Is $i = j$? If yes, go to Step 4; otherwise, go to Step 5.
Step 4: Is $v_i = w_j$? If yes, go to Step 1; otherwise, go to Step 6.
Step 5: Adjust the capacity level of j according to the demand of customer k. If it is not possible to serve the demand of customer k, exit from this move.
Step 6: Select the best location for customer k in route w_j without changing the order of its customers.
Step 7: Is $i = j$? If yes, stop; otherwise, go to Step 8.
Step 8: By using the SDA heuristic, reassign open distribution centres i and j to suppliers.

Mov5: This move selects two routes randomly, for example v_i and w_j; then randomly selects a customers in each route, for example k_i and k_j, in routes v_i and w_j, respectively; and, changes the routes of customers k_i and k_j. After that, it adjusts the capacities of the distribution centres

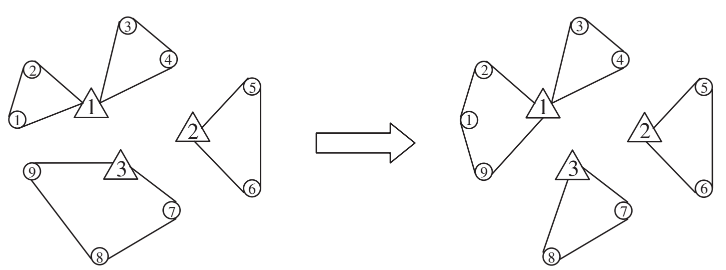

Figure 6. Illustration of Mov4.

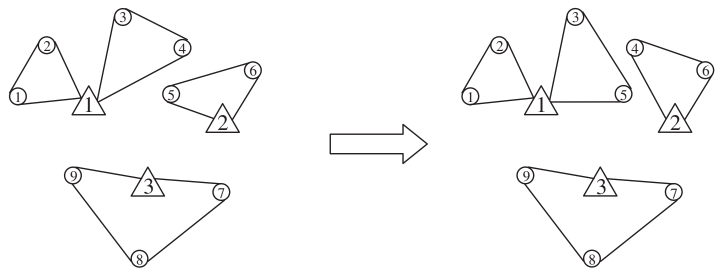

Figure 7. Illustration of Mov5.

corresponding to routes v_i and w_j to satisfy the demands of customers k_j and k_i, respectively. Then it determines the best locations of customers k_j and k_i in their new routes without changing the order of the other customers. The SDA heuristic is used to reassign the open distribution centres corresponding to routes v_i and w_j to suppliers. This move is shown in Figure 7 and its steps are as follows:

Step 1: Select two routes randomly, for example v_i and w_j, then determine the corresponding distribution centres, *i.e.* i and j.
Step 2: Randomly select two customers, for example, k_i and k_j, in routes v_i and w_j, respectively.
Step 3: Is $i = j$? If yes, go to Step 6; otherwise, go to Step 4.
Step 4: Adjust the capacities of the distribution centres i and j to satisfy the demands of customers k_j and k_i, respectively. If there is no possibility of satisfying the demands of customers k_j and k_i, exit from this move.
Step 5: By using the SDA heuristic, reassign the open distribution centres i and j to suppliers.
Step 6: Select the best locations for customers k_i and k_j in the new routes, without making changes to the other customers.

As in the previous phase, in the case that the selected move is unsuccessful at generating a solution, a new move is randomly selected until a new solution is generated. This process is repeated until $T < FT$. The solution obtained at the end of this phase is an input for the routing-2 phase (see Section 3.5), whose purpose is to further improve the total cost by modifying routes.

3.5. *Hybrid ant colony system algorithm: Routing-2 phase*

In the final phase of the proposed heuristic, an ant colony system (ACS) algorithm is used to improve the vehicle routes of each open distribution centre. ACS was introduced by Dorigo and Gambardella (1997) to solve the classic TSP problem. This algorithm is a swarm-intelligence method based on the foraging behaviour of ants.

The problem of building the routes of each open distribution centre and its associated customers is considered as a capacitated-vehicle-routing problem. Then, a hybrid ACS (HACS) algorithm is used to find the near-optimal routing. The HACS algorithm is a hybridization of the ACS algorithm given by Dorigo and Gambardella (1997) and the idea used in the saving heuristic. For further details on the HACS algorithm, see El Hassani *et al.* (2008). Note that in this algorithm, vehicle capacities are checked in a similar way to that explained in Step 2.2 of the nearest-neighbour algorithm (see Section 3.1).

After improving the routes of each open distribution centre in this phase, using the HACS algorithm, the stopping criterion is checked. If the criterion is not matched, the solution is an input for the location phase.

4. Computational results

The computational experiments presented in this section are conducted to evaluate the performance of the proposed heuristic algorithm with respect to a series of test problems. The heuristic method is coded using Visual C++ 6 and run on a Pentium 4 with a 2.7 GHz processor.

The test problems are generated as follows. Demands are drawn from a uniform distribution between 400 and 1500. Four possible capacity levels are considered for each potential distribution centre. The different capacities of potential distribution centre j are defined as follows:

$$c_j^1 = cap(j), c_j^2 = 1.5 \times cap(j), c_j^3 = 2 \times cap(j), c_j^4 = 2.5 \times cap(j)$$

where

$$cap(j) = \left[\alpha_j \times \frac{D}{|J|} \right],$$

D is the total demand of the distribution network and α_j is a random number between 0.8 and 1.2. Furthermore, the fixed setup costs for potential distribution centre j are computed as follows:

$$f_j^1 = [0.65 \times k_j'], f_j^2 = [0.9 \times k_j'], f_j^3 = [1.1 \times k_j'], f_j^4 = [1.35 \times k_j'],$$

where k'_j is drawn from a uniform distribution between 200 and 400. In addition, consider the following settings:

h_j is uniformly drawn from [0.001, 0.002];
p_{sj} is uniformly drawn from [0.02, 0.03];
a_{sj} is uniformly drawn from [0.003, .005];
e_{sj} is uniformly drawn from [0.025, 0.03].

The vehicle capacity is set at 200 and the capacity of each supplier is computed as $g_s = 4 \times \left[r_s \times \frac{D}{|S|} \right]$, in which r_s is a random number between 0.8 and 1.2 for each supplier.

The coordinates of the customer and distribution-centre locations are drawn from a uniform distribution over [0, 1000] (in kilometres). If ed_{kl} denotes the Euclidian distance between k and l, then

$$tc_{kl} = ed_{kl} \times r_{kl}^1$$

$$tt_{kl} = \frac{ed_{kl}}{24 \times 365 \times r_{kl}^2}$$

where r_{kl}^1 (transportation cost per kilometre) and r_{kl}^2 (speed of a vehicle in kilometres per hour) are drawn from uniform distributions over [0.0005, 0.001] and [60, 70], respectively.

The heuristic algorithm is run 20 times for each instance, and the average objective value is reported in all of the tables. Since the algorithm parameters significantly affect the performance of the heuristic algorithm, the tuning of the SA algorithm parameters is done in each instance by carrying out a full factorial experiment designed for three SA parameters, $90 \leq IT_0 \leq 150$, $0.85 \leq CS \leq 0.95$, $8 \leq FT \leq 12$, at five levels and with 20 random replications. In the coming tables, the CPU times are in seconds, and the abbreviation DC stands for distribution centre.

4.1. *Applying the heuristic to an LRP benchmark*

To verify the performance of the proposed heuristic, it is applied to the LRP benchmark provided by Barreto (2004). This benchmark includes the 15 test problems with their lower bounds for a capacitated location-routing problem. The capacitated LRP considered in Barreto (2004) can be solved by the proposed heuristic with a little adaptation; therefore, one can apply the adapted heuristic to the benchmark to assess its performance. The results are reported in Table 1. From this table, it can be seen that the solutions found by the heuristic are optimal for seven instances, and that, for the others, the maximum gap is less than 6.5%. This shows that the quality of the solutions obtained by the heuristic is promising.

4.2. *Comparison of the heuristic with an algorithm adapted from the literature*

This subsection compares the performance of the proposed heuristic with that of the heuristic adapted based on the algorithm from Ahmadi-Javid and Azad (2010). The stopping criterion and initial solutions for both algorithms are set the same. Table 2 shows the results of this comparison for 15 instances (see the fifth column). It can be seen that the proposed heuristic considerably outperforms the adapted algorithm. The percentage of improvement obtained by the proposed heuristic varies between 5.06% and 6.74%.

4.3. *Effect of initial solution quality*

This section investigates the impact of the initial solution quality on the final solution quality. It compares two different procedures to construct initial solutions: the proposed heuristic given in Section 3.3 and a random procedure explained briefly in the following. First, customers are randomly assigned to a set of distribution centres, and then routes through the customers are built by using the nearest-neighbour algorithm, given in Section 3.1. Finally, the SDA heuristic, proposed in Section 3.2, is used to assign each open distribution centre to a supplier.

Table 2 presents the results of this comparison (see the sixth column). As seen from this comparison the method proposed in Section 3.3 significantly improves the quality of the final solutions.

Table 1. Solutions obtained by the proposed heuristic for Barreto's LRP benchmark.

			Heuristic algorithm		
No.	Instance's name in Barreto (2004)	Lower bound	Total cost	CPU time (sec)	Gap (%)
1	Christofides69-50x5	549.4	565.6	4.8	2.95
2	Christofides69-100x10	788.6	839.8	64.4	6.49
3	Daskin95-88x8	356.4	356.6	61.9	0.06
4	Daskin95-150x10	43406.0	45065.0	108.6	3.82
5	Gaskell67-21x5	424.9	424.9	0.9	0.00
6	Gaskell67-22x5	585.1	585.1	1.6	0.00
7	Gaskell67-29x5	512.1	512.1	2.4	0.00
8	Gaskell67-32x5	556.5	562.2	3.6	1.02
9	Gaskell67-32x5	504.3	504.3	3.4	0.00
10	Gaskell67-36x5	460.4	460.4	4.3	0.00
11	Min92-27x5	3062.0	3062.0	1.4	0.00
12	Perl83-12x2	204.0	204.0	0.5	0.00
13	Perl83-55x15	1074.8	1115.5	7.8	3.79
14	Perl83-85x7	1568.1	1631.1	17.1	4.02
15	Or76-117x14	12048.4	12614.6	135.8	4.70

Table 2. Summary of numerical study.

No.	No. of customers	No. of potential DCs	No. of suppliers	I1	I2	I3	I4
16	20	6	3	5.67	5.30	15.01	6.26
17	40	12	5	5.82	4.94	25.43	6.03
18	50	15	6	7.54	7.99	20.28	8.59
19	80	20	9	6.74	7.48	18.79	7.76
20	100	22	10	5.73	6.83	17.70	6.02
21	120	24	11	5.74	5.72	21.75	5.80
22	150	28	13	5.30	5.10	16.15	5.86
23	180	30	14	5.12	8.87	15.13	5.47
24	200	34	16	5.06	4.84	19.54	5.61
25	220	36	17	6.26	9.04	17.51	6.57
26	250	40	19	5.91	7.71	18.92	6.04
27	280	42	20	6.49	9.24	14.44	6.66
28	300	45	22	5.91	8.82	15.97	6.06
29	320	47	23	6.32	9.63	14.18	6.59
30	350	50	25	6.12	9.52	16.49	6.60

Notes:
I1, improvement (%) obtained by the proposed heuristic over the one adapted based on the heuristic of Ahmadi-Javid and Azad (2010).
I2, improvement (%) obtained by the proposed heuristic given in Section 3.3 over random procedure to construct initial solution.
I3, improvement (%) obtained by the proposed heuristic over initial solution.
I4, improvement (%) obtained by integrating inventory with location and routing decisions.

4.4. *Analysis of the heuristic's components*

This section analyses the impact of each component of the proposed heuristic; *i.e.* Mov1–Mov5, and routing-2 phase, on the overall improvement. The averages of improvement percentages obtained by using each component are 19.92%, 20.62%, 26.26%, 6.33%, 21.77% and 5.10%, respectively. This indicates that each component significantly contributes to the overall improvement.

Table 2 reports the overall improvement by the proposed heuristic over initial solutions (see the seventh column). These improvements are considerably high, ranging from 14.18% to 25.43%.

4.5. *Impact of ternary integration*

This section assesses the effect of integrating inventory with location and routing decisions. First the location-routing problem is solved by adapting the proposed heuristic to solve an LRP which only involves location and routing costs, with the objective function

$$\sum_{j\in J}\sum_{n\in N} f_j^n U_j^n + \sum_{v\in V}\left\{\frac{\sum_{k\in M}\sum_{l\in M} tc_{kl} R_{klv}}{\sum_{k\in M}\sum_{l\in M} tt_{kl} R_{klv}}\right\}.$$

Then, for the resulting solutions, the objective incorporating the inventory cost, *i.e.* the objective function in (16), is computed. Also, for each instance, based on the proposed ternary-integration problem a solution is found. Finally, the two objective values for each instance are compared.

Table 2 shows the results of this comparison (see the eighth column). According to this table, integrating inventory decisions with location and routing decisions during the optimization procedure significantly reduces the total cost, when compared with the traditional approach, in which the location and routing decisions are first determined, and then the best inventory policy for each open distribution centre is set. The improvement percentages are between 5.47% and 8.59%. This analysis proves that the integration of location, routing and inventory decisions considerably reduces the total cost.

5. Conclusions

This article outlines an integration multisource-distribution-network design problem that optimizes location, inventory and routing decisions simultaneously. The objective of the problem is to minimize the yearly total cost of location, routing and inventory. A multisource distribution network and a realistic cyclic-routing pattern are features of the proposed problem which are not considered in Ahmadi-Javid and Azad (2010).

A heuristic method that iteratively improves the current solution in three phases is proposed. After generating an initial solution, a simulated annealing algorithm is used to improve the initial solution in the first two phases, and then a hybrid ant colony system algorithm is applied in the third phase.

The performance of the proposed heuristic is evaluated by applying it to Barreto's benchmark provided for a location-routing problem, and also it is compared to the heuristic derived from the one given by Ahmadi-Javid and Azad (2010). This numerical study shows the effectiveness of the solution method.

Finally it is shown that the new integration approach significantly decreases the total cost needed to design and plan the multisource distribution network, in comparison to the traditional approach, in which, first, only location and routing decisions are optimized simultaneously, and then, for each open distribution centre, an inventory problem is solved.

References

Ahmadi-Javid, A. and Azad, N., 2010. Incorporating location, routing and inventory decisions in supply chain network design. *Transportation Research Part E*, 46, 582–597.

Barreto, S., 2004. *Analysis and modeling of location-routing problems.* Thesis (PhD). University of Aveiro.

Bertazzi, L., Speranza, M.G. and Savelsbergh, M.W.P., 2008. Inventory routing. *In*: B. Golden, R. Raghavan, E. and Wasil, eds. *Vehicle routing: Latest advances and new challenges*. New York: Springer, 49–72.

Daskin, M., Coullard, C. and Shen, Z.J., 2002. An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*, 110, 83–106.

Dorigo, M. and Gambardella, L.M., 1997. Ant Colony System: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1, 53–66.

El Hassani, A.H., Bouhafs, L. and Koukam, A., 2008. A hybrid ant colony system approach for the capacitated vehicle routing problem and the capacitated vehicle routing problem with time windows, *in* T. Caric and H. Gold, eds. *Vehicle routing problem*. Vienna: InTech, 57–70.

Federgruen, A. and Simchi–Levi, D., 1995. Analysis of vehicle routing and inventory-routing problems. *In*: M.O. Ball, T.L. Magnanti, C.L. Monma, G.L. Nemhauser, eds. *Handbooks in operations research and management science*. North-Holland: Elsevier Science, 8, 297–373.

Goldberg, D.E., 1989. *Genetic algorithms in search, optimization and machine learning.*, Reading, MA: Addison-Wesley.

Kirkpatrick, S., Gelatt Jr, C.D. and Vecchi, M.P., 1983. Optimization by simulated annealing. *Science*, 220, 671–680.

Kulkarni, R.V. and Bhave, P.R., 1985. Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20, 58–67.

Miller, C.E., Tucker, A.W. and Zemlin, R.A., 1960. Integer programming formulations and traveling salesman problems. *Journal of the ACM*, 7, 326–329.

Min, H., Jayaraman, V. and Srivastava, R., 1998. Combined location–routing problems: A synthesis and future research directions. *European Journal of Operational Research*, 108, 1–15.

Miranda, P.A. and Garrido, R.A., 2008. Valid inequalities for Lagrangian relaxation in an inventory location problem with stochastic capacity. *Transportation Research Part E*, 44, 47–65.

Nagy, G. and Salhi, S., 2007. Location-routing, issues, models, and methods: A review. *European Journal of Operational Research*, 117, 649–672.

Ozsen, L., Daskin, M.S. and Coullard, C.R., 2006. Capacitated warehouse location model with risk pooling. *Naval Research Logistics*, 55, 295–312.

Perl, J. and Daskin, M.S., 1985. A warehouse location–routing problem. *Transportation Research Part B*, 19, 381–396.

Raa, B. and Aghezzaf, E-H., 2008. Designing distribution patterns for long-term inventory routing with constant demand rates. *International Journal of Production Economics*, 112, 255–263.

Raa, B. and Aghezzaf, E.-H., 2009. A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, 192, 429–441.

Shen, Z.J., 2005. A multi-commodity supply chain design problem. *IIE Transactions*, 37, 753–762.

Shen, Z.J. and Qi, L., 2007. Incorporating inventory and routing cost in strategic location models. *European Journal of Operational Research*, 179, 372–389.
Yu, Y., Chen, H. and Chu, F., 2008. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research*, 189, 1022–1040.
Yu, V.F., Lin, S.W., Lee, W. and Ting, C.J., 2010. A simulated annealing heuristic for the capacitated location routing problem. *Computers and Industrial Engineering*, 58, 288–299.
Zhao, Q., Wang, S.Y., Lai, K.K. and Xia, G.P., 2007. A partition approach to the inventory/routing problem. *European Journal of Operational Research*, 177, 786–802.