Transportation Research Part E 46 (2010) 582–597

Contents lists available at ScienceDirect

Transportation Research Part E

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

Incorporating location, routing and inventory decisions in supply chain network design

Amir Ahmadi Javid *, Nader Azad

Department of Industrial Engineering, Amirkabir University of Technology (Tehran Polytechnic), P.O. Box 15875-4413, Tehran, Iran

ARTICLE INFO

Article history:
Received 14 May 2008
Received in revised form 7 June 2009
Accepted 19 June 2009

Keywords:
Distribution network
Integrated supply chain design
Facility location
Inventory control
Vehicle routing
Capacity level

ABSTRACT

This paper for the first time presents a novel model to simultaneously optimize location, allocation, capacity, inventory, and routing decisions in a stochastic supply chain system. Each customer's demand is uncertain and follows a normal distribution, and each distribution center maintains a certain amount of safety stock. To solve the model, first we present an exact solution method by casting the problem as a mixed integer convex program, and then we establish a heuristic method based on a hybridization of Tabu Search and Simulated Annealing. The results show that the proposed heuristic is considerably efficient and effective for a broad range of problem sizes.

1. Introduction

A key driver of the overall productivity and profitability of a supply chain is its distribution network which can be used to achieve a variety of the supply chain objectives ranging from low cost to high responsiveness. Designing a distribution network consists of three subproblems: location–allocation problem, vehicle routing problem, and inventory control problem. Because of high dependency among these problems, in the literature there are several papers integrating two of the above problems: location–routing problems, inventory–routing problems, and location–inventory problems. Location–routing problems are surveyed and classified by Min et al. (1998) and Nagy and Salhi (2007). Inventory–routing problems are studied in several papers, e.g. Baita et al. (1998), Jaillet et al. (2002), Kleywegt et al. (2002), Adelman (2004), Gaur and Fisher (2004), Zhao et al. (2008), Yu et al. (2008), Oppen and Loketangen (2008) and Day et al. (2009). Also, several papers considered location–inventory problems, e.g. Erlebacher and Meller (2000), Daskin et al. (2002) and Shen (2005).

Recently, Shen and Qi (2007) modified the inventory–location model given in Daskin et al. (2002). The objective function of their model is the sum of the inventory–location cost and an approximate routing cost which depends only on the locations of the opened distribution centers. They showed that significant cost saving can be obtained by their model in comparison with the sequential approach. However, their model optimizes only the inventory and location decisions and does not determine transportation decisions. Furthermore, there is no guarantee that their model can be used for real-world cases since their approximation method is applicable only under some restrictive assumptions.

In this paper, for the first time we present a model which simultaneously optimizes location, allocation, capacity, inventory and routing decisions without any approximation. To solve the problem, first we present an optimal solution method by expressing the problem as a mixed integer convex program. Since location–routing problems have been shown to be

* Corresponding author.
E-mail address: ahmadi_javid@aut.ac.ir (A. Ahmadi Javid).

1366-5545/$ - see front matter © 2009 Published by Elsevier Ltd.
doi:10.1016/j.tre.2009.06.005

NP-hard (Perl and Daskin, 1985), our problem belongs to the class of NP-hard problems too. Hence, in the following to solve the large-sized instances, a heuristic method is developed. The heuristic method is decomposed into two stages: constructive stage and improvement stage. In the constructive stage an initial solution is built at random. In the improvement stage we have two phases: location phase and routing phase, and a hybrid algorithm based on Tabu Search and Simulated Annealing is used to improve the initial solution in each phase.

The remainder of the paper is organized as follows. In Section 2, the mathematical formulation of the problem is given. Section 3 presents the solution methods for solving the problem. Section 4 studies the model under extra constraints. The computational results are presented in Section 5. We conclude the paper in Section 6.

2. Problem description and formulation

The goal of our model is to choose, locate and allocate a set of distribution centers, to determine the inventory policy and to schedule vehicles' routes to meet customers' demands such that the total cost is minimized. We assume that each customer has an uncertain demand that follows a normal distribution. In the model, we use different capacity levels for each distribution center, which makes the problem more realistic and increases the capacity utilization of distribution centers to a high level. Our assumptions and decisions determined by the model are explained as follows.

2.1. Assumptions

- Each customer has an uncertain demand that follows a normal distribution, and the customers' demands are independent.
- All the possible capacity levels for the set of distribution centers are known, and the company pays a fixed location cost for opening a distribution center with a capacity level.
- The company pays a fixed cost for placing each order and a cost for holding inventory at each distribution center.
- Each distribution center j is assumed to follow a (Q_j, R_j) inventory policy, i.e., when the inventory level at distribution center j falls to or below a reorder point R_j, a fixed quantity Q_j is ordered to the supplier. Also, each distribution center holds a safety stock to buffer the system against stock out during lead times.
- Vehicles' capacities are the same, and fleet type is homogeneous.

2.2. Decisions

- *Location, capacity level and allocation decisions*: how many distribution centers to locate, where to locate the opened distribution centers, what capacity level to consider for each of them, and how to allocate the customers to them.
- *Routing decisions*: how to build the vehicles' routes starting from an opened distribution center to serve its allocated customers.
- *Inventory decisions*: how often to reorder at a distribution center and what level of safety stock to maintain.

Now we integrate these three decisions in a mathematical programming model under the aforementioned assumptions. Before presenting the model, let us introduce the notation used throughout the paper.

2.3. Index sets

K	set of customers
J	set of potential distribution centers
N_j	set of capacity levels available to distribution center $(j \in J)$
V	set of vehicles
M	merged set of customers and potential distribution centers, i.e. $(K \cup J)$

2.4. Parameters and notations

B	number of customers contained in set K, i.e. $B = \lvert K \rvert$
μ_k	mean of yearly demand at customer k $(\forall k \in K)$
σ_k^2	variance of yearly demand at customer k $(\forall k \in K)$
f_j^n	yearly fixed cost for opening and operating distribution center j with capacity level n $(\forall j \in J,\ \forall n \in N_j)$
b_j^n	capacity with level n for distribution center j $(\forall j \in J,\ \forall n \in N_j)$.
d_{kl}	transportation cost between node k and node l $(\forall k, l \in M)$
vc	annual delivery capacity of a vehicle
q	number of visits of each vehicle in a year
h_j	inventory holding cost per unit of product per year at distribution center j $(\forall j \in J)$
p_j	fixed cost per order placed to the supplier by distribution center j $(\forall j \in J)$
lt_j	lead time of distribution center j in years $(\forall j \in J)$

g_j fixed cost per shipment from supplier to distribution center j ($\forall j \in J$)
a_j cost per unit of shipment from the supplier to distribution center j ($\forall j \in J$)
α desired percentage of customer orders that should be satisfied (fill rate), $\alpha > 0.5$
z_α left α-percentile of standard normal random variable Z, i.e. $p(Z \leqslant z_\alpha) = \alpha$
β weight factor associated with transportation cost
θ weight factor associated with inventory cost.

The weights β and θ are used to increase or decrease the relative importance of the location, transportation and inventory costs in the objective function.

2.5. Decision variables

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

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

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

Q_j: Order size at distribution center j ($\forall j \in J$).
M_{kv}: Auxiliary variable defined for customer k for subtour elimination in route of vehicle v ($\forall k \in K$, $\forall v \in V$).

2.6. Objective function

The objective function includes the following costs:

1. The fixed cost of locating the opened distribution centers, given as $\sum_{j \in J} \sum_{n \in N_j} f_j^n U_j^n$.
2. The expected working inventory and safety stock costs. The expected working inventory cost includes the expected costs of placing orders and holding working inventory, given as $\theta p_j \frac{D_j}{Q_j} + \theta h_j \frac{Q_j}{2}$ where D_j denotes the total annual expected demand going through the distribution center j, i.e. $D_j = \sum_{k \in k} \mu_k Y_{jk}$. The yearly safety stock cost at the distribution center j is $\theta\, h_j z_\alpha \sigma'_j$ with $\sigma'_j = \sqrt{lt_j \sum_{k \in K} \sigma_k^2 Y_{jk}}$.
3. The annual routing cost from the opened distribution centers to the customers, given by $\beta q \sum_{v \in V} \sum_{k \in M} \sum_{l \in M} d_{kl} R_{klv}$, and shipment cost, given by $\beta(g_j + a_j Q_j) \frac{D_j}{Q_j}$. We assume that cost of shipping an order of size R from the supplier to the distribution center j is $g_j + a_j \times R$.

The problem formulation is as follows.

$$\begin{aligned} \min: \quad & \sum_{j \in J} \sum_{n \in N_j} f_j^n U_j^n + \beta q \sum_{v \in V} \sum_{k \in M} \sum_{l \in M} d_{kl} R_{klv} \\ & + \sum_{j \in J} \left[(\theta p_j + \beta g_j) \frac{\sum_{k \in K} \mu_k Y_{jk}}{Q_j} + \beta a_j \sum_{k \in K} \mu_k Y_{jk} + \frac{\theta h_j Q_j}{2} + \theta h_j z_\alpha \sqrt{lt_j \sum_{k \in K} \sigma_k^2 Y_{jk}} \right] \end{aligned} \tag{1}$$

Subject to:

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

$$\sum_{l \in K} \mu_l \sum_{k \in M} R_{klv} \leqslant vc \quad \forall v \in V \tag{3}$$

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

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

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

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

$$\sum_{n\in N_j} U_j^n \leqslant 1 \quad \forall j \in J \tag{8}$$

$$\sum_{k\in K} \mu_k Y_{jk} \leqslant \sum_{n\in N_j} b_j^n U_j^n \quad \forall j \in J \tag{9}$$

$$\begin{aligned} &Y_{jk} \in \{0,1\} \quad \forall j \in J, \quad \forall k \in K \\ &U_j^n \in \{0,1\} \quad \forall j \in J, \quad \forall n \in N_j \\ &R_{klv} \in \{0,1\} \quad \forall k,l \in M, \quad \forall v \in V \end{aligned} \tag{10}$$

$$\begin{aligned} &M_{kv} \geqslant 0 \quad \forall k \in K, \quad \forall v \in V \\ &Q_j > 0 \quad \forall j \in J \end{aligned} \tag{11}$$

The model minimizes the total expected cost consisting of the fixed cost for opening distribution centers, the annual routing and shipment cost, and the expected annual inventory cost. Constraints (2) make sure that each customer is placed on exactly one vehicle route. Constraints (3) are the vehicle capacity constraints. Constraints (4) are the subtour elimination constraints which guarantee each tour must contain a distribution center from which it originates, i.e. each tour must consist of a distribution center and some customers (Desrochers and Laporte, 1991). Constraints (5) are flow conservation constraints saying that whenever a vehicle enters a customer or distribution center node, it must leave again and ensuring that the routes remain circular. Constraints (6) imply that only one distribution center is included in each route. Constraints (7) link the allocation and the routing components of the model: the customer k is assigned to the distribution center j if the vehicle v, which visits the customer k, starts its trip from the distribution center j. Constraints (8) ensure that each distribution center can be assigned to only one capacity level. Constraints (9) are the capacity constraints associated with the distribution centers. Constraints (10) enforce the integrality restrictions on the binary variables, and finally Constraints (11) enforce the non-negativity restrictions on the other decision variables.

3. Solution method

In this section, we propose two methods for solving the problem: an optimal solution method and a heuristic method.

3.1. Finding optimal solution

In the model (1)–(10), the decision variable Q_j, only has appeared in the objective function. Also, the objective function is convex in $Q_j > 0$; consequently, we can obtain the optimal value of Q_j by taking the derivative of the objective function with respect to Q_j as:

$$Q_j^* = \sqrt{\frac{2(\theta p_j + \beta g_j)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}} \tag{12}$$

By substituting (12) in the objective function (1), the problem can be formulated as:

$$\begin{aligned} \min: \quad & \sum_{j\in J}\sum_{n\in N_j} f_j^n U_j^n + \beta q \sum_{v\in V}\sum_{k\in M}\sum_{l\in M} d_{kl} R_{klv} \\ & + \sum_{j\in J}\left[\sqrt{2\theta h_j(\theta p_j + \beta g_j)\sum_{k\in K}\mu_k Y_{jk}} + \beta a_j \sum_{k\in K}\mu_k Y_{jk} + \theta h_j z_\alpha \sqrt{lt_j \sum_{k\in K}\sigma_k^2 Y_{jk}}\right] \end{aligned} \tag{13}$$

Subject to: (2)–(11).

The continuous relaxation of the objective function (13) is concave and the problem cannot be optimized globally by the existing optimization methods. Since the variable Y_{jk} is a binary variable, we can substitute Y_{jk} with Y_{jk}^2 in the objective function (13) to convexify it. In this case, the model can be written as follows.

$$\begin{aligned} \min: \quad & \sum_{j\in J}\sum_{n\in N_j} f_j^n U_j^n + \beta q \sum_{v\in V}\sum_{k\in M}\sum_{l\in M} d_{kl} R_{klv} \\ & + \sum_{j\in J}\left[\sqrt{2\theta h_j(\theta p_j + \beta g_j)\sum_{k\in K}\mu_k Y_{jk}^2} + \beta a_j \sum_{k\in K}\mu_k Y_{jk}^2 + \theta h_j z_\alpha \sqrt{lt_j \sum_{k\in K}\sigma_k^2 Y_{jk}^2}\right] \end{aligned} \tag{14}$$

Subject to: (2)–(11).

The nonlinear terms in the objective function (14) are the functions:

$$\sqrt{2\theta h_j(\theta p_j + \beta g_j)\sum_{k\in K}\mu_k Y_{jk}^2},\quad \sum_{k\in K}\mu_k Y_{jk}^2,\quad \sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}^2}$$

The continuous relaxation of the second function is a convex quadratic function. Moreover, the relaxation of the first and third terms can be written as linear transformations of Euclidean norms, so they are convex too. Therefore, the continuous relaxation of the above model is a convex program. As a result, the model is a mixed integer convex program, and we can solve the model with branch and bound methods for small sized instances optimally. However, for medium-sized or large-sized problems, optimally solving the model is impractical since the model has a nonlinear objective and consists of numerous binary variables and constraints. This motivates us to develop an appropriate heuristic in the next section.

3.2. Heuristic method

As we mentioned in the introduction, the proposed problem belongs to the class of NP-hard problems, so in this section we develop a heuristic method to solve the problem in large scales. The heuristic method is decomposed into two stages: constructive stage, where an initial solution is built at random, and improvement stage, where the solution is iteratively improved by modifying the location and routing decisions in two phases: location phase and routing phase. The stopping criterion of the algorithm is as follows

Is $count = max\text{-}count$?

where we initially set $count = 1$ and $max\text{-}count = 4$. If the above criterion is met, we stop; otherwise, we set $count = count + 1$ and continue to improve the current solution in the location phase and then in the routing phase. In Fig. 1 we have demonstrated the flowchart of the proposed heuristic.

In both phases, we use a hybrid heuristic based on Tabu Search (TS) and Simulated Annealing (SA) to improve the current solution. Tabu Search and Simulated Annealing are two important heuristic solution approaches which are extremely successful to solve hard combinatorial problems. Simulated Annealing uses a stochastic approach to continue the search. It allows the search to proceed to neighboring state even if the move causes the value of the objective function become worse. This important aspect can allow it to prevent falling in local optimum traps. The most important feature of Tabu Search is to avoid search cycling by systematically preventing moves that generate the solutions previously visited in the solution space. Therefore, our reason to choose a hybridization of SA and TS is to simultaneously have the two powerful features of Tabu Search and Simulated Annealing, i.e. to avoid search cycling using Tabu Search and to prevent falling in local optimum traps

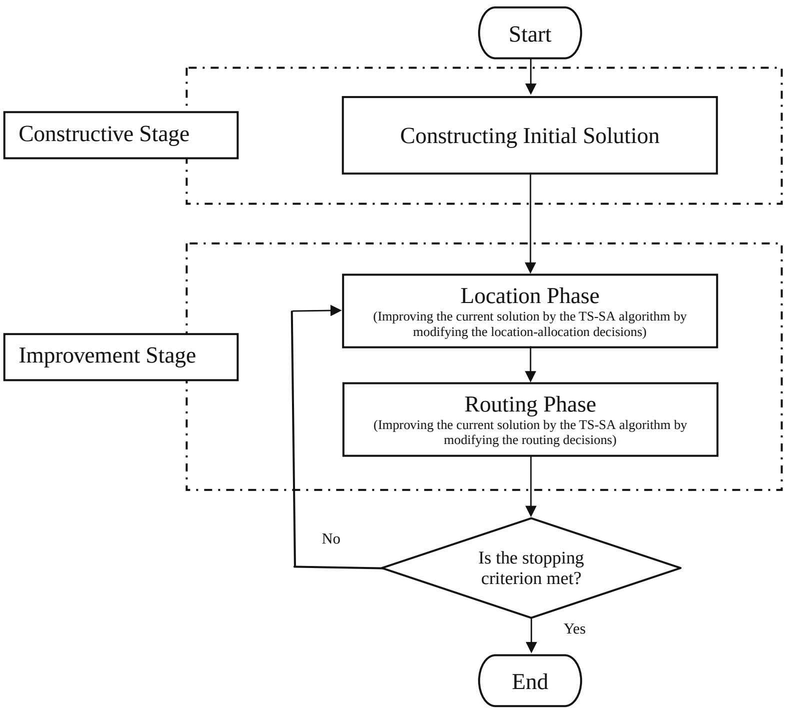

Fig. 1. Flowchart of proposed heuristic method.

by Simulated Annealing. Now, we describe the hybrid TS–SA algorithm which we use in each phase of the improvement stage. The parameters of the algorithm are as follows.

IT_0: Initial temperature.
CS: Decreasing rate of current temperature (cooling schedule).
FT: Freezing temperature (the temperature at which the desired energy level is reached).
MNT: Maximum number of accepted solutions at each temperature.
nt: Counter for number of accepted solutions at each temperature.
X_0: Initial solution.
X: Current solution in algorithm.
X_{nh}: Solution which is selected in neighborhood of X in each iteration.
X_{best}: Best solution obtained in algorithm.
$C(X)$: Objective function value for solution X.

The steps of the hybrid TS–SA algorithm which we use in the improvement phase are as follows.

Step 1: Take the initial solution X_0, and set $X_{best} = X_0$, $X = X_0$.
Step 2: Generate the solution X_{nh} in the neighborhood of X .
Step 3: Is the move generated X_{nh} in the tabu list? If yes, go to Step 4; otherwise, go to Step 5.
Step 4: If $C(X_{nh}) \leqslant C(X_{best})$, then $X = X_{nh}, X_{best} = X_{nh}$ and go to Step 6; otherwise, go to Step 2 for choosing another candidate move, and update the tabu list.
Step 5: Update the tabu list and let $\Delta C = C(X_{nh}) - C(X)$.
5.1. If $\Delta C \leqslant 0$, then $X = X_{nh}$. If $C(X_{nh}) < C(X_{best}), X_{best} = X_{nh}$.
5.2. If $\Delta C > 0$, $y \leftarrow U(0,1)$, $z = e^{-\frac{\Delta C}{T}}$. If $y < z$, then $X = X_{nh}$.
Step 6: Is the number of iterations in temperature T not greater than MNT ? If yes, go to Step 2; if not go to Step 7.
Step 7: $T = CS \times T$.
Step 8: Is the stopping criterion ($T < FT$) matched? If yes, stop; or else go to Step 2.

In Fig. 2 the pseudo code of the hybrid TS–SA algorithm is given. Note that this algorithm will be used in each phase. When the algorithm is applied for the two phases all the things are the same with the exception of how the algorithm generates a neighboring solution X_{nh} in the neighborhood of the current solution X. This means that in the location and routing phases we use different moves to generate neighboring solutions by changing the location and routing decisions, respectively.

In Section 3.2.1 we describe the constructive stage where we show how we generate an initial solution. In Section 3.2.2 we study the moves used in the two phases of the improvement stage. In Section 3.2.3 we discuss about the convergence of the heuristic, and in Section 3.2.4 we present a method for providing an analytical error bound to evaluate the solutions obtained by the heuristic.

3.2.1. Constructive stage

In the constructive stage, we construct an initial solution. To this end, first we randomly assign customers to the distribution centers. For each of the opened distribution centers, the capacity level is selected randomly, and we build routes through the customers by using the nearest neighbor algorithm. The procedure of determining the initial solution is as follows.

Step 1: Put all the customers into set K' .
Step 2:
2-1. Select a customer from K' randomly.
2-2. Delete the customer from K'.

Step 3: Select a distribution center randomly.
Step 4: If the distribution center is selected for the first time, then select a capacity level for this distribution center randomly.
Step 5: If the remaining capacity of the distribution center is greater than the demand of the customer selected in Step 2, then assign the customer to the distribution center and go to Step 6; otherwise, go to Step 3 for selecting another distribution center.
Step 6: Is K' empty? If yes, go to step 7, if not go to step 2.
Step 7: Build routing of the customers, by using the nearest neighbor algorithm for each opened distribution center.

3.2.2. Improvement stage

In the improvement stage, we have two phases: the location phase and routing phase. In this stage, the current solution improved by alternatively modifying the location and routing decisions. First the initial solution is improved in the location

```
Take X_0
X_best = X_0, X = X_0, T = IT_0
While T ≥ FT do
    nt = 0
    While nt ≤ MNT do
        Generate X_nh in the neighborhood of X (the generation differs from a phase to another)
        ΔC = C(X_nh) − C(X)
        If the move generated X_nh is in the tabu list then
            If C(X_nh) < C(X_best) then
                X = X_nh, X_best = X_nh, Update the tabu list
            End If
        Else
            Update the tabu list
            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
                y ← U(0,1)
                z = e^(−ΔC/T)
                If y < z then
                    X = X_nh, nt = nt + 1
                End If
            End If
        End If
    End While
    T = CS × T
End While
```

Fig. 2. Pseudo code of hybrid TS–SA algorithm which is used in each phase of improvement stage of heuristic method.

phase, and then the resulting solution is improved in the routing phase. Subsequently the obtained solution from the routing phase is improved in the location phase, and the procedure is continued in the same manner until the stopping criterion is matched. In Sections 3.2.2.1 and 3.2.2.2 we respectively present the moves employed by the hybrid TS–SA algorithm in the location and routing phases to generate a neighboring solution in the neighborhood of the current solution.

3.2.2.1. Location phase. In this phase, the main objective is to improve the current solution by modifying the number and locations of distribution centers, the capacity level of each opened distribution center and allocation of the customers. The solution obtained in this phase is used as an input to the routing phase. In this phase we apply three different moves Mov1, Mov2, and Mov3. We randomly select a candidate move from Mov1, Mov2, and Mov3 to obtain a new solution X_{nh} in the neighborhood of the current solution X.

Mov1: One of the opened distribution centers (j) is randomly closed and all of its customers are reallocated among the remaining opened distribution centers. If the remaining capacities of the opened distribution centers are not enough for serving the customers of j, then we randomly select an opened distribution center and increase its capacity level to a higher level, and finally we rebuild the routes using the nearest neighbor algorithm.

In the case that Mov1 is frequently unsuccessful to generate a neighborhood solution; we give up this move during the location stage. To this end, we consider the input parameter *max-Mov1* as the maximum number we allow this move can be failed.

Mov2: In this move we select two opened distribution centers i and j randomly, and exchange the customers assigned to i with the customers assigned to j. Then the nearest neighbor algorithm is implemented for the distribution centers i and j to build the routes. In this move the capacities of i and j are adjusted for serving the customers that are newly assigned to them.

Mov3: One of the opened distribution centers (i) is closed randomly, and a closed distribution center (j) is opened randomly, and then we assign all of the routes corresponding to the eliminated distribution center i to the new opened distribution center j. In this move the capacity of j is updated for serving its customers.

By the above three moves the hybrid TS–SA algorithm proceeds until $T < FT$ at this phase**.** Then, the obtained solution will be an input to the routing phase described in the next subsection.

3.2.2.2. Routing phase. In this phase, we try to improve the current solution by making modifications on the routes. In this phase, we apply two moves Mov4 and Mov5. We randomly select a candidate move from Mov4 and Mov5 to obtain a new solution X_{nh} in the neighborhood of the current solution X .

Mov4: One of the opened distribution centers (j) is selected randomly and then one of the routes (v_j) of j is chosen at random. Then route v_j is eliminated and all of its customers are randomly reallocated among the remaining opened distribution centers. If the remaining capacities of the opened distribution centers are not enough to serve the customers of the route v_j, we randomly select an opened distribution center and increase its capacity level to a higher level to satisfy the demands of its customers. Finally we rebuild the routes of all the opened distribution centers to which the customers of v_j assigned by using the nearest neighbor algorithm.

For this move, we define the input parameter *max-Mov4* as the maximum number we allow this move can be failed. In fact, if Mov4 is unsuccessful as many times as *max-Mov4* to generate a neighborhood solution, we give up this move during the routing phase.

Mov5: Select two routes (v_i, v_j) randomly. Then randomly select a customer (k_i) in v_i and a customer (k_j) in v_j, and then exchange k_i with k_j. In this move we must check the capacities of the opened distribution centers and the routes.

The heuristic is applied to the solution found in the location phase, and used Mov4 and Mov5 to generate neighboring solutions. This process is repeated until $T > FT$ at this phase**.** When we exit from this phase, the stopping criterion of the algorithm is checked. If the stopping criterion (i.e. $count = 1$ *max-count*) is met we stop, else we set $count = count + 1$ and go to the location phase.

3.2.3. Convergence of heuristic

The convergence of the proposed heuristic method follows from the wealthy theoretical convergence analysis developed for Simulated Annealing algorithms. In the literature several authors (e.g. Geman and Geman, 1984; Mitra et al., 1986; Bertsimas and Tsitsiklis, 1993) studied the convergence of Simulated Annealing algorithms. It can be proved that a Simulated Annealing algorithm converges in the limit to a globally optimal solution with probability 1. This means that an exponentially long annealing schedule can guarantee convergence to the global optimum. We refer to Spall (2003) for an extra introduction to both the theoretical and practical aspects of Simulated Annealing.

3.2.4. Error bound

For a heuristic solution method presenting an error bound helps us to understand the quality of the solutions obtained by the heuristic. Here to develop an error bound we find a lower bound for the optimal objective value by solving the model (14), (2)–(10). In fact, this model is obtained from the model presented in Section 3.1 by relaxing the subtour elimination constraints, i.e. Constraints (4). Our computational results in Section 5.3 show that by the proposed lower bound we can give acceptable error bounds for relatively large instances with uncapacitated distribution centers.

4. Studying model under extra constraints

This section studies more realistic constraints that can be added to the proposed model, and also shows how the exact and heuristic solution methods presented in the two previous sections can be updated in the presence of these additional constraints.

4.1. Modification of Constraints (9)

Constraints (9) are considered in Miranda and Garrido (2004), and then more realist extensions of them introduced in Miranda and Garrido (2008) as:

$$Q_j + z_\alpha\sqrt{lt_j\sum\nolimits_{k\in K}\sigma_k^2 Y_{jk}} \leqslant \sum_{n\in N_j} b_j^n U_j^n \quad \forall j \in J \tag{15}$$

By replacing Constraints (9) by these constraints in the model, the decision variable Q_j appears in the objective function (1) and Constraints (15). Thus, the optimal value of Q_j could be obtained by solving the following model.

$$\begin{aligned}
\min \quad & f(Q_j) = (\theta p_j + \beta g_j)\frac{\sum_{k\in K}\mu_k Y_{jk}}{Q_j} + \frac{\theta h_j Q_j}{2} + \text{terms not including } Q_j \\
\text{subject to :} \quad & Q_j + z_\alpha\sqrt{lt_j\sum\nolimits_{k\in K}\sigma_k^2 Y_{jk}} \leqslant \sum_{n\in N} b_j^n U_j^n \\
& Q_j > 0
\end{aligned}$$

The optimizer of the objective function without considering the first constraint is:

$$\sqrt{\frac{2(\theta p_j + \beta g_j)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}}.$$

Moreover, the objective function tends to infinity as Q_j tends to zero or infinity. Hence, from the convexity of the objective function we have:

$$Q_j^* = \min\left\{\sqrt{\frac{2(\theta p_j + \beta g_j)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}},\quad \sum_{n\in N_j} b_j^n U_j^n - z_\alpha\sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}}\right\} \tag{16}$$

By substituting (16) in the objective function (1) and eliminating the Constraints (15), Q_j will be eliminated from the model, and so the problem can be formulated as:

$$\begin{aligned}\min:\quad & \sum_{j\in J}\sum_{n\in N_j} f_j^n U_j^n + \beta q\sum_{v\in V}\sum_{k\in M}\sum_{l\in M} d_{kl}R_{klv}\\ & +\sum_{j\in J}\left[(\theta p_j + \beta g_j)\max\left\{\sqrt{\frac{\theta h_j\sum_{k\in K}\mu_k Y_{jk}}{2(\theta p_j+\beta g_j)}},\quad \frac{\sum_{k\in K}\mu_k Y_{jk}}{\sum_{n\in N_j} b_j^n U_j^n - z_\alpha\sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}}}\right\}\right.\\ & \left. +\beta a_j\sum_{k\in K}\mu_k Y_{jk} + \frac{\theta h_j}{2}\min\left\{\sqrt{\frac{2(\theta p_j+\beta g_j)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}},\quad \sum_{n\in N_j} b_j^n U_j^n - z_\alpha\sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}}\right\} + \theta h_j z_\alpha\sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}}\right]\end{aligned} \tag{17}$$

Subject to: (2), (3), (4), (5), (6), (7), (8), (10), (11).

- We can see that our heuristic method can be used to solve the problem under Constrains (15). Only the objective function is different from the objective function considered in Section 3.2. The new objective function is given in (17).
- Now we show that the above model can be reformulated as a mixed integer convex program. To do this, first we add some auxiliary variables and constraints to the program as follows.

$$\begin{aligned}\min:\quad & \sum_{j\in J}\sum_{n\in N_j} f_j^n U_j^n + \beta q\sum_{v\in V}\sum_{k\in M}\sum_{l\in M} d_{kl}R_{klv}\\ & +\sum_{j\in J}\left[(\theta p_j+\beta g_j)W_j + \beta a_j\sum_{k\in K}\mu_k Y_{jk} + \frac{\theta h_j}{2}\left(\sqrt{\frac{2(\theta p_j+\beta g_j)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}} - Z_j^-\right) + \theta h_j z_\alpha\sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}}\right]\end{aligned} \tag{18}$$

Subject to: (2), (3), (4), (5), (6), (7), (8), (10), (11) and

$$\sqrt{\frac{\theta h_j\sum_{k\in K}\mu_k Y_{jk}}{2(\theta p_j+\beta g_j)}} \leqslant W_j \quad \forall j\in J \tag{19}$$

$$\frac{\sum_{k\in K}\mu_k Y_{jk}}{\sum_{n\in N_j} b_j^n U_j^n - z_\alpha\sqrt{L_j\sum_{k\in K}\sigma_k^2 Y_{jk}}} \leqslant W_j \quad \forall j\in J \tag{20}$$

$$\sum_{n\in N_j} b_j^n U_j^n - z_\alpha\sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}} - \sqrt{\frac{2(\theta p_j+\beta g_j)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}} \leqslant Z_j^+ - Z_j^- \quad \forall j\in J \tag{21}$$

$$Z_j^+ Z_j^- = 0 \quad \forall j\in J \tag{22}$$

$$Z_j^+, Z_j^- \geqslant 0, W_j > 0 \quad \forall j\in J \tag{23}$$

By noting the fact that we can substitute Y_{jk} with Y_{jk}^2 (since variable Y_{jk} is a binary variable), and observing that all of the following terms:

$$\sqrt{\frac{2(\theta p_j+\beta g_j)\sum_{k\in K}\mu_k Y_{jk}^2}{\theta h_j}},\quad \sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}^2},\quad \frac{Y_{jk}^2}{W_j},\ \sqrt{\frac{\theta h_j\sum_{k\in K}\mu_k Y_{jk}^2}{2(\theta p_j+\beta g_j)}},\quad -z_\alpha\sqrt{lt_j\sum_{k\in K}\sigma_k^2 Y_{jk}},-\sqrt{\frac{2(\theta p_j+\beta g_j)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}}$$

are convex functions, the above program can be reformulated as the following mixed integer convex program.

$$\min: \quad \sum_{j\in J}\sum_{n\in N_j} f_j^n U_j^n + \beta q \sum_{v\in V}\sum_{k\in M}\sum_{l\in M} d_{kl} R_{klv} + \sum_{j\in J}\left[(\theta p_j+\beta g_j)W_j + \beta a_j \sum_{k\in K}\mu_k Y_{jk} + \frac{\theta h_j}{2}\left(\sqrt{\frac{2(\theta p_j+\beta g_j)\sum_{k\in K}\mu_k Y_{jk}^2}{\theta h_j}} - Z_j^-\right) + \theta h_j z_\alpha \sqrt{lt_j \sum_{k\in K}\sigma_k^2 Y_{jk}^2}\right] \tag{24}$$

Subject to: (2), (3), (4), (5), (6), (7), (8), (10), (11) and

$$\sqrt{\frac{\theta h_j \sum_{k\in K}\mu_k Y_{jk}^2}{2(\theta p_j+\beta g_j)}} \leqslant W_j \quad \forall j\in J \tag{25}$$

$$\sum_{k\in K}\mu_k \frac{Y_{jk}^2}{W_j} + z_\alpha \sqrt{lt_j \sum_{k\in K}\sigma_k^2 Y_{jk}^2} \leqslant \sum_{n\in N_j} b_j^n U_j^n \quad \forall j\in J \tag{26}$$

$$\sum_{n\in N_j} b_j^n U_j^n - z_\alpha \sqrt{lt_j \sum_{k\in K}\sigma_k^2 Y_{jk}} - \sqrt{\frac{2(\theta p_j+\beta g_j)\sum_{k\in K}\mu_k Y_{jk}}{\theta h_j}} \leqslant Z_j^+ - Z_j^- \quad \forall j\in J \tag{27}$$

$$Z_j^- \leqslant MV_j \quad \forall j\in J \tag{28}$$

$$Z_j^+ \leqslant M(1-V_j) \quad \forall j\in J \tag{29}$$

$$Z_j^+, Z_j^- \geqslant 0, \quad W_j > 0 \quad \forall j\in J \tag{30}$$

$$V_j \in \{0,1\} \quad \forall j\in J \tag{31}$$

The constant M appeared in Constraints (28) and (29) is a large number.

4.2. Modification of Constraints (3) and (9)

We may be interested in adding other realistic constraints. For example, we can add some dispersion measures of the demands in the left hand-side of Constraints (3) and (9) as:

$$\sum_{l\in K}\mu_l \sum_{k\in M} R_{klv} + z_\rho \sqrt{\sum_{l\in K}\sigma_l^2 \sum_{k\in M} R_{klv}} \leqslant vc \quad \forall v\in V \tag{32}$$

$$\sum_{k\in K}\mu_k Y_{jk} + z_\rho \sqrt{\sum_{k\in K}\sigma_k^2 Y_{jk}} \leqslant \sum_{n\in N_j} b_j^n U_j^n \quad \forall j\in J \tag{33}$$

for some confidence level $\rho > 0.5$. Clearly the heuristic method can be modified to consider such constraints. Furthermore, to find the optimal solution under these constraints, using the technique exploited in Section 3.1, Constraints (32) and (33) can be rewritten as the following convex constraints.

$$\sum_{l\in K}\mu_l \sum_{k\in M} R_{klv} + z_\rho \sqrt{\sum_{l\in K}\sigma_l^2 \sum_{k\in M} R_{klv}^2} \leqslant vc \quad \forall v\in V \tag{34}$$

$$\sum_{k\in K}\mu_k Y_{jk} + z_\rho \sqrt{\sum_{k\in K}\sigma_k^2 Y_{jk}^2} \leqslant \sum_{n\in N_j} b_j^n U_j^n \quad \forall j\in J \tag{35}$$

5. Computational results

To evaluate the performance of our overall solution procedure, in this section we provide extensive computational experiments. The test instances are constructed as follows. The averages of customers' yearly demands are drawn from a uniform distribution between 400 and 1500, and the variances are drawn from a uniform distribution between 10 and 30. In addition, the values of the problem parameters are determined as the following.

d_{kl} is uniformly drawn from [0, 300].
h_j is uniformly drawn from [5, 10].
p_j is uniformly drawn from [10, 15].
lt_j is uniformly drawn from [6/365, 10/365].

g_j is uniformly drawn from [10, 15].
a_j is uniformly drawn from [5, 10].

We assume that the number of working days per year is 300 and a vehicle visits its customers every three days, i.e. q = 100. Also, the fill rate is 97.5%, and so z_α = 1.96.

The vehicle capacity is computed as $vc = 2 \times \left[\frac{D}{|V|}\right]$ where $|V|$ is the number of vehicles. For each distribution center four capacity levels are considered. Let D represent the total average of the customers' demands and $|J|$ be the number of the potential distribution centers, then the four capacities for the distribution center j are defined as:

$$b_j^1 = [cap(j)], \quad b_j^2 = [1.5 \times cap(j)], \quad b_j^3 = [2 \times cap(j)], \quad b_j^4 = [2.5 \times cap(j)]$$

where $cap(j) = c_j \times \frac{D}{|J|}$ and c_j is a random number between 0.8 and 1.2. Also, the corresponding four fixed set up costs of locating and operating are as follows:

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

where k_j' is drawn from a uniform distribution between 300 and 450.

The program of the heuristic method is coded in Visual Basic 6 and run on a Pentium 4 with 2.8 GB processor. For each instance, we run the heuristic method 20 times, and the average objective value is reported in Tables 1–11. In Tables 1 and 2 for each instance the coefficient of variation is reported (Coefficient of variation for the random variable Y is defined as $\mathrm{SD}(Y)/\mathrm{E}(Y)$, where $\mathrm{SD}(Y)$ and $\mathrm{E}(Y)$ are the standard deviation and expected value of Y, respectively). In the following tables DC and CV are the abbreviations of *Distribution Center* and *Coefficient of Variation*, respectively, and the CPU times are in seconds.

5.1. Comparison of optimal and heuristic solution methods

To evaluate the proposed heuristic, several instances have been solved optimally by the method described in Section 3.1 and the heuristic method (see Table 1). To obtain optimal solutions, we solve the objective function (14) with Constraints (2)–(11) by LINGO 8. From Table 1, it can be seen that in instances 1 to 7 the heuristic solutions are optimal or near optimal, and the average CPU times of the heuristic method are less than or equal to 11 s, while as for the exact method the maximum average CPU time is 31,458 s. For instances 8–18, LINGO cannot find the optimal solutions within a reasonable amount of time, and we can see the heuristic solutions are significantly better than the best solutions obtained by LINGO.

5.2. Comparison of heuristic method based on hybrid algorithm with heuristic method based on SA algorithm

In this section, for evaluating the proposed hybrid TS–SA algorithm which we use in each phase of the heuristic, we compare the TS–SA-based heuristic with the heuristic method that uses an SA algorithm in each phase (see Table 2). In the SA algorithm the procedure of obtaining a candidate move is similar to the TS–SA algorithm.

Table 1
Comparison of optimal and heuristic solution methods.

No.	β	θ	# Customers	# Potential DCs	# Vehicles	Optimal method		Heuristic method			
						Average cost	CPU time	Average cost	CV	CPU time	Gap (%)*
1	0.003	0.7	4	2	2	1812.2	9	1812.2	0	1	0.00
2	0.003	0.7	4	3	2	2023.4	29	2023.4	0	1	0.00
3	0.003	0.7	6	2	2	2412.3	150	2412.3	0	2	0.00
4	0.003	0.7	6	3	2	2598.1	379	2598.1	0	3	0.00
5	0.003	0.7	6	4	3	2741.8	587	2741.8	0	4	0.00
6	0.003	0.7	8	3	3	3011.6	19,874	3038.7	0	9	0.90
7	0.003	0.7	9	3	2	3215.1	31,458	3250.3	0	11	1.09
8	0.003	0.7	11	4	3	3684.7	12 h limit	3414.3	0.0001	16	−7.34
9	0.003	0.7	13	4	3	4176.5	12 h limit	3750.6	0.0001	20	−10.20
10	0.003	0.4	20	6	5	4462.8	24 h limit	3804.3	0.0001	34	−14.76
11	0.003	0.7	20	6	5	6800.6	24 h limit	5909.4	0.0001	35	−13.10
12	0.02	0.7	20	6	5	8039.2	24 h limit	7279.1	0.0001	34	−9.45
13	0.003	0.4	40	12	12	7011.2	60 h limit	6093.7	0.0001	55	−13.09
14	0.003	0.7	40	12	12	10393.7	60 h limit	9056.1	0.0001	55	−12.87
15	0.02	0.7	40	12	12	15078.3	60 h limit	13252.9	0.0001	56	−12.11
16	0.002	0.3	50	15	15	9023.4	72 h limit	7743.8	0.0001	66	−14.18
17	0.002	0.7	50	15	15	15006.2	72 h limit	12262.2	0.0001	65	−18.29
18	0.02	0.7	50	15	15	22843.1	72 h limit	18955.1	0.0001	66	−17.02

* Gap (%) = 100 × (Heuristic solution value − LINGO best solution value)/LINGO best solution value.

Table 2
Comparison of heuristic method based on hybrid TS–SA algorithm with heuristic method based on SA algorithm.

No.	β	θ	# Customers	# Potential DCs	# Vehicles	Based on hybrid algorithm			Based on SA algorithm		
						Average cost	CV	CPU time	Average cost	CV	CPU time
1	0.003	0.4	40	12	12	6093.7	0.0001	55	6171.2	0.0002	54
2	0.003	0.7	40	12	12	9056.1	0.0001	55	9154.1	0.0002	54
3	0.02	0.7	40	12	12	13252.9	0.0001	56	13427.5	0.0002	55
4	0.002	0.3	50	15	15	7743.8	0.0001	66	7864.2	0.0002	65
5	0.002	0.7	50	15	15	12262.2	0.0001	65	12469.1	0.0002	63
6	0.02	0.7	50	15	15	18955.1	0.0001	66	19226.8	0.0002	63
7	0.002	0.4	80	18	20	13230.4	0.0001	93	13547.7	0.0003	89
8	0.002	0.6	80	18	20	19133.9	0.0001	92	19635.6	0.0003	90
9	0.01	0.6	80	18	20	28124.3	0.0001	92	28839.7	0.0003	88
10	0.003	0.25	100	20	22	17943.3	0.0001	108	18571.2	0.0003	104
11	0.009	0.5	100	20	22	33516.8	0.0001	109	34775.9	0.0003	104
12	0.003	0.25	150	25	28	24219.5	0.0001	149	25188.6	0.0004	143
13	0.009	0.5	150	25	28	42577.1	0.0001	151	44374.2	0.0004	146
14	0.003	0.25	200	30	35	33847.1	0.0001	196	35446.3	0.0004	189
15	0.009	0.5	200	30	35	54730.4	0.0001	195	57546.1	0.0004	191
16	0.003	0.25	300	40	48	56277.6	0.0002	315	59285.9	0.0005	306
17	0.009	0.5	300	40	48	83964.7	0.0002	318	88947.4	0.0005	308
18	0.003	0.25	400	50	65	82751.2	0.0002	448	88085.6	0.0005	437
19	0.009	0.5	400	50	65	117271.8	0.0002	452	125307.5	0.0005	439

Table 3
Studying quality of lower bound presented in Section 3.2.4 for small instances.

No.	β	θ	# Customers	# Potential DCs	# Vehicles	Deviation (%)*
1	0.003	0.7	4	2	2	7.41
2	0.003	0.7	4	3	2	7.53
3	0.003	0.7	6	2	2	6.92
4	0.003	0.7	6	3	2	6.93
5	0.003	0.7	6	4	3	6.85
6	0.003	0.7	8	3	3	6.64
7	0.003	0.7	9	3	2	6.52
8	0.003	0.4	10	4	3	6.46
9	0.003	0.7	10	4	3	6.45
10	0.02	0.7	10	4	3	6.38

* Deviation (%) = 100 × (optimal solution value − lower bound value)/lower bound value.

Table 4
Error bounds to assess quality of solutions obtained by heuristic method.

No.	β	θ	# Customers	# Potential DCs	# Vehicles	Lower bound	Heuristic solution	
						CPU time	Error bound (%)*	CPU time
1	0.003	0.4	20	6	5	3121	6.8	18
2	0.003	0.7	20	6	5	3183	6.78	19
3	0.02	0.7	20	6	5	3318	6.82	19
4	0.003	0.4	30	9	8	8813	6.57	27
5	0.003	0.7	30	9	8	8444	6.56	27
6	0.02	0.7	30	9	8	9012	6.58	28
7	0.003	0.4	40	12	12	18,365	6.38	37
8	0.003	0.7	40	12	12	18,728	6.39	36
9	0.02	0.7	40	12	12	18,602	6.38	36
10	0.002	0.3	50	15	15	31,339	6.31	48
11	0.002	0.7	50	15	15	30,788	6.3	48
12	0.02	0.7	50	15	15	30,665	6.31	47
13	0.002	0.4	80	18	20	105,462	6.25	70
14	0.002	0.6	80	18	20	106,089	6.24	69
15	0.01	0.6	80	18	20	106,338	6.24	71
16	0.003	0.25	100	20	22	207,683	6.21	84
17	0.009	0.5	100	20	22	209,162	6.22	84
18	0.009	0.6	100	20	22	208,452	6.21	85

* Error bound (%) = 100 × (Heuristic solution value − lower bound value)/lower bound value.

Table 5
Impacts of weight factors, computational results for 40 customers.

No.	Input				Output		
	β	θ	# Potential DCs	# Vehicles	# Opened DCs	Average cost	CPU time
1	0.003	0.15	12	12	7	4305.1	54
2	0.003	0.4	12	12	6	6093.7	55
3	0.003	0.7	12	12	6	9056.1	55
4	0.003	0.7	12	12	6	9056.1	54
5	0.02	0.7	12	12	7	13252.9	56
6	0.04	0.7	12	12	7	18127.6	55

Table 6
Impacts of weight factors, computational results for 80 customers.

No.	Input				Output		
	β	θ	# Potential DCs	# Vehicles	# Opened DCs	Average cost	CPU time
1	0.002	0.2	18	20	11	10151.1	92
2	0.002	0.4	18	20	10	13230.4	93
5	0.002	0.6	18	20	9	19133.9	92
4	0.002	0.6	18	20	9	19133.9	93
5	0.01	0.6	18	20	11	28124.3	92
6	0.03	0.6	18	20	11	38215.4	94

Table 7
Impacts of weight factors, computational results for 150 customers.

No.	Input				Output		
	β	θ	# Potential DCs	# Vehicles	# Opened DCs	Average cost	CPU time
1	0.003	0.15	25	28	15	19976.4	149
2	0.003	0.25	25	28	14	24219.5	149
3	0.003	0.5	25	28	13	33127.4	150
4	0.003	0.5	25	28	13	33127.4	148
5	0.009	0.5	25	28	14	42577.1	151
6	0.02	0.5	25	28	15	54276.3	150

Table 8
Impacts of weight factors, computational results for 300 customers.

No.	Input				Output		
	β	θ	# Potential DCs	# Vehicles	# Opened DCs	Average cost	CPU time
1	0.003	0.15	40	48	29	49195.1	316
2	0.003	0.25	40	48	27	56277.6	315
3	0.003	0.5	40	48	23	68461.3	317
4	0.003	0.5	40	48	23	68461.3	317
5	0.009	0.5	40	48	25	83964.7	318
6	0.02	0.5	40	48	28	99761.2	318

Table 9
Comparison of proposed model and Shen and Qi's model, computational results for 40 customers.

					Average cost		
No.	β	θ	# Potential DCs	# Vehicles	Proposed model	Shen and Qi's model	Improvement (%)
1	0.003	0.15	12	12	4199.8	5127.9	22.10
2	0.003	0.4	12	12	5943.7	7065.2	18.87
3	0.003	0.7	12	12	8847.9	10306.1	16.48
4	0.003	0.7	12	12	8847.9	10306.1	16.48
5	0.02	0.7	12	12	12956.4	15584.4	20.28
6	0.04	0.7	12	12	17760.3	22716.3	27.90

Table 10
Comparison of proposed model and Shen and Qi's model, computational results for 100 customers.

No.	β	θ	# Potential DCs	# Vehicles	Average cost		Improvement (%)
					Proposed model	Shen and Qi's model	
1	0.002	0.2	20	22	9847.5	11629.6	18.10
2	0.002	0.4	20	22	12796.3	14812.5	15.76
3	0.002	0.6	20	22	18615.8	21082.3	13.25
4	0.002	0.6	20	22	18615.8	21082.3	13.25
5	0.01	0.6	20	22	27549.5	32245.1	17.04
6	0.03	0.6	20	22	33417.6	41169.4	23.20

Table 11
Comparison of proposed model and Shen and Qi's model, computational results for 200 customers.

No.	β	θ	# Potential DCs	# Vehicles	Average cost		Improvement (%)
					Proposed model	Shen and Qi's model	
1	0.003	0.15	30	35	27933.4	31923.3	14.28
2	0.003	0.25	30	35	32933.2	36961.6	12.23
3	0.003	0.5	30	35	43274.6	47953.7	10.81
4	0.003	0.5	30	35	43274.6	47953.7	10.81
5	0.009	0.5	30	35	53375.9	60806.5	13.92
6	0.02	0.5	30	35	70149.8	83767.1	19.41

From Table 2, it can be seen that the quality obtained from the heuristic based on the TS–SA algorithm is better than the heuristic based on the SA algorithm.

5.3. Computing error bounds

In this section we analyze our heuristic method by presenting error bounds for the solutions obtained by our proposed heuristic method for relatively large instances with uncapacitated distribution centers. First we have provided some numerical results in Table 3 to assess the quality of the lower bounds obtained from the model proposed in Section 3.2.4, i.e. (14), (2)–(10). In Table 3 we can see the deviation percentage moderately decreases as the number of the customers increases, and the average of the deviation percentages is 6.83%.

After studying the quality of the lower bound for small instances, we have solved several instances and computed the associated error bounds based on the proposed lower bound. The results are given in Table 4. It could be seen that the average of the error bounds is 6.43% which is less than 6.83%. The results show that the error bound percentages decrease as the number of the customers increases, and the behavior of the heuristic is stable under various problem parameters. This shows that the quality of the solutions obtained by the heuristic method is promising.

5.4. Impacts of weight factors

In Tables 5–8 we give several computational results to study the impacts of the weight factors θ and β on the number of the optimal opened distribution centers. First, we can see that the number of the opened distribution centers increases when the weight of the transportation (β) cost increases. Second, the number of the opened distribution centers decreases when the weight of the inventory cost (θ) increases. As a result, it can be anticipated that the impact of the fill rate α is similar to θ.

5.5. Benefit of new model

In this paper, we develop the model of Shen and Qi (2007) by considering vehicle routing decisions. Now we numerically show that the resulting decisions from the new model can save a considerable amount of cost. In Tables 9–12 we present the cost (including location, inventory and routing costs) associated with the decisions determined by our model and Shen and Qi's model for several instances. Because Shen and Qi's model determines only the location–allocation and inventory decisions, we should determine routing decisions after solving their model. To this end, first we apply Shen and Qi's model to make location–allocation and inventory decisions, and then we calculate the exact vehicle routing cost from each opened distribution center to its assigned customers. We apply a technique similar to that described in Section 3.1 to obtain the optimal solution of Shen and Qi's model.

From Tables 9–12, we see that the expected saving achieved by our model is big compared to Shen and Qi's model. The ranges of improvement are: 16.48–27.9% for 40 customers, 13.25–23.2% for 100 customers, 10.81–19.41% for 200 customers, and 9.78–18.05% for 400 customers.

Table 12
Comparison of proposed model and Shen and Qi's model, computational results for 400 customers.

No.	β	θ	# Potential DCs	# Vehicles	Average cost		Improvement (%)
					Proposed model	Shen and Qi's model	
1	0.003	0.15	50	65	71122.6	79975.3	12.45
2	0.003	0.25	50	65	80525.8	89164.1	10.73
3	0.003	0.5	50	65	95248.1	104567.2	9.78
4	0.003	0.5	50	65	95248.1	104567.2	9.78
5	0.009	0.5	50	65	113762.9	128019.3	12.53
6	0.02	0.5	50	65	139543.4	164732.7	18.05

6. Conclusions

Design of a supply chain distribution network consists of three major problems: distribution center location problem, vehicle routing problem and inventory control problem. In the literature, there are three problems considering the integration of two of the above problems: inventory–routing problem, location–routing problem and location–inventory problem. Recently, Shen and Qi (2007) presented an inventory–location model with the objective function including the location-inventory cost and an approximate routing cost which depend only on the locations of the opened distribution centers. They showed that significant cost saving can be obtained by their model instead of sequential approach. However, their model cannot determine the routing decisions; additionally, there is no guarantee that their model can be exploited for real-world cases since the approximation of routing cost is appropriate only under some assumptions, and there is no error analysis for the approximation.

The main contribution of this paper to the literature is to simultaneously optimize location, allocation, capacity, inventory and routing decisions without any approximation. We show that the expected saving achieved by our model is big compared to the model in Shen and Qi (2007). The range of improvement is 9.78–27.90% for different problem sizes.

We show that our model can be reformulated as a mixed integer convex program. Also, we present an effective heuristic method. The heuristic method is decomposed into two stages: constructive stage, where an initial solution is built at random, and improvement stage, where the solution is iteratively improved in two phases: location phase and routing phase. In the improvement stage, a hybrid algorithm based on Tabu Search and Simulated Annealing is used to improve the current solution in each phase. The computational results indicate that the heuristic method is effective for a wide variety of problem sizes and structures. Also, we study the model under extra constraints such as stochastic capacity constraints.

Afterward, we investigate the impacts of the objective weight factors associated with the inventory and routing costs on the number of the opened distribution centers. We observe that, when we increase the weight factor of the routing cost, the number of the opened distribution centers increases, and when we increase the weight factor of the inventory cost, the number of the opened distribution centers decreases.

For future work, it is interesting to develop more effective and elegant heuristic methods to solve the model. Moreover, the model can be extended in several realistic and practical directions.

References

Adelman, D., 2004. A price-directed approach to stochastic inventory/routing. Operations Research 52, 499–514.

Baita, F., Ukovich, W., Pesenti, R., Favaretto, D., 1998. Dynamic routing-and-inventory problem: a review. Transportation Research: Part A 32, 585–598.

Bertsimas, D., Tsitsiklis, J., 1993. Simulated annealing. Statistical Science 8, 10–15.

Daskin, M., Coullard, C., Shen, Z.J., 2002. An inventory–location model: formulation, solution algorithm and computational results. Annals of Operations Research (2002), 83–106.

Day, J.M., Wright, P.D., schoenherr, T., Venkataramanan, M., Gaudette, K., 2009. Improving routing and scheduling decisions at a distributor of industrial gasses. Omega 37, 227–237.

Desrochers, M., Laporte, G., 1991. Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. Operations Research Letters 10, 27–36.

Erlebacher, S.J., Meller, R.D., 2000. The interaction of location and inventory in designing distribution systems. IIE Transactions 32, 155–166.

Gaur, V., Fisher, M.L., 2004. A periodic inventory routing problem at a supermarket chain. Operations Research 52, 813–822.

Geman, S., Geman, D., 1984. Stochastic relaxation gibbs distribution and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6, 721–741.

Jaillet, P., Bard, J., Huang, L., Dror, M., 2002. Delivery cost approximations for inventory routing problems in a rolling horizon framework. Transportation Science 3, 292–300.

Kleywegt, A., Nori, V.S., Savelsbergh, M.W.P., 2002. The stochastic inventory routing problem with direct deliveries. Transportation Science 36, 94–118.

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

Miranda, P.A., Garrido, R.A., 2004. Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. Transportation Research Part E 40, 183–207.

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

Mitra, D., Romeo, F., Vincentelli, A.S., 1986. Convergence and finite-time behavior of simulated annealing. Advances in Applied Probability 18, 747–771.

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

Oppen, J., Loketangen, A., 2008. A Tabu search approach for the livestock collection problem. Computers and Operations Research 35, 3213–3229.

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

Shen, Z.J., 2005. A multi-commodity supply chain design problem. IIE Transactions 37, 753–762.
Shen, Z., Qi, L., 2007. Incorporating inventory and routing cost in strategic location models. European Journal of Operational Research 179, 372–389.
Spall, J., 2003. Introduction to Stochastic Search and Optimization. John Wiley & Sons, New Jersey.
Yu, Y., Chen, H., Chu, F., 2008. A new model and hybrid approach for large scale inventory routing problems. European Journal of Operational Research 189, 1022–1040.
Zhao, Q., Chen, S., Zhang, C., 2008. Model and algorithm for inventory/routing decision in a three-echelon logistics system. European Journal of Operational Research 191, 623–635.