ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 3, No. 4, 2008, *pp*. 290-304

A hybrid Tabu-SA algorithm for location-inventory model with considering capacity levels and uncertain demands

Nader Azad [1], Hamid Davoudpour

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

(*Received January 28, 2008, accepted July 18, 2008*)

Abstract. In this paper, we present a complex distribution network design problem in supply chain system which includes location and inventory decisions. Customers' demand is generated randomly and each distribution center maintains a certain amount of safety stock in order to achieve a certain service level for the customers it services. Unlike most of past research, our model allows for multiple levels of capacities available to the distribution centers. This consideration helps to achieve the capacity utilization to a high level as our computational results show this fact.

We show that this problem can be formulated as a non-linear integer programming model. A hybrid heuristic combining Tabu search with Simulated Annealing (SA) sharing the same tabu list is developed for solving the problem. We comprise the hybrid algorithm with the optimal solution, Simulated Annealing algorithm and Tabu search algorithm. The results indicate that the method is efficient for a wide variety of problem sizes.

Keywords: facility location, inventory, integrated supply chain design, capacity level, simulated annealing, tabu search.

1. Introduction

All companies that aim to be competitive on the market have to pay attention to their organizations related to the entire supply chain. In particular, companies have to analyze the supply chain in order to improve the customer service level without an uncontrolled growth of cost. In few words, companies have to increase the efficiency of their logistics operations.

Traditionally, supply chain models have focused on either the strategic aspects of supply chain design or the tactical aspects, but not both simultaneously. Ozen [11] shows that cost saving can be obtained by considering location and inventory decisions simultaneously instead of the sequential approach, where location decisions are made before the inventory decisions.

For a thorough review of facility location problem see Hamacher and Drezner [6], Barmel and Simchi-Levi [2], and Daskin [3].

Recently some authors have incorporated inventory control decisions into the facility location problem, Simchi-Levi [16] considers a hierarchical planning model for stochastic distribution system, in which the locations and demand of customers are determined according to some probability distribution. Different decisions are grouped into three classes: strategic planning, tactical planning, and operational control. Jayaraman [7] incorporates the inventory costs into a facility location problem, assuming fixed lot sizes and deterministic demands. Nozick and Turnquist [10] incorporate inventory costs assuming the demands to arrive in a Poisson manner and a base stock inventory policy. Barahonal and Jensen [1] solve a location problem with a fixed cost for stocking a given product at a distribution center.

Shen [12], studied the joint location-inventory model in which location, shipment and non-linear safety stock inventory costs are included in the same model. Teo et.al [18] present an approximation algorithm for the problem of choosing distribution centers to minimize location and inventory costs, ignoring transportation cost. Vidal and Geotschalckx [19] study a global supply chain design model that maximizes

[1] Corresponding Author., Tel : +989122253912
Email Address: ndr_azad@yahoo.com (Nader Azad)

the after-tax profits of a multi-national corporation. Their model simultaneously considering transferring prices, transportation cost allocation, inventory costs and their impact on the selection of international transportation modes. Erlebacher and Meller [5] formulate a non-linear integer location-inventory model. They use a continuous approximation for solving the problem. Daskin et.al [4] apply lagrangian relaxation to solve the location-inventory model. Shen et.al [14] present a location-inventory model that is similar to the model of Shen [12] and use column generation for solving the problem. Miranda and Garrido [9] present a location-inventory model that is similar to the model of Daskin et.al [4], and they apply lagrangian relaxation method for solving the model. Shu et.al [15] study a more general location – inventory model and use column generation for solving the problem. Shen [13] propose multi-commodity location-inventory model and use column generation for solving the problem.

In this paper we present an integrated stochastic supply chain design model which optimizes location and inventory decisions, simultaneously. One major drawback in most of past research is that they assume the capacity of distribution centers are known and assignment of customers to the distribution centers are limited by the capacity of distribution centers. In our model, we use different capacity levels for distribution centers that make the problem more realistic and assignment of customers to the distribution centers more flexible.

We assume each customer has uncertain demand that follows a given normal distribution. The goal of our model is to choose a set of distribution centers to serve the customers and allocate customers to the opened distribution centers, and to determine the inventory policy based on the information of customers' demand in order to minimize the total expected cost of locating distribution centers, shipment, transportation and inventory costs, while ensuring a pre-specified level of service.

One of the possible objectives in supply chain design models is to maximize the capacity utilization of distribution centers. Our results show that use of capacity levels for distribution centers increase the capacity utilization to a high level.

The reminder of this paper is organized as follows. In section 2, mathematical formulation of the problem is presented. In section 3, the hybrid Tabu-SA algorithm is developed for solving the problem. Section 4, discusses some computational results. Finally, section 5 contains some conclusions and future research development.

2. Model formulation

In our model we consider the following assumptions:

We assume that each customer has uncertain demand that follows a normal probability distribution, and customers' demands are independent.

We assume to know capacity levels for distribution centers, and the company pays a fixed location cost for opening a distribution center with a capacity level.

We assume that the company pays a fixed cost for each order placed at a distribution center and a holding cost for inventory at each distribution center. The distribution centers hold working inventory and safety stock inventory intended to buffer the system against stock out during ordering lead times.

The distribution centers are assumed to follow a (Q, R) inventory policy.

Before presenting the model, let us introduce the notations that will be used throughout the paper:

Index sets

K : Set of customers.

J : Set of potential distribution centers.

N : Set of capacity levels available to the potential distribution centers.

Parameters and notations

μ_k : Mean daily demand at customer k , ($\forall k \in K$)

σ_k^2 : Variance of daily demand at customer k , ($\forall k \in K$)

F_J^n : Fixed cost for opening and operating distribution center j with capacity level n, ($\forall j \in J, \forall n \in N$)

b_j^n : Capacity with level n for the potential distribution center j, ($\forall j \in J, \forall n \in N$)

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, ($\forall j \in J$)

L_j : Distribution center j lead time in days, ($\forall j \in J$)

g_j : Fixed cost per shipment from the supplier to distribution center j, ($\forall j \in J$)

a_j : Cost per unit of a shipment from the supplier to potential distribution center j, ($\forall j \in J$)

c_{jk} : Cost per unit of shipment from distribution center j to customer k ($\forall j \in J, \forall k \in K$)

α : Desired percentage of customer orders satisfied (fill rate).

z_α : Standard normal deviate such that $p(z \le z_\alpha) = \alpha$.

χ : Days per year.

Decision variables

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

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

Objective function

The objective function minimizes the following costs:

The fixed cost of locating distribution centers with capacity level, given by the term: $\sum_{j \in J} \sum_{n \in N} F_j^n U_j^n$

The annual shipment cost from distribution centers to the customers, given by the term: $\sum_{j \in J} \sum_{k \in K} \chi c_{jk} \mu_k Y_{jk}$

The expected annual inventory cost: is the sum of working inventory and safety stock costs.

Working inventory representing product that has been ordered from the supplier but not yet requested by the customers and safety stock cost inventory intended to buffer the system against stock outs during ordering lead times.

We assume that distribution centers orders inventory from the supplier using a (Q, R) policy with service level constraints. Working inventory cost includes the fixed costs of placing orders as well as the shipment costs from the supplier to the distribution centers as well a the holding costs of working inventory.

Let D_j denote the total annual demand going through distribution center j ($D_j = \sum_k \mu_k Y_{jk}$) and w number of shipments per year from the supplier. Then the total annual cost of ordering inventory from the supplier to distribution center j is given by:

$$P_j w + \left(g_j + a_j \frac{D_j}{w} \right) \times w + h_j \frac{D_j}{2w} \tag{1}$$

We assume that cost of shipping an order of size M form the supplier to distribution center j is given by the term of: $(g_j + a_j \times M)$.

The first term of (1) represents the total fixed cost of placing w orders per year. The second term represents the delivery cost from the supplier to distribution center j per year, and the third term is the cost of average working inventory.

By taking the derivative of this expression with respect to w, we obtain the optimal value of w is given by:

$$w=\sqrt{\frac{h_j D_j}{2(P_j+g_j)}} \tag{2}$$

By substituting (2) into the total cost (1), we obtain the optimal value of total annual working inventory associated with distribution center j is given by:

$$\sqrt{2 h_j D_j (P_j+g_j)} + a_j D_j = \sqrt{2 h_j (P_j+g_j)\sum_k \chi \mu_k Y_{jk}} + a_j \sum_k \chi \mu_k Y_{jk} \tag{3}$$

The yearly safety stock cost at distribution center j is given by: $\theta \quad h_j z_\alpha \sigma_j$ and $\sigma_j = \sqrt{L_j \sum_k \sigma_k^2 Y_{jk}}$

The holding cost for the safety stock at distribution center j is: $h_j z_\alpha \sqrt{L_j \sum_k \sigma_k^2 Y_{jk}}$

The expected annual inventory cost is given by the term:

$$\sum_{j\in J}\left[\sqrt{2 h_j (P_j+g_j)\sum_k \chi \mu_k Y_{jk}} + a_j \sum_k \chi \mu_k Y_{jk} + h_j z_\alpha \sqrt{L_j \sum_k \sigma_k^2 Y_{jk}} \right]$$

The problem formulating is as follows:

$$MIN: \quad \sum_{j\in J}\sum_{n\in N} F_j^n U_j^n + \sum_{j\in J}\sum_{k\in K} \chi c_{jk} \mu_k Y_{jk}$$

$$+\sum_{j\in J}\left[\sqrt{2 h_j (P_j+g_j)\sum_k \chi \mu_k Y_{jk}} + a_j \sum_k \chi \mu_k Y_{jk} + h_j z_\alpha \sqrt{L_j \sum_k \sigma_k^2 Y_{jk}} \right] \tag{4}$$

Subject to:

$$\sum_{j\in J} Y_{jk} = 1 \qquad \forall k \in K \tag{5}$$

$$\sum_{n\in N} U_j^n \le 1, \qquad \forall j \in J \tag{6}$$

$$\sum_{k\in K} \chi \mu_k Y_{jk} \le \sum_{n\in N} b_j^n U_j^n \qquad \forall j \in J \tag{7}$$

$$Y_{jk} \in \{0,1\} \qquad \forall j \in J, \forall k \in K$$
$$U_j^n \in \{0,1\} \qquad \forall j \in J, \forall n \in N \tag{8}$$

The model minimizes the total expected costs made of: the fixed cost for opening distribution centers, the annual shipment cost from distribution centers to the customers, and the expected annual inventory cost. Constraints (5) ensure that each customer is assigned to exactly one distribution center. Constraints (6) ensure that each distribution center can be assigned at most one capacity level. Constraints (7) are the capacity constraints associated with the distribution centers. Constraints (8) enforce the integrality restrictions on the binary variables.

3. Solution approach

A hybrid heuristic combining Simulated Annealing with Tabu Search sharing the same tabu list is used for solving the problem. In section 3.1, 3.2 we describe the SA algorithm and Tabu search algorithm, respectively, and in section 3.3, we describe the hybrid Tabu-SA algorithm which we use for solving the problem.

3.1. Simulated annealing algorithm

Simulated annealing (SA) is one of the novel algorithms which was initially presented by Kirkpatrick et.al [8]. The SA methodology draws its analogy from the annealing process of solids. In the annealing process, a solid is heated to a high temperature and gradually cooled to a low temperature to be crystallized. As the heating process allows the atoms to move randomly, if the cooling is done too rapidly, it gives the atoms enough time to align themselves in order to reach a minimum energy state that named stability or equipment. This analogy can be used in combinatorial optimization in which the state of solid corresponds to the feasible solution, the energy at each state corresponds to the improvement in the objective function and the minimum energy state will be the optimal solution.

The Steps of SA algorithm are shown in Fig.1.

$X_{best} = \phi$

Select an initial solution, X_0

$X_{best} = X_0$, $X = X_0$

While ($T_0 < ST$) Do

$S = 0$

While ($S < L$) Do

Generate solution X_n in the neighborhood of X ,

$\Delta C = C(X_n) - C(X)$

If $\Delta C \leq 0$ then

$X = X_n$

$S = S + 1$

If $C(X_n) < C(X_{best})$ then

$X_{best} = X_n$

End If

Else

Generate $y \rightarrow U(0,1)$ Randomly

Set $z = e^{-\frac{\Delta C}{T_0}}$

If $y < z$ then

$X = X_n$

$S = S + 1$

End If

End If

End While

$T_0 = C \times T_0$

End While

Fig.1. SA algorithm

The SA parameters are as follows:

T_0 : Initial temperature,

C : Rate of the current temperature decreases (cooling schedule),

ST : Freezing temperature (the temperature at which the desired energy level is reached),

L : Number of accepted solution at each temperature,

S : Counter for the number of accepted solution at each temperature,

X : A feasible solution

$C(X)$: The value of objective function for X ,

SA uses a stochastic approach to direct 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 feature, can allow it to prevent falling in the local optimum trap. SA 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, repeated while $S < L$, a neighboring solution X_n of the current solution X is generated. If this move decreases the objective function, or leaves it unchanged, then the move is always accepted. Moves, which increase the objective function value, are accepted with a probability $e^{-\Delta c / T_0}$ to allow the search to escape a local optimum. The value of the temperature decreases in each iteration of the outer cycle of the algorithm. Obviously the probability of accepting worst solution decreases as the temperature decreases in each outer cycle. Two important issues that need to be defined when adopting this general algorithm to a specific problem are the procedures to generate both initial solution and neighboring solutions.

3.2. Tabu Search algorithm

The overall approach in Tabu search algorithm is to avoid entrainment in cycles by forbidding or penalizing moves which take the solution, in the next iteration, to points in the solution space previously visited. Tabu search uses a local or neighborhood search procedure to iteratively move from a solution X to a solution X_n in the neighborhood of X , until some stopping criterion has been satisfied. To explore regions of the search space that would be left unexplored by the local search procedure, tabu search modifies the neighborhood structure of each solution as the search progresses. The solutions admitted to $N^*(X)$, the new neighborhood, are determined through the use of special memory structures. The search then progresses by iteratively moving from a solution X to a solution X_n in $N^*(X)$.

Perhaps the most important type of short-term memory to determine the solutions in $N^*(X)$; also, the one that gives its name to tabu search, is the use of a tabu list. Tabu list contains the solutions that have been visited in the recent past (less than m moves ago, where m is the tabu tenure). Solutions in the tabu list are excluded from $N^*(X)$.

The steps of tabu search are shown in Fig.2.

$X_{best} = \phi$

Select an initial solution, X_0

$X_{best} = X_0$, $X = X_0$

$r = 0$

While ($r < G$) Do

Generate solution X_n in the neighborhood of X ,

$\Delta C = C(X_n) - C(X)$

If the candidate move is in the tabu list, then

If $C(X_n) < C(X_{best})$ then

$X = X_n, r = r+1$

$X_{best} = X_n$, update the tabu list

Else

Generate another solution X_n in the neighborhood of X

End If

Else

If $\Delta C \leq 0$ then

$X = X_n$, $r = r+1$

Update the tabu list

If $C(X_n) < C(X_{best})$ then

$X_{best} = X_n$

End If

Else

Generate another solution X_n in the neighborhood of X

End If

End If

End While

Fig.2. Tabu Search algorithm

3.3. The proposed hybrid algorithm

A hybrid heuristic combining Simulated Annealing with Tabu Search sharing the same tabu list is used for solving the problem.

The steps of proposed hybrid Tabu-SA based heuristic are as follows:

Step1: $X_{best} = \phi$, select an initial solution, (X_0)

$X_{best} = X_0$, $X = X_0$

Step2: Generate solution X_n in the neighborhood of X .

Step3: Is the candidate move in the tabu list? If yes, go to Step 4. Otherwise go to Step 5.

Step4: If $C(X_n) \le C(X_{best})$ then $X = X_n$, $X_{best} = X_n$, update the tabu list and go to Step 6, otherwise go to step 2 for choosing another candidate move.

Step5: Let $\Delta C = C(X_n) - C(X)$.

5.1. If $\Delta C \le 0$ then $X = X_n$, $r = r+1$, update the tabu list and if $C(X_n) < C(X_{best})$ then $X_{best} = X_n$.

5.2. If $\Delta C > 0$ then $y \to U(0,1)$, $z = e^{-\frac{\Delta C}{T_0}}$. If $y < z$ then $X = X_n$.

Step6: Should the procedure stop under temperature T_0? If yes, go to Step 7, otherwise go to Step 2.

When the number of accepted solutions under temperature T_0 reaches to a predefined value, the following condition should be checked:

$$\frac{|AOV_c - AOV_b|}{AOV_b} \le \varepsilon$$

where AOV_c is the average objective value of accepted solutions under the temperature T_0, AOV_b is the average objective value of accepted solutions before the temperature T_0, ε is a predefined equilibrium value ($0 < \varepsilon < 1$). If the above condition is satisfied, the procedure stops under temperature T_0. This condition was proposed by Skiscim & Golden [17].

Step7: $T_0 = C \times T_0$.

Step8: Is the stopping criterion ($T_0 < ST$) matched? If yes, stop, otherwise, go to Step 2.

The steps of proposed hybrid Tabu-SA Algorithm are shown in Fig.3.

$X_{best} = \phi$

Select an initial solution, X_0

$X_{best} = X_0$, $X = X_0$

While ($T_0 < ST$) Do

While ($\frac{|AOV_c - AOV_b|}{AOV_b} \le \varepsilon$) Do

Generate solution X_n in the neighborhood of X ,

$\Delta C = C(X_n) - C(X)$

If the candidate move is in the tabu list, then

If $C(X_n) < C(X_{best})$ then

$X = X_n$, $X_{best} = X_n$, update the tabu list

Else

Generate another solution X_n in the neighborhood of X

End If

Else

If $\Delta C \le 0$ then

$X = X_n$, update the tabu list

If $C(X_n) < C(X_{best})$ then

$X_{best} = X_n$
End If
Else
Generate $y \rightarrow U(0,1)$ Randomly
Set $z = e^{-\frac{\Delta C}{T_0}}$
If $y < z$ then
$X = X_n$, update the tabu list
End If
End If
End If
End While
$T_0 = C \times T_0$
End While

Fig.3. Proposed hybrid Tabu-SA Algorithm

In the section 3.3.1, 3.3.2, we describe the initial solution construction and different types of move for generating the candidate move which we use for hybrid Tabu-SA algorithm.

3.3.1. Initial solution construction

For obtaining the initial solution, first we assign customers to the distribution centers, randomly. For each of the opened distribution center the capacity level is selected randomly. The procedure for obtaining the initial solution is as follows.

Step1: Put customers into a set K.

Step2: 1- Select a customer from K randomly. 2- Delete the customer from K.

Step3: Select a distribution center randomly.

Step4: If we select this distribution center for the first time then select a capacity level for this distribution center randomly.

Step5: If remaining capacity of the distribution center is greater than the demand of the customer then assign the customer to the distribution center and go to Step 6 otherwise go to Step 3 for selecting another distribution center.

Step6: Is K empty? If yes, stop, otherwise go to Step 2.

3.3.2. Improving the initial solution

In this phase, the main objective is to improve the initial solution. We apply four different types of move for generating a candidate move: mov1, mov2, mov3, mov4. We generate a candidate move (from X_0 to the candidate solution X_n) using one of the four moves randomly.

Mov1: Randomly, one of the opened distribution centers (a_j) is closed and all of the customers are reallocated among the remaining opened distribution centers. If the remaining capacities of the opened distribution centers are not enough for the customers of a_j then we randomly select an open distribution center and increase its capacity level to one higher level. The procedure of mov1 is as follows:

Step1: Select an open distribution center randomly (a_j). Let D_j be the set of customers that assigned to a_j.

Step2: Select a customer (k) from D_j, randomly.

Step3: Determine the opened distribution centers that have enough remaining capacity for demand of customer k. Let O_k be the opened distribution centers which have enough remaining capacity.

Step4: If O_k is empty, (we can not find the opened distribution center that have enough remaining capacity) then go to Step 5, otherwise go to step 8.

Step 5: If all of the opened distribution centers have the highest capacity level. Then stop and exit from this move and select a move randomly for generating a candidate move, otherwise go to Step 6.

Step6: Randomly select one of the opened distribution centers.

Step7: If this distribution center has the highest capacity level then go to Step 6 for selecting another open distribution center, otherwise increase its capacity level to one higher level and assign customer k to this distribution center and go to Step 9.

Step8: Select a distribution center from O_k randomly and assign customer k to this distribution center.

Step9: Delete the customer k from D_j.

Step10: Is D_j empty? If yes, close a_j and stop, otherwise go to Step 2.

In Step 5 of the mov1 by the time mov1 is not performed as many number as max-mov1, we give up mov1 and we do not apply mov1 during the algorithm, note that max-mov1 is as an input for the heuristic method. In fact, mov1 is terminated when a max-mov1 number of moves are not performed based on Step 5.

Mov2: In this move we select two open distribution centers randomly, (a_i, a_j), and exchange a_i and a_j. In this move capacities of a_i and a_j are checked for serving the customers.

Mov3: One of the opened distribution centers (a_i) is closed randomly, and a closed distribution center (a_j) is opened randomly and one of the capacity levels is selected for a_j randomly. Then we assign all of the customers corresponding to the eliminated distribution center (a_i) to the new opened distribution center (a_j). In this move the capacity of a_j is checked for serving the customers.

Mov4: Select two open distribution centers, randomly, (v_i, v_j). Then randomly select a customer (c_i) in v_i and a customer (c_j) in v_j and exchange c_i and c_j. In this move we must check the capacities of distribution centers.

4. Computational results

The computational experiments described in this section were designed to evaluate the performance of our overall solution procedure with respect to a series of test problems.

It was coded in visual basic 6 and run on a Pentium 4 with 2.8 GB processor. The daily demand of the customers was drawn from a uniform distribution between 1 and 5. The variances of daily demands of customers were drawn from a uniform distribution between 1 and 3. Also we use the following parameter values.

h_j Is uniformly drawn from [2, 4]

p_j Is uniformly drawn from [15, 20]

L_j Is uniformly drawn from [6, 10]

g_j Is uniformly drawn from [15, 20]

a_j Is uniformly drawn from [2, 5]

c_{jk} Is uniformly drawn from [2, 5]

$\chi = 250$, $z_\alpha = 1.96$ (97.5% service level)

Four capacity levels are used for the capacities of the potential distribution centers. If we let D

represents total demand requirements ($D=\sum_{k\in K}\mu_k$) and $|J|$ be the number of potential distribution centers and c_j is a random number between 0.8 and 1.2 for each distribution center. Then we define for each distribution center: $cap(j)=\left[c_j\times\frac{D}{|J|}\right]$, ($[A]$ is the integer part of A), so the different capacities of the potential distribution center j are computed as follows.

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

Fixed set up costs of locating and operating distribution centers are as follows. Let k_j for each distribution center were drawn from a uniform distribution between 4500 and 5500. Then fixed set up cost for distribution center j is computed as follows.

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

Our goal in this section is to find out (1) performance of the heuristic algorithm, and (2) what the benefit of considering capacity levels is and how impact to the capacity utilization and how capacity utilization varies with the changing number of capacity levels.

4.1. Performance of the hybrid algorithm

4.1.1. Comparison of optimal solution and hybrid algorithm

Table 1Comparison of optimal solution and hybrid algorithm

			Optimal Solution		Hybrid Algorithm		
NO.	# Customers	# DCs	Cost	CPU time	Cost	CPU time	Gap(%)
1	4	2	18095.3	3	18095.3	1	0.00
2	6	3	24726.3	6	24726.3	3	0.00
3	7	3	28546.7	11	28546.7	5	0.00
4	8	4	31586.5	16	31586.5	9	0.00
5	9	4	34182.4	27	34182.4	11	0.00
6	20	6	77710.2	117	77710.2	26	0.00
7	30	8	106483.1	178	106483.1	35	0.00
8	40	12	139745.8	291	141049.8	47	0.93
9	50	15	174343.3	435	176076.7	61	0.99
10	60	17	200685.1	984	203097.6	74	1.20
11	70	19	241089.4	1824	243865.2	89	1.15
12	80	21	286329.2	2841	289993.4	104	1.28
13	90	23	324514.1	4981	329187.5	120	1.44
14	100	25	355169.4	8493	360789.7	137	1.58
15	120	30	439584.2	3 hours limit	421216.4	172	
16	150	35	542596.7	3 hours limit	508769.2	221	
17	180	38	679879.1	3 hours limit	611297.5	272	

Gap: Gap from optimal solution (%).

For evaluating the proposed heuristic, seventeen problems are solved by LINGO.8 software (table.1.). The two non-linear terms of the objective function (4) (the annual working inventory cost and the holding cost for safety stock) are the concave terms, and the other terms of the objective function are linear, so the objective function (4) is absolutely concave and the problem is to minimize the concave integer programming model. So the solution which is obtained by branch & bound methods (such as LINGO software) is optimal and we can compare our heuristic solution by optimal solution which is obtained by LINGO.8. Parameter tuning is a matter of serious concern for any optimization problem as it induces good performances. The parameters affect the working of the hybrid algorithm, drastically. For each problem, the tuning of the parameters is done by carrying out random experiments. It can be seen that the solutions of the proposed hybrid algorithm are optimal (or near optimal) in different problems (table.1.). The average CPU time are less than or equal to 137 seconds for proposed hybrid algorithm (CPU times are in the seconds).

However, the maximal average CPU time for obtaining the optimal solutions is equal to 8493 seconds, and for instance 15 to 17 by a 3 hours time limit, LINGO can not find the optimal solution, and the hybrid algorithm in this instance is better than the best solutions that are obtained by LINGO.

4.1.2. Comparison of hybrid algorithm with SA algorithm and Tabu search algorithm

In this section, we compare our hybrid algorithm with SA method and Tabu search method. The procedure for obtaining initial solution and candidate move, we use in SA method and tabu search method, are the same to the procedure of obtaining solution and candidate move in hybrid algorithm. In SA algorithm and Tabu search algorithm, for each problem, the tuning of the parameters is done by carrying out random experiments

The comparison of hybrid algorith with SA algorithm and Tabu search are shown in table 2. It can be seen that the solution quality in hybrid algorithm is better than the solution quality in SA algorithm and Tabu Search algorithm.

Table 2 comparison of hybrid algorithm with SA Algorithm and Tabu Search Algorithm

			Hybrid Algorithm		SA Algorithm		Tabu Algorithm	
NO.	# Customers	# DCs	Cost	CPU Time	Cost	CPU Time	Cost	CPU Time
1	30	8	106483.1	35	107533.2	34	107549.2	34
2	40	12	141049.8	47	142461.2	45	142496.3	46
3	50	15	176076.7	61	178719.6	58	179249.6	59
4	60	17	203097.6	74	206742.8	71	207348.3	72
5	70	19	243865.2	89	249242.5	86	249465.3	87
6	80	21	289993.4	104	297542.7	100	296948.8	101
7	90	23	329187.5	120	337074.6	115	337897.1	118
8	100	25	360789.7	137	371847.8	132	370586.7	136
9	120	30	421216.4	172	436812.6	164	437594.1	169
10	150	35	508769.2	221	520846.7	214	521057.4	216
11	180	38	611297.5	272	628150.4	264	630202.9	269
12	200	40	695327.5	308	618147.7	300	621643.1	305
13	230	43	814384.1	361	845488.8	352	848724.3	357
14	250	45	900196.4	400	942587.3	391	947381.4	398
15	280	48	1020382.8	454	1074527.1	444	1082176.4	450
16	300	50	1112395.6	494	1177196.4	482	11991237	486

4.2. Impact of considering capacity levels

In this section, we study the benefit of considering capacity levels on the capacity utilization, and we show that how capacity utilization varies with the changing number of capacity levels.

For comparing the capacity utilization by varying the number of capacity levels, we consider three different number of capacity levels. n = 4, 5, 6. The capacities and fixed costs of four capacity levels were described in the section 4. For five capacity levels we use the capacities and fixed costs as follows.

$$b_j^1 = cap(j)\ ,\ b_j^2 = 1.25\times cap(j)\ ,\ b_j^3 = 1.5\times cap(j)\ ,\ b_j^4 = 2\times cap(j)\ ,\ b_j^5 = 2.5\times cap(j)$$

$$F_j^1 = \left[0.65\times k_j\right],\ F_j^2 = \left[0.78\times k_j\right],\ F_j^3 = \left[0.9\times k_j\right],\ F_j^4 = \left[1.1\times k_j\right],\ F_j^5 = \left[1.35\times k_j\right]$$

For six capacity levels we use:

$$b_j^1 = cap(j)\ ,b_j^2 = 1.25\times cap(j)\ ,b_j^3 = 1.5\times cap(j)\ ,b_j^4 = 1.75\times cap(j)\ ,b_j^5 = 2\times cap(j),$$
$$b_j^6 = 2.5\times cap(j)$$

$$F_j^1 = \left[0.65\times k_j\right],\ F_j^2 = \left[0.78\times k_j\right],\ F_j^3 = \left[0.9\times k_j\right],\ F_j^4 = \left[1\times k_j\right],\ F_j^5 = \left[1.1\times k_j\right],\ F_j^6 = \left[1.35\times k_j\right]$$

Table.3. shows the computational results. By varying the number of capacity levels, we can see clearly the impact of number of capacity levels on the average capacity utilization of the opened distribution centers.

Table 3 Impact of considering capacity level

NO.	# Customers	# DCs	A.C.U n=4	A.C.U n=5	A.C.U n=6
1	30	8	88	88	88
2	40	12	89	89	89
3	50	15	88	88	92
4	60	17	88	88	93
5	70	19	87	90	93
6	80	21	90	92	94
7	90	23	92	92	92
8	100	25	90	93	96
9	120	30	89	93	95
10	150	35	95	95	95
11	180	38	94	96	96
12	200	40	93	93	97
13	230	43	94	96	97
14	250	45	96	97	98
15	280	48	96	98	98
16	300	50	95	96	98

A.C.U: Average capacity utilization for the opened distribution centers (%)

We observe that average capacity utilizations for the opened distribution centers for the different number of capacity levels are very high, also we observe that, when we increase number of capacity levels, the average capacity utilizations for the opened distribution centers also increase.

5. Conclusions

In this paper we have outlined an integrated stochastic supply chain design model which optimizes location, inventory and transportation decisions. We assumed each customer has uncertain demand that follows a normal distribution. The goal of our model was to choose a set of distribution centers to serve the customers and allocate customers to the opened distribution centers, and to determine the inventory policy based on the information of customers' demand in order to minimize the total expected cost, also, we used different capacity levels for distribution centers that make the problem more realistic and assignment of customers to the distribution centers more flexible.

A hybrid heuristic combining Tabu search with Simulated Annealing sharing the same tabu list was developed for solving the problem. We comprised the hybrid algorithm with the optimal solution, SA algorithm and Tabu search method. The results of extensive computational tests indicated that the hybrid algorithm is both effective and efficient for a wide variety of problem sizes. Also, we showed that consideration of considering capacity levels helps to achieve the capacity utilization to a high level. For future works it is interesting to consider the multi-period and multi product model.

6. References

[1] Barahona, F., Jensen, D.. Plant Location with Minimum Inventory. *Mathematical Programming*. 1998, **83**: 101-111.

[2] Bramel, J., Simchi-Levi, D.. The Logic of Logistics. Springer-Verlag, New York, 2000.

[3] Daskin, M.S.. Network and Discrete Location: Models Algorithms and Applications. Wiley-Interscience, New York, 1995.

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

[5] Erlebacher, S.J., Meller, R.D.. The interaction of location and inventory in designing distribution systems. *IIE Transactio.* 2000, **32**: 155-166.

[6] Hamacher, H., Drezner, Z.. Facility Location: Applications and Theory. Springer-Verlag, Berlin, 2002.

[7] Jayaraman, V.. Transportation, facility location and inventory issues in distribution network design. *International Journal of Physical Distribution and Logistics Management*. 1998, **18** (5), 471-494.

[8] Kirkpatrick, S., Gelatt, C. & Vecchi, M.. Optimization by simulated annealing. *Science*. 1983, **220**: 671-680.

[9] Miranda, P.A., Garrido, R.A.. Incorporating inventory control decisions into a strategic distribution network

design model with stochastic demand. *Transportation Research, Part E*. 2004, **40** (3): 183-207.

[10] Nozick, L.K., Turnquist, M.. Integrating inventory impacts into a fixed-charge model for locating distribution centers. *Transportation Research, Part E, Logistics and Transportations Review*. 1998, **34** (3), 173-186.

[11] Ozen, L.. Location-inventory planning models: Capacity issues and solution algorithms. Dissertation for the Degree of Doctor of Philosophy, Northwestern University, 2004.

[12] Shen, Z.J.. Efficient algorithms for various supply chain problems. Ph.D. Dissertation, Northwestern University, Evanston, IL, 2000.

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

[14] Shen, Z.J., Coullard, C., Daskin, M.. A joint location-inventory model. *Transportation Science*. 2003, **37**: 40-55.

[15] Shu, J., Teo, C.P., Shen, Z.J.. Stochastic transportation-inventory network design problem. *Operations Research.* 2005, **53**: 48-60.

[16] Simchi-Levi, D.. Hierarchical design for probabilistic distribution systems in euclidean spaces. *Management Science*. 1992, **38**: 198-211.

[17] Skiscim, C.C., Golden, B.L.. Optimization by simulated annealing: a preliminary computational study for the TSP. Winter Simulation Conference, 1983.

[18] Teo, C.-P., Ou, J., Goh, M.. Impact on Inventory Costs with Consolidation of Distribution Centers. *IIE Transactions*. 2001, **33**: 99-110.

[19] Vidal, C.J., Goetschalckx, M.. A global supply chain model with transfer pricing and transportation cost allocation. *European Journal of Operational Research*. 2001, **129**: 134-158.