Middle-East Journal of Scientific Research 10 (3): 374-388, 2011
ISSN 1990-9233

Solving Combined Model Inventory Control with Queuing Theory Approach Using Meta-Heuristic Algorithms

Asghar Hemmati, Saeed Bahrami, Mohammad Alaghebandha and Mohammad Ahmadifard

Department of Indstry, Abhar Branch, Islamic Azad University, Zanjan, Iran

Abstract: Nowadays, there is a lots of minimizing methods for preserving & goods ordering which is Complicated or offering wrong trends in obtaining optimized answer. Recently, Stock chilling & blend with array theory have been noticed to overcome the deficiency. In this paper, considered compound model consist of periodic review of stock checking & multitude input of array theory in multiple product having wane ho use space, permitted deficit number, minimum servicing & expected cost deduction in defaulted state, missed selling. This problem is a sort of NP-hard problems & is so hard to achieve a non-liner integer programming, so we have used three ultra innovation algorithm of genetic, refrigerating imperialist competition simulation in this paper. In order to probe the efficiency of the algorithms, a negative percentage of index criterion in a set of small, medium & large problems which presents that imperialism competition algorithm have had an acceptable function.

Key words: Multiple stock checking system · Multi fade Maputo of array theory · Periodical review · Genetic algorithm · Refrigeration & imperialism competition

INTRODUCTION

Inventories control and maintenance that are in form of physical products is the general problem of all companies in each section of economy. When supply and demand are not in the same size and nonuniform; inventory is produced. The objective of inventory management is therefore to balance conflicting goals like keeping stock levels down to have cash available for other purposes and having high stock levels for the continuity of the production and for providing a high service level to customers [1]. Stochastic modelling is the application of probability theory to the description and analysis of real world phenomena. One of the most important domains in stochastic modelling is the field of queueing theory. Many real systems can be reduced to components which can be modelled by the concept of a so–called queue. A queue in the more exact scientific sense consists of a system into which there comes a stream of users who demand some capacity of the system over a certain time interval before they leave the system again. Thus a queueing system can be described by a (stochastic) specification of the arrival stream and of the system demand for every user as well as a definition of the service mechanism [2]. Product suppliers have definite abilities, so they have to make contraptions in order to decrease waiting time of their customers. For making optimum decision these suppliers have to use queueing theory so that they not only specify required level of resources for investment but also satisfy their customers as much as possible. The quality of connection between queueing theory and inventory control systems and combination usage of them is the topic that is considered by several researchers in recent years. In this paper we also try to find a way for using queueing theory in inventory control by means of proposed system.

In recent years, researchers have expanded the inventory models to increase their applications. For instance, Mohsen ElHafsi. [3] studied a pure assemble-to-order system subject to multiple demand classes where customer orders arrive according to a compound Poisson process. He showed that the optimal production policy of each component is a state-dependent base-stock policy and the optimal inventory allocation policy is a multi-level state-dependent rationing policy. Because both order inter-arrival times and production times are exponentially distributed, the system is memoryless and decision epochs can be restricted to only times when the state changes. Xiaoming Liu *et al*. [4] in their paper considered

Coresspending Author: Asghar Hemmati, Department of Indstry, Abhar Branch, Islamic Azad University, Zanjan, Iran.

the cost-effective inventory control of work-in-process (WIP) and finished products in a two-stage distributed manufacturing system. This paper first uses a network of inventory-queue model to evaluate the inventory cost and service level achievable for given inventory control policy and then derives a very simple algorithm to find the optimal inventory control policy that minimizes the overall inventory holding cost and satisfies the given service level requirements. Taleizadeh *et al*. [5] investigated a stochastic replenishment multi-product inventory model and proposed two models for two cases of uniform and exponential distribution of the time between two replenishments. They showed that the models were integer-nonlinear programming problems. Attahiru S. Alfa *et al*. [6] studied the discrete time $GI^X/G^Y/1$ queueing system. First, some general results are obtained for the stability condition, stationary distributions of the queue lengths and waiting times. In addition, they studied a GI /M/1 type Markov chain associated with the age process of the customers in service. R.M. Hill [7] considered continuous-review lost-sales inventory models with no fixed order cost and a Poisson demand process. There is a holding cost per unit per unit time and a lost sales cost per unit. The objective is to minimise the long run total cost and explore alternative approaches which might offer better solutions. G.P. Kiesmuller *et al*. [8] considered a single node in a supply chain that faces stochastic demand. In this paper, they investigate this waiting time in an (R,s,Q) inventory system under compound renewal demand. They provide an approximation for the distribution function of the customer waiting time and they determine the minimal reorder level subject to a maximum average waiting time.

A.K. Maiti *et al*. [9] solved mixed-integer non-linear programming problem with constraints by a realcoded genetic algorithm (RCGA). This GA is based on Roulette wheel selection, whole arithmetic crossover and non-uniform mutation. Also, demand rate is a linear function of selling price, time and non-linearly on the frequency of advertisement. The model is formulated with infinite replenishment and shortages are not allowed. The mathematical model becomes a constrained non-linear mixed-integer problem. Their aim is to determine the optimal shipments, lot size of the two warehouses, shipment size and maximum profit by maximizing the profit function. Stanislaw Bylka [10] presented a periodic review capacitated lot sizing model with limited backlogging and a possibility of emergency orders. The main intention of placing emergency orders is to satisfy the demand as soon as a shortage occurs. It is assumed that the demands are independently distributed in successive periods. The measure of effectiveness is the total (or average per period) expected cost, which includes holding cost, shortage cost and two types of order costs. The minimum cost is obtained by considering this system as a discrete-time Markovdecision process. Eungab Kim [11] treated an inventory control problem in a facility that provided a single type of service for customers. Items used in service are supplied by an outside supplier. To incorporate lost sales due to service delay into the inventory control, they model a queueing system with finite waiting room and non-instantaneous replenishment process and examine the impact of finite buffer on replenishment policies. Employing a Markov decision process theory, they characterize the optimal replenishment policy as a monotonic threshold function of reorder point under the discounted cost criterion.

Ming Dong *et al*. [12] developed a network of inventory-queue models for the performance modeling and analysis of an integrated logistic network. This paper extended the previous work done on the supply network model with base-stock control and service requirements. Instead of one-for-one base stock policy, batch-ordering policy and lot-sizing problems are considered. In practice, the assumption of uncapacitated production is often not true; therefore, $GI^x/G1$ queueing analysis is used to replace the $M^x/G/\infty$ queue based method. To include lot-sizing issue in the analysis of stores, a fixed-batch target-level production authorization mechanism is employed to explicitly obtain performance measures of the logistic chain queueing model. Chingping Han *et al*. [13] constructed a mathematical model to describe the stochastic multiple-period two-echelon inventory with the many-to-many demand supplier network problem. Genetic algorithm was applied to derive optimal solutions through a two-stage optimization process. The goal is to determine the target inventories and the allocation quantities from warehouse i to market j in order to minimize the expected inventory and distribution costs in a finite planning horizon. Arslan, H *et al*. [14] proved the optimal inventory policy structure for both continuous and discrete-time M/G/1 and G/M/1 models with an alternate source of goods and make-to-order production. They also provide an expression from which inventory costs can be calculated for an M/M/1 model, although no closed-form expression for the optimal policies is possible. Geremie Gallien *et al*. [15] examined the component procurement problem in a single-item, make-to-stock assembly system. The suppliers are uncapacitated and have independent but non-identically distributed stochastic delivery lead times. Assembly is instantaneous, product demand follows a Poisson process and unsatisfied demand is backordered. The objective is to minimize the sum of steady state holding and backorder costs over a pre-specified class of replenishment policies. Combining existing results from queueing theory with original results concerning distributions that are closed under maximization and translation; they drive a simple approximate solution to the problem when lead time variances are identical. J.Koyanagi *et al*. [16] studied preventive maintenance for an M/M/1 queueing system which deals with the maintenance policy of a queueing system under periodic observation, in which the observation time is determined and the decision is made on the basis of the queue length and the age of the system.

In this article, we consider a multiple-products (R,T) model in which there are limited warehouse spaces, number of shortage, service level and cost of expected shortage. To do this first we define and model the problem in Sections 2 and 3, respectively. Then, in Section 4, we propose a genetic algorithm, Simulated Annealing and Imperialist Competitive Algorithm to solve the problem. In order to demonstrate the application of the proposed methodology, we provide a numerical example along with parameter adjust and compare algorithms in Section 5 just in backorder state. Finally, the conclusion and some recommendations for future research come in Section 6.

Problem Definition: In this paper we consider a system consisting of warehouse, retailer, external supplier and customer (Fig 1). We assume that the retailer faces a Poisson demand and unsatisfied demand will be backorder. The transportation time for an order to arrive at a retailer from the warehouse has exponential distribution (retailer's lead time). The warehouse orders to an external supplier and the lead time for an order to arrive at the warehouse is assumed to be constant. The demands that are reached to retailer caused him to order some demands to warehouse with T intervals and Qr quantity. Warehouse orders also supply from external supplier.

Because arrival of products to the retailer is in form of stockpile amount, so we have batch arrival with Q_r variable that time between groups arrival has exponential distribution. Due to existence of one retailer in the under assessment system, we have one server. Lead time is exponential random variables and we indicate it with L. We suppose that interval between two orders or continuous receipt are random that has exponential distribution. Retailer's service times for j customer are exponential random variables. Suppose services independently that are fixed for every customer, but they are different from one customer to another. It is assumed that demand arrival and service time are independent. Hence, it can be inferred that when an order is obtained it needs a service time, so if service time to previous customer don't finish, or it takes time more than arrival time of next demand, we will observe formation of queue, therefore, we have exponential queue system that both time between arrival and service time have exponential distribution and because we have one server there is $M^{qr}/M/1$ queue system. Now with regard of this subject and periodic order system as inventory order policy, we can use average number of customers in the system for a long time for calculating the average inventory in periodic order inventory system. In our model the warehouse uses continues review policy and the retailer uses periodic ordering policy. In this system, the retailer orders to the warehouse in a stochastic time interval; i.e., the ordering size and the time interval between any two consecutive orders from the retailer to the warehouse are stochasic numbers. The advantage of this policy is that the retailers' orders, which constitute warehouse demand, are stochastic. The stochastic demand for the warehouse leads to a harden inventory control and one of whose advantages is superimpose of the safety stock at the warehouse. To define the problem, consider a warehouse that works with a supplier. The situation by which the warehouse and the supplier interact with each other is defined as follows:

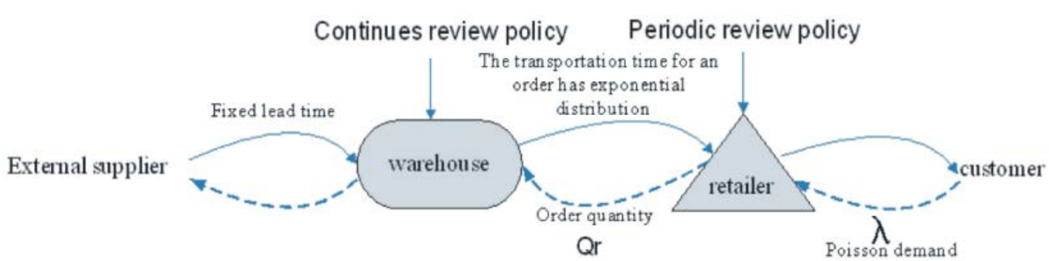

Fig. 1: A system in M^{Qr} / M/1

- The retailer faces a Poisson demand.
- The warehouse faces a stochastic demand.
- The time interval between two consecutive orders of the retailer is exponentional distribution.
- Unsatisfied demand by the retailer will be backorder or lost sale or combination of them.
- Shortage is not allowed at the warehouse.
- There is no lot-splitting at the warehouse.
- The transportation time for an order to arrive at the retailer from the warehouse (retailer's lead time) is exponential distribution.
- The warehouse orders to an external supplier with infinite capacity.
- The lead time for an order to arrive at the warehouse is constant.
- The problem is to determine the inventory position up to R, such that the total cost of inventory is minimized while the constraints are satisfied.

Problem Modeling: To develop the proposed models, we adopt the following notations and parameters in Section 3.1. Then, different costs are derived in Section 3.2. Finally, we present the model of the problem in Section 3.3.

Parameters and Notations: For j = 1,2,…,n, we define the parameters of the model as follows:

n: The number of products

h_{wj}: Holding cost rate at the warehouse for product j

A_{wj}: The fixed cost of Ordering for the warehouse for product j

T_{wj}: Time interval between any two consecutive orders of the warehouse for product j

Q_{wj}: Order quantity of the warehouse for product j

h_{rj}: Holding cost rate at the retailer for product j

A_{rj}: The fixed cost of Ordering for the retailer for product j

T_{rj}: Time interval between any two consecutive orders of the retailer for product j

φ_j: Arrival rate of the customer for product j

μ_j: Service rate of the server for product j

ρ_j: Productivity coefficient for product j

$\bar{I}_j$: Average Inventory level at the retailer between (0, T) during backorder period for product j

π_j: Fixed shortage cost for product j in backorder manner

π'_j: Fixed shortage cost for product j in lost sale manner

L_j: Length of a lead time of product j for retailer

ℓ_j: Queue length in queueing theory for product j

F: Available warehouse space for retailer for all products

f_j: Space occupied by each unit of product j

G: Number of allowed shortage

P_j: Service level for product j

S: Expected allowable shortage cost in backorder state

S': Expected allowable shortage cost in lost sale state

Γ_j: Maximum inventory at the warehouse for product j

SS_j: Safety stock for product j

Q_{rj}: Stockpile amount random variable in batch arrival system for product j (Order quantity of the retailer for product j)

$E[Q_{rj}]$: Average stockpile amount of product j

$(y_1)_j$: Demand in T time for product j that has Poisson with λ_{1j} parameter ($y_1 \sim pp(\lambda_1)$)

$(y_2)_j$: Demand in L time for product j that has Poisson with λ_{2j} parameter ($y_2 \sim pp(\lambda_2)$)

$y_j=(y_1+y_2)$: Demand in L+T time for product j that has Poisson with $\lambda_j = \lambda_{1j} + \lambda_{2j}$ parameter

R_j: The maximum inventory position after order for product j

$P(y_j)$: Demand probability density function

$\bar{b}(R_j)$: Average shortage for product j

Ch_r: Expected holding cost per time unit at the retailer in the steady state.

CB_r: Expected shortage (backorder) cost per time unit at the retailer in the steady state.

TC_r: Expected total cost per time unit at the retailer in the steady state.

TC_w: Expected total cost per time unit at the warehouse in the steady state.

TC_B: Expected total system (retailer and warehouse) cost per time unit in the steady state.

$K_1(R,T)$: Expected total cost per time unit at the retailer in the steady state in (R, T) system in backorder state

$K_2(R,T)$: Expected total cost per time unit at the retailer in the steady state in (R, T) system in lost sale state

K(R,T): Expected total cost per time unit at the retailer in the steady state in (R, T) system in combination state.

So in (R, T) system we have: demand in interval L+T have a Poisson distribution with $\lambda_1+\lambda_2$ parameter like statements (1) and (2).

$$D_{r_{L+T}} \sim pp(\lambda_1 + \lambda_2) \quad (1)$$

$$L_r + T_r \sim \text{Erlang}(2,\lambda) \quad (2)$$

Costs Calculations: In this section, we develop two type of cost for present problem that involve retailer and warehouse cost in subsections 3.2.1, 3.2.2, 3.2.3 and 3.2.4, respectively.

Formulation of the Retailer Cost in Backorder Manner: The expected total cost per time unit at the retailer which contains the ordering, holding and shortage costs in (R, T) system is as follow [17]:

$$K_1(R,T) = \frac{1}{T_r}A + h\bar{I} + \frac{\pi}{T_r}\bar{b}(R) \quad (3)$$

Because T has exponential distribution in under assessment system; we use $\frac{1}{E[T_r]}$ instead of $\frac{1}{T_r}$ in calculating ordering cost and CB_r. We operate as follow to achieve $\bar{b}(R_j)$ [17]:

$$\bar{b}(R_j) = \sum_{y_j=R_j}^{\infty} P(y_j).(y_j - R_j) \quad (4)$$

Because orders arrive in form of same size packs and they are discrete and, moreover, service method is in single form; it means supply of demands takes place in single form, so we calculate $\bar{b}(R_j)$ discretely. On the other hand we have discrete state. We define $P(y_j)$ like Eq. (5):

$$P(y_j) = P(N(L+T)=n) = \int_{t=0}^{\infty} P(N(L+T)=n \mid T=t)\, f_T(t)d_t$$

$$= \int_{t=0}^{\infty} \frac{P(N(L+T)=n \cap T=t)}{P(T=t)} f_T(t)d_t = \lambda e^{-\lambda n}(n+\frac{1}{\lambda}) \quad (5)$$

Now, we can calculate $\bar{b}(R_j)$ as follow:

$$\bar{b}(R_j) = \sum_{y_j=R_j}^{\infty} \lambda_j e^{-\lambda_j y_j}(y_j + \frac{1}{\lambda_j})(y_j - R_j) \quad (6)$$

With regard to this fact that average number of customers in system for a long time in M^{qr} / M/1 system is [18]:

$$\ell_j = \frac{\rho_j}{1-\rho_j} + \frac{\rho_j(\frac{E[Q_{rj}^2]-1}{E[Q_{rj}]})}{2(1-\rho_j)} \quad (7)$$

And with knowing $\rho_j = \frac{\varphi_j E[Q_{rj}]}{\mu_j}$ (refer to [18]) and because system has to be in steady state it must be $\varphi_j E[Q_{rj}] < \mu_j$; thus, we can use $\bar{I}_j$ instead of ℓ_j in (R, T) system when T has exponential distribution. It means we can write $\bar{I}_j$ as follow:

$$\bar{I}_j = \frac{\rho_j}{1-\rho_j} + \frac{\rho_j(\frac{E[Q_{rj}^2]-1}{E[Q_{rj}]})}{2(1-\rho_j)} \quad (8)$$

Now we have to calculate $E[Q_{rj}]$. With regard of this fact that P(Y=y) has Poisson distribution per T time that means number of demands per T time have y amount have Poisson distribution is probable, for calculating of $E[Q_r]$ we can operate as follow:

For instance, at the end of a period with T time inventory quantity is 0, it means amount of demand has had P (D≥R), therefore, it has to be ordered in R amount compatible with (R,T) system until inventory level up to pre-determined R amount. Now if inventory quantity is 1 at the end of T time, demand amount will have probability P(D≥R-1) and, therefore, it has to ordered in amount of R-1 compatible with (R, T) system until inventory level up to pre-determined R amount. With this reasoning we make table below and then calculate $E[Q_r]$ by this Table.

Inventory position at the end of the T time (Q_r)	Order quantity	P(y)
0	R	P(D≥R)
1	R-1	P(D≥R-1)
2	R-2	P(D≥R-2)
.	.	.
.	.	.
.	.	.
R	0	P(D≥0)

With use upper table we can perform underneath calculations:

$$E[Q_r] = R \times P(D \geq R) + (R-1) \times P(D \geq R-1) + (R-2) \times P(D \geq R-2) + ... + ... + 2 \times P(D \geq 2) + 1 \times P(D \geq 1) + 0 \times P(D \geq 0) \quad (9)$$

$$P(D \geq R-i) = \sum_{Q_r=R-i}^{\infty} \frac{e^{-\lambda_1}\lambda_1^{Q_r}}{Q_r!}$$

Now, with upper information we obtain $E[Q_{rj}]$ and $E[Q^2_{rj}]$ as follow:

$$E[Q_{rj}] = \sum_{i=0}^{R_j} \sum_{Q_{r_j}=R_j-i}^{\infty} (R_j - i) \times \frac{e^{-\lambda_{1j}} \lambda_{1j}^{Q_{rj}}}{Q_{rj}!} \quad (10)$$

$$E[Q^2_{rj}] = \sum_{i=0}^{R_j} \sum_{Q_{r_j}=R_j-i}^{\infty} (R_j - i)^2 \times \frac{e^{-\lambda_{1j}} \lambda_{1j}^{Q_{rj}}}{Q_{rj}!} \quad (11)$$

It has to be reminded here because (R-i) indicate arrived demand amount into the system, infact, this amount is considered as arrived stockpile (Q_r) in $M^{Qr}/M/1$ queueing system. In mentioned system with queueing theory expected total system cost in steady state is calculated. Total cost of system is included: holding cost, order and shortage cost for retailer and holding cost and order cost for warehouse. The objective is determining optimum R_j until we could minimize the total cost of system. In mentioned system with queueing theory we derive the expected total system cost in the steady state. The total system cost contains the holding, ordering and shortage costs at the retailer and the holding and ordering costs at the warehouse. Now, retailer cost of (R,T) model in multi product manner, can be define as follow:

$$K_1(R,T) = \sum_{j=1}^{n} \left(\frac{1}{E\left[T_j\right]} A_j + h_j \bar{I}_j + \frac{\pi_j}{E\left[T_j\right]} \bar{b}(R_j)\right) \quad (12)$$

Formulation of the Retailer Cost in Lost Sale Manner: Now, if we want to use Eq. (12) for lost sale state, we must change some parameter and Eq. as below. First, we change Eq. (8) to (13) as follow:

$$\bar{I}'_j = \bar{I}_j + \bar{b}(R_j) = \frac{\rho_j}{1-\rho_j} + \frac{\rho_j\left(\frac{E[Q^2_{rj}] - 1}{E[Q_{rj}]}\right)}{2(1-\rho_j)} + \bar{b}(R_j) \quad (13)$$

Then use π'_j instead of π_j for obtain Eq. (14) in lost sale state as follow:

$$K_2(R,T) = \sum_{j=1}^{n} \left(\frac{1}{E\left[T_j\right]} A_j + h_j \bar{I}'_j + \frac{\pi'_j}{E\left[T_j\right]} \bar{b}(R_j)\right) \quad (14)$$

Formulation of the Retailer Cost in Combination Manner: For having combination model we construct Eq. (15) that involves alpha coefficient for backorder manner:

$$K(R,T) = \alpha.K_1(R,T)+(1-\alpha).\ K_2(R,T) \quad (15)$$

Formulation of the Warehouse Cost: The retailer's orders which constitute warehouse demands, with Q_r quantity in T times entered to the warehouse. The retailers' orders are stochastic. The stochastic demand for the warehouse leads to superimpose of the safety stock at the warehouse. It is assumed that there is no lot-splitting at the warehouse and shortage is not allowed at the warehouse so that the order quantity of the warehouse is an integer multiple (m_j) of the order quantity of the retailer. For the optimal solution the arrival of an order to the warehouse must correspond to the delivery of an order to the retailer. Thus, the maximum inventory level at the warehouse is $Q_{wj} - Q_{rj}$ (or$(m_j-1)Q_{rj}$) and $T_{wj} = m_j\, E[T_{rj}]$. The holding cost per time unit at the warehouse which contains the safety stock can be obtained with uses service level. We know that shortage is not allowed at the warehouse, thus we have:

$$SS_j = \Gamma_j - \mu_{j}\ {}_{L+T} \quad (16)$$

$$P(y \le \Gamma) = 1 \Rightarrow \int_0^{\Gamma} P(y)dy = 1 \quad (17)$$

Now based on Eq. (5) the statement (17) is obtained by Eq. (18):

$$\int_0^{\Gamma} P(y)dy = -\,ye^{-\lambda y} + 2[-\frac{1}{\lambda}e^{-\lambda y}]_0^{\Gamma} \quad (18)$$

We can write (18) like Eq. (19):

$$-ye^{-\lambda y} + 2[-\frac{1}{\lambda}e^{-\lambda\Gamma} + \frac{1}{\lambda}] = 1 \quad (19)$$

With having (19) we can obtain safety stock:

$$SS_j = \frac{-Ln\frac{\lambda_j}{2}}{\lambda_j} - \lambda_j \quad (20)$$

The expected total cost per time unit at the warehouse in the steady state is the sum of the ordering and holding costs which is formulated as follows:

$$TC_{wj} = \frac{A_{wj}}{T_{wj}} + h_{wj}\left(\frac{-Ln\frac{\lambda_j}{2}}{\lambda_j} - \lambda_j + \frac{(Q_{wj} - Q_{rj})}{2}\right)$$
$$\begin{aligned} &\text{s.t}: \\ &Q_{wj} = m_j Q_{rj} \\ &T_{wj} = m_j T_{rj} \\ &m_j: \text{Integer} \end{aligned} \tag{21}$$

Substituting $Q_{wj} = m_j Q_{rj}$ and $T_{wj} = m_j\, E[T_{rj}]$ in (21) we have:

$$TC_{wj} = \frac{A_{wj}}{m_j E[T_{rj}]} + h_{wj}\left(\frac{-Ln\frac{\lambda_j}{2}}{\lambda_j} - \lambda_j + \frac{(m_j - 1)Q_{rj}}{2}\right) \qquad m_j: \text{Integer} \tag{22}$$

Problem Modeling: As we mentioned earlier, the goal is to determine the inventory position up to R, such that the total cost of inventory is minimized while the constraints are satisfied. The constraints are:

- The space of the warehouse
- The number of shortage
- minimum of service level
- Expected shortage cost

Space of the Warehouse Constraint: With having F and f_j for retailer in (R, T) model we consider this constraint in the model as follow:

$$\sum_{j=1}^{n} f_j R_j^* \le F \tag{23}$$

Number of Shortage Constraint: For calculate this constraint, we needed goal of the allowable number of shortage allowable (G) and Eq. (6). Thus, we have:

$$\sum_{j=1}^{n} \bar{b}(R_j) \le G \tag{24}$$

Minimum of Service Level Constraint: Service level indicates the ability to meet customer demand by available inventory that is important operation index and key factor in measuring reliability in supply chain. Service level 100% cause to increase holding cost of inventory and low service level might lead to lost sell. Hence, inventory control has a great impression on service level of costumers. To obtain minimum of service level constraint, first the demand amount per L+T times has to be obtained from (Eq. 5):

$$\lambda_j e^{-\lambda_j y_j}\left(y_j + \frac{1}{\lambda_j}\right)$$

And then with consideration of a primary objective amount and usage of $P(y_{j(L+T)} \le R_j) \ge P_j$ total relationship of minimum service level we calculate the amount of this constraint as follow:

$$\sum_{y_j=0}^{R_j} \lambda_j e^{-\lambda_j y_j}\left(y_j + \frac{1}{\lambda_j}\right) \ge P_j \tag{25}$$

Expected Shortage Cost Constraint: with knowing S and π_j and π'_j and S' we can obtain this constraint for backorder state (ie. Eq. (26)) and lost sale state (ie. Eq. (27)) as follow:

$$\sum_{j=1}^{n}\left(\pi_j \times \sum_{y_j=R_j}^{\infty} \lambda_j e^{-\lambda_j y_j}\left(y_j + \frac{1}{\lambda_j}\right)(y_j - R_j)\right) \le S \tag{26}$$

$$\sum_{j=1}^{n}\left(\pi'_j \times \sum_{y_j=R_j}^{\infty} \lambda_j e^{-\lambda_j y_j}\left(y_j + \frac{1}{\lambda_j}\right)(y_j - R_j)\right) \le S' \tag{27}$$

Minimum of waiting time for customer (product) constraint (just in combination manner): For calculate this constraint (ie. Eq 29), we needed goal of the allowable waiting time for each customer (product j) (u_j) and Eq.s (6) and (28). Thus, we have:

$$E[W_j] \le u_j \tag{28}$$

$$E[W_j] = \frac{\bar{b}(R_j)}{\lambda_j} \tag{29}$$

The total system cost per time unit in backorder state is the sum of the total cost per time unit at the retailer and the total cost per time unit at the warehouse. So, the expected total system cost per time unit in the steady state is:

$$TC_B = \sum_{j=1}^{n}\left(\frac{1}{E\left[T_j\right]}A_j + h_j\bar{I}_j + \frac{\pi_j}{E\left[T_j\right]}\bar{b}(R_j)\right)$$
$$+\sum_{j=1}^{n}\frac{A_{wj}}{m_j E\left[T_j\right]} + h_{wj}\left(-\frac{1}{\lambda_j}\ln\left(\frac{\lambda_j - 2}{-\Gamma_j\lambda_j - 2}\right) - \lambda_j + \frac{(m_j - 1)Q_{rj}}{2}\right)$$

s.t :

$$\sum_{j=1}^{n} f_j R_j^* \le F$$
$$\sum_{j=1}^{n} \bar{b}(R_j) \le G$$
$$\sum_{y_j=0}^{R_j} \lambda_j e^{-\lambda_j y_j}\left(y_j + \frac{1}{\lambda_j}\right) \ge P_j$$
$$\sum_{j=1}^{n}\left(\pi_j \times \sum_{y_j=R_j}^{\infty} \lambda_j e^{-\lambda_j y_j}\left(y_j + \frac{1}{\lambda_j}\right)(y_j - R_j)\right) \le S$$
$$\lambda_j Q_{jr} < T_{jr}$$
$$R_j \ge 0$$
$$Q_{rj},\ m_j : \text{is a positive integer} \qquad (30)$$

Solution Method: The presented model of linear integral number. For solving suggested model because it is hard-NP, used from three algorithm; extra innovative, genetic algorithm, refrigeration algorithm and imperialistic competition algorithm.

Genetic algorithm (GA'): GAs are stochastic search algorithms based on the mechanism of natural selection and natural genetics. GA, differing from conventional search techniques, start with an initial set of random solutions called population satisfying boundary and/or system constraints to the problem. Each individual in the population is called a chromosome, representing a solution to the problem at hand. Chromosome is a string of symbols usually, but not necessarily, a binary bit string. The chromosomes evolve through successive iterations called generations. During each generation, the chromosomes are evaluated, using some measures of fitness. To create the next generation, new chromosomes, called offspring, are formed by either merging two chromosomes from current generation using a crossover operator or modifying a chromosome using a mutation operator. A new generation is formed by selection, according to the fitness values, some of the parents and offspring and rejecting others so as to keep the population size constant. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimum or suboptimal solution to the problem. In most cases, GAs can find the global optimum solution with a high probability [19]. In the next subsections, we demonstrate the steps required to solve the model given in Eq. (30) by a genetic algorithm.

Initial Conditions: The required initial information to start the GA is:

- *Population size*: It is the number of the chromosomes that we will keep in each generation and we denote it by *pop_size*.
- *Crossover probability*: Is defined as the probability of the number of offspring produced in each generation to the population size, denoted by *Pc*.
- *Mutation probability*: Is defined as the percentage of the total number of genes in the population that controls the probability with which new genes are introduced into the population for trial denoted by *Pm*.

Chromosome: In the GA method, we present a chromosome by a matrix that has one row and n columns. Each column shows the number of product. In addition, the row of the matrix show the R. Fig 2 presents the general form of a chromosome.

Evaluation: In a GA method, as soon as a chromosome is generated, we need to assign a fitness value for it. In optimization problems, it is the value of the objective function. Since there are 4 constraints in the model of the problem given in Eq. (30), some generated chromosomes may not be feasible. In order to control infeasible solutions, we may employ the penalty policy. In the proposed algorithm, the penalty is defined as a positive and known coefficient of violation of constraints. When the coefficient is selected larger, the penalty is larger too. Furthermore, when a chromosome is feasible, its penalty is zero. Then, the fitness function for a chromosome is defined as the sum of its objective function and penalty.

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & & n \end{array} \\ R_i & \left[\begin{array}{ccccc} R_1 & R_2 & R_3 & \ldots & R_n \end{array}\right] \end{array}$$

Fig. 2: Choromosome presentation

Initial Population: In this stage, a collection of chromosomes is randomly generated. We note that the row is generated within its limits presented in the constraints of the model Eq.(30). To generate the random solutions for the initial population, the procedure is as follow:

Generate randomly R values within interval [Max(R_j), Min(R_j)]. If constraints are violated, then, act according to subsection 4.1.3 for penalty policy.

Crossover: Crossover is the main genetic operator. It operates on two chromosomes at a time and generates offspring by combining both chromosomes' features. The crossover probability denoted by (*Pc*) is defined as the probability of the number of offspring produced in each generation to the population size. A higher crossover probability allows exploration of more of the solution space and reduces the chances of settling for a false optimum; but if this probability is too high, it results in the wastage of a lot of computation time in exploring unpromising regions of the solution space. It is important to maintain the feasibility of the newly generated offspring for the problem at hand. Thus, we use the arithmetic crossover (AC) [22] operator to explore the solution space and maintaining the feasibility of the newly generated offspring simultaneously. The AC produces a new offspring as complimentary linear combination of the parents as follow: offspring ≡ Pc × Parent + (1–Pc) × Parent2. Since, R_j are integers values, the integer part of R_j offspring i.e., [R (offspring)] must be considered as a true value.

Mutation: Mutation is a background operator which produces spontaneous random changes in various chromosomes. A simple way to achieve mutation would be to alter one or more genes. In GA, mutation serves the crucial role of either (a) replacing the genes lost from the population during the selection process so that they can be tried in a new context or (b) providing the genes that were not present in the initial population. The mutation probability (denoted by *Pm*) is defined as the percentage of the total number of genes in the population. The mutation probability controls the probability with which new genes are introduced into the population for trial. The main task of the mutation operator is to maintain the diversity of the population in the successive generations and to exploit the solution space. In this paper, a mutation operator, called Stepping Stone mutation (SSM) [22], is used that is inspired by the Stepping Stone method for solving the classical transportation problem. The SSM guarantees the generated offspring will remain feasible if its parent is feasible. The procedure of the SSM is as follows:

- Select randomly a solution from current population.
- Calculate values , $R_q = \min_{j=1}^{n} R_j$ $R_p = \max_{j=1}^{n} R_j$ and σ $= R_p - R_q$
- Change values R_p and R_p to $R_p \rightarrow R_p - \left\lfloor \frac{\sigma}{2} \right\rfloor$ and $R_q \rightarrow R_q + \left\{ \sigma - \left\lfloor \frac{\sigma}{2} \right\rfloor \right\}$, recpectively.

Chromosomes Selection: In the next phase of the genetic algorithm, we select the chromosomes for the next generation. This selection is based on the fitness value of each chromosome. We select N chromosomes among the parents and offsprings with the best fitness values.

Stopping Criteria: The last step in the methodology is to check if the method has found a solution that is good enough to meet the user's expectations. Stopping criteria is a set of conditions such that when the method satisfies them, a good solution is obtained. In this research, we stop when we reach to maximum generation.

Simulated Annealing: Simulated annealing (SA) is so named because of its analogy to the process of physical annealing with solids, in which a crystalline solid is heated and then allowed to cool very slowly until it achieves its most regular possible crystal lattice configuration (i.e., its minimum lattice energy state) and thus is free of crystal defects. If the cooling schedule is sufficiently slow, the final configuration results in a solid with such superior structural integrity. Simulated annealing establishes the connection between this type of thermodynamic behavior and the search for global minima for a discrete optimization problem. Furthermore, it provides an algorithmic means for exploiting such a connection. At each iteration of a simulated annealing algorithm applied to a discrete optimization problem, the objective function generates values for two solutions (the current solution and a newly selected solution) are compared. Improving solutions are always accepted, while a fraction of non-improving (inferior) solutions are accepted in the hope of escaping local optima in search of global optima. The probability of accepting non-improving solutions depends on a temperature parameter, which is typically non-increasing with each iteration of the algorithm [20]. General framework of SA for our problem (Eq. 30) is as below:

- Initial *T*
- Initial *Solve*
- Evaluate *Solve*
- For *Iteration* Begin 1 To *Max-Iteration*
- Update *Solve*
- Evaluate *Solve*
- Compare *New-Eval* with *Pre-Eval*
- If *New-Eval* is best then
- accept *New-Eval*
- Else
  accept *New-Eval* with exp((*New-Eval* – *Pre-Eval*) / *T*)
- Update *T*

In line 1, an initial temperature for T is taken into account. Then an initial solution for problem is regarded. This initial solution is in form of an array with size of num×2. num is an indicator of product number. Every row of this array represents one product has a column that allocates to R variable. In third line this initial solution is assessed. If this initial solution is valid, the obtained cost will be saved. Fourth line is indicator of main loop of program. In each repetition, first, a random value is produced and then this value is assessed. If new cost is better than the cost that is obtained until now, new solution will be saved and if it is not this solution will save with *exp((New-Eval – Pre-Eval) / T)* probability. In line 4.4 temperatures is updated. In this program maximum repetition, probability to accept wrong solution and temperature updating can be varied.

Imperialistic Competition of the Problem: Algorithm of colonized compares is like other is refitted completing method start by then number of primary population.In this algorithm, every one of population, called one country and divided into two parts the in aerialist and colony every colonizer depend on his power compare some countries and contrail and control them absorption policy and colonizer competition, is the main point of this algorithm. During this competition, waked rules demonstrate and the stronger one get colonies. The production process of primary countries The countries like chromosome in genetic defined as [pi: p1, p2,…., pn] that by using an accidental amount in span [Max(P_j),Min(P_j)]. Then counting costs in every country does in consider targeting function in delay mood, limitation of question and position of each country.

Imperialists Assignment: We recognize the smallest amounts of target functions from cost matrix and call it imperialist then normalize their costs by using $C_n = c_n - \max_i\{c_i\}$ and the power of cost in every imperialist getting by formula $P_n = \left| C_n / \sum_{i=1}^{N_{mp}} C_i \right|$. Then the numbers of colonies in each imperialist will obtain on the basis of his normalized power as $Nc_i = round\ \{P_i \times N_{col}\}$ and we specified the imperialist to him. The point is here:

$$\sum_{i=1}^{N_{mp}} P_i = 1$$

$$\sum_{i=1}^{N_{mp}} N.C_i = \sum_{i=1}^{N_{mp}} \text{round}\{P_i.N_{col}\} = N_{col}$$

For example, suppose that the number of countries N_{pop} = 15, the numbers of colonies are N_{col} = 10 and the numbers of imperials are N_{imp} = 15 =S then:

$$c_1 = 8\ c_2 = 9\ c_3 = 4\ c_4 = 1\ c_5 = 6$$

$$C_1 = 8-9 = 1\ ;\ C_2 = 9-9 = 0;\ C_3 = -5;\ C_4 = -8$$

$$C_5 = -3 \Rightarrow \sum_{i=1}^{N_{mp}} C_i = \sum_{i=1}^{5} C_i = -1-0-5-8-3 = -17$$

$$P_1 = \left|\frac{-1}{-17}\right| = \frac{1}{17};\ P_2 = 0;\ P_3 = \frac{5}{17};\ P_4 = \frac{8}{17};P_5 = \frac{3}{17} \Rightarrow \sum_{i=1}^{5} P_i = 1$$

$$N.C_1 = \text{round}\left\{\frac{1}{17}\times 10\right\} = 0;N.C_2 = 0;N.C_3 = 3;N.C_4 = 5$$

$$N.C_5 = 2 \Rightarrow \sum_{i=1}^{N_{imp}} N.C._i = 10 = N_{col}$$

Really, improvement circle of this algorithm start from this step one colony in size x until move direct to imperialist. In the way, $x \sim U(0, \beta\times d)$, β is a number larger than 1 and d is the space between colony and imperialist. For this reason: β>1 that colonies in both sides near to one imperialist. The moment of colonies to imperialist and get colony and can on the base of a corner. With this target it is possible to reach through this movement to a country with lower cost. By searching target function and limitations of model, if get a lower cost, then the situation of that colony and imperialist replace. For searching other points around imperialist, we plus an amount of accidental deviation in direction of movement.Picture 2. Shows a new direction. In this picture θ is as $\theta \sim U(-\gamma, \gamma)$, y is a parameter that regulate the deviation from first direction. Consider that a mounts of β and γ was desirable and they can hare every suitable distribution in more performers showed it get good result with $\gamma = \pi/4$ and $\beta = 2$, for convergence of countries in direction to the minimum one.

Whole power of one gets an effect of imperialist's country power, but the power of colonies in one rule even if it was nothing, effect on the whole power of the rule. Whole cost defined as a:

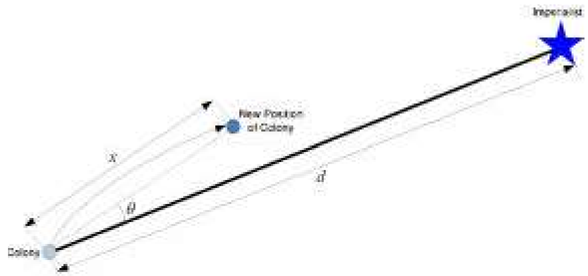

Fig. 2: Colonies movement towards their imperialist in a direction with accidental deviation The whole power of a sovereignty :

Tc_n= cost(*imperialist$_n$*) + ξ.*mean*{cost(*colonies of empire$_n$*)}

While is considered TC_n the whole cost of n^{th} of rule and ξ is a positive number less than 1. Small amount of ξ is due to the whole power of rule just define by imperialist and increasing that, will arise the rule of colonies in de fining of the whole power of a rule.

Imperialistic Competition: For starting a imperialistic competitor need to an average power of each colony of imperialist (*mean_col$_i$* = *mean_col$_i$ / NC$_i$*). Then we get the whole cost of that rule which is the compounding of imperialist costs and colonies so the imperialistic competition will start an the base of the power of each rule and the possible occupation of each rule (such as roulette cycle in genetic algorithm) for possessing colonies of faint rules. The possibility of each rule obtains on the bases of its whole power. The whole normalized cost obtain as $\mathrm{N.T.C_n} = \mathrm{T.C_n} - \max_{\mathrm{i}}\{\mathrm{T.C_i}\}$ while $T.C_n$ is whole cost and $N.T.C_n$ is the whole normalized cost of n^{th} rule. Having whole normalized cost, the possibility of each rule possession is as follows:

$$p_{p_n} = \left| \frac{N.T.C_n}{\sum_{i=1}^{N_{imp}} N.T.C_i} \right|$$

For dividing colonies among rules on the bases of their possibility of possession, vector P is:

$$\mathrm{P} = [\mathrm{p_{p_1}}, \mathrm{p_{p_2}}, \mathrm{p_{p_3}}, ..., \mathrm{p_{p_{N_{imp}}}}]$$

And, at the and we form vector P from subtracting R and p vectors:

$$\mathrm{R} = [\mathrm{r_1}, \mathrm{r_2}, \mathrm{r_3}, ..., \mathrm{r_{N_{imp}}}] \quad \mathrm{r_1}, \mathrm{r_2}, \mathrm{r_3}, ..., \mathrm{r_{N_{imp}}} \sim \mathrm{U}(0,1)$$

Refer to D vector, we have colonies of one rule that according to D indictor are maximum, for example:

$$\mathrm{T.C._1} = 8 + 0.1 \times (0) = 8 \qquad \mathrm{T.C._2} = 9$$
$$\mathrm{T.C._3} = 5.4 \quad \mathrm{T.C._4} = 15.8 \quad \mathrm{T.C._5} = 15.5$$
$$\mathrm{N.T.C._1} = 8 - 15.8 = -7.8; \mathrm{N.T.C._2} = -6.8$$
$$\mathrm{N.T.C._3} = -10.4 ; \mathrm{N.T.C_4} = 0$$
$$\mathrm{N.T.C_5} = -0.3 \Rightarrow \sum_{\mathrm{i=1}}^{\mathrm{N_{imp}}} \mathrm{N.T.C._i} = 25.3$$

$$P_{P_1} = 0.3 \,;\, P_{P_2} = 0.26; P_{P_3} = 0.4; P_{P_4} = 0 \,;\, P_{P_5} = 0.01$$

Fourth empire was the weakest one and lose one colony.

$$P = [0.3 \;\; 0.26 \;\; 0.4 \;\; 0 \;\; 0.01]$$
$$R = [0.1 \;\; 0.4 \;\; 0.3 \;\; 0.05 \;\; 0.6]$$
$$D = [0.2 \;\; -0.14 \;\; 0.1 \;\; -0.05 \;\; -0.59]$$

As a result the first empire possesses the colony.

Stopping Criteria: We continue the above process until there is just one imperialist for considered problem.

The Results of Experiment: Example problems is studied in this paper conclude 4 group of problems, one production, there productions, five productions and 40 production that are in the index A.1. because these problems concluding wide spectrum of problems solve model of com pounding ordering system with array theory's approach. For clearing the subjects of accidental result, every one of problems group will do 10 times and the average of the results will be the base of comparisons. In consider to these algorithms when they were on the assuming time they get to a best answer.

In this paper suggested algorithms did with programming language (Matlab 7.0) under Microsoft windows XP system and with two pointed code 256 ram, 2.50 Hz the use fullness of extra algorithms has a relation with regulating of its parameters, so that choosing the incorrect good algorithm parameters cause to charging to a bad one.

Purposely different ways used in lateral true that most of team are impaired.

One of the new approach in this field is using from vegetating TAGOCHI parameter tech nique. For performing the TAGOCHI method, first you love to needed surfaces in consideration to algorithms kind an the type of distinctive program. Specified designs Tagochi in algorithms ICA, SA and GA is equal to $L_{16}(4^4)$, $L_{16}(4^2)$, and $L_{27}(3^6)$. Then it gets ratio of MEAN and SNR for different problems and also gets in used algorithms and in Table 2 regulated parameters too.

Appendix A: The used problems in experiments

Table A.1. Data of the test problems

Parameter	Prablem1	Prablem2	Prablem3	Prablem4
λ_{1j}	[1]	[1 2 3]	[1 2 3 2 5]	[1 4 7 2 4 8 9 5 3 6]
λ_{2j}	[4]	[4 5 6]	[4 5 6 4 7]	[4 2 4 1 7 5 8 4 6 5]
h_j	[3]	[3 4 7]	[3 4 7 4 9]	[3 5 7 2 9 5 1 8 5 10]
A_j	[20]	[20 19 10]	[20 19 10 15 22]	[20 10 32 12 30 49 19 30 46 33]
$E[T_j]$	[.2]	[0.2 0.1 0.3]	[.2.1.3.2.5]	[0.2 0.1 0.3.8.4.6.2.55.9.34]
f_j	[2]	[2 3 4]	[2 3 4 5 1]	[2 5 3 8 4 3 9 5 3 5]
π_j	[.2]	[0.2 0.3 0.1]	[.2.3.1.005.9]	[.2.4.2.7.4.6.7.9.3.553]
F	10000	10000	10000	10000
G	10	10	10	10
P_j	[0.95]	[0.95 0.92 0.9]	[.95.92.9.9.98]	[.95.9.97.98.93.92.98.95.96.94]
S	1000	1000	1000	1000
φ_j	[5]	[5 10 3]	[5 10 3 9 5]	[5 3 5 4 8 6 10 3 9 21]
μ_j	[70]	[60 110 70]	[60 110 70 50 100]	[70 40 59 82 78 29 48 77 81 94]

Table2. The parameters used in these algorithms.

parameter	Algorithms											
	GA				SA		ICA					
problem	Pc	Pm	Pop-size	Max_Gen	T	Max	Betta	Sigma	Landa	Max_scape_angle	Npop,Nimp	Max It
1	75/0	005/0	70	30	800	50	1	60/0	60/0	8	(2,20)	40
2	55/0	05/0	55	75	800	60	1	05/0	05/0	16	(3,30)	50
3	75/0	001/0	20	20	800	150	25-Jan	6/0	6/0	16	(5,90)	80
4	95/0	025/0	45	25	1200	120	1	05/0	05/0	16	(5,60)	70

Table 3: Comparision of GA,SA,ICA according to average criteria of RPI, least answer and average time of algorithm execution

Problem	1	2	3	4
Average RPI				
GA	0.00517	0.01691	0.19503	0.25338
SA	0.00613	0.01179	0.1292	0.23277
ICA	0.0041	0.01231	0.03033	0.01596
Average Min solution				
GA	100.475	334.655	547.267	641.915
SA	100.571	332.97	522.618	636.725
ICA	100.371	333.14	471.841	520.32
Average CPU Time				
GA	3.5106	9.8899	16.7627	33.6822
SA	3.6828	10.1906	15.1977	31.0448
ICA	3.888	12.302	29.094	52.478

Table 4: Pair-wise comparisons of results from RPI algorithms GA,SA and ICA against each other

Results of *P-Value*	Test	Algorithms
0.779	0.88	GA: SA
0.908	1.72	GA: ICA
0.884	1.49	SA: ICA

Means the Paired *T_* Test Is Significant, That Is *P–Value*<α

Comparision and Evaluation of Problem Solving Criteria: For comparing algorithms used from three standards, percentage of retile error, minimum obtained answer and also the calculation time of used algorithms which explain of each later on.

Measuring Criteria of Algorithm Efficiency: Now here the question is asked that how do have to do these coma risen in algorithms? One of the best and famous standard of counting efficiency in one target problems, is using the standard of retail percentage indicator (PRI) as a standard of use these algorithms performances.

$$RPI = \frac{Alg_{sol} - Min_{sol}}{Worst_{sol} - Min_{sol}} \tag{25}$$

The way to use these standards is as follows: that, after counting amount of target function for every one of compared algorithms, the best worst amount among obtained amounts specify by Min_{so}l and $Worat_{so}$l then by using this formula you get the RPI of every algorithm. In this formula Alg_{so}l is the symbol of amount of obtained compared algorithm. It is clear if the amount of RPI will be smaller is better. The result of comparing GA, SA and ICA on the bases of average standards of RPI and the average of smallest answer and also the aviary time of performing algorithm has been brought in Table 3.

Considering to table 3 we see that the ICA algorithm regarding to GA, SA is better on the bases of average standards RPI and the average of smallest answer but on the bases of average standard of performing time of algorithm doesn't have a good performance also SA regarding to GA is better an the basis of every three of them. Just damming about the be her one the amount of RPI average algorithms regarding to each other can as an exam zero H1:D<0 in front of H0:D=0 that D is average difference RPI in algorithms therefore, zero hypothesis has been returned if:

$$t = \frac{\overline{D}}{S_D / \sqrt{n}} < -t_{\alpha,n-1} \tag{26}$$

The result of exam and the amount of-p amount in trust surface 95% for average RPI and for algorithm brought in Table 4.

As you see in table 4 we see by 95% trust, average RPI, he the obtained answer by suggested algorithm ICA regarding to other algorithms is better for this problem and also SA method in front of average GA has a less retailer deviation percentage.

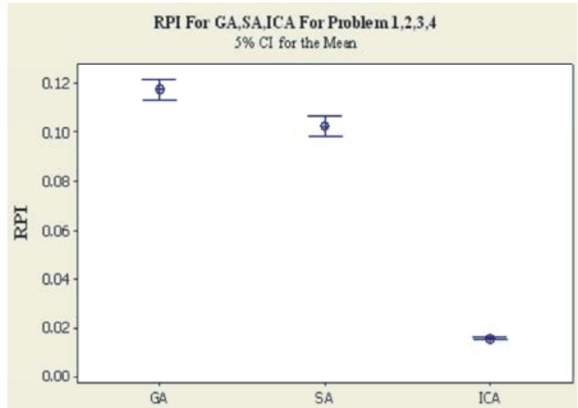

Fig. 3: The chart of average RPI for different algorithm

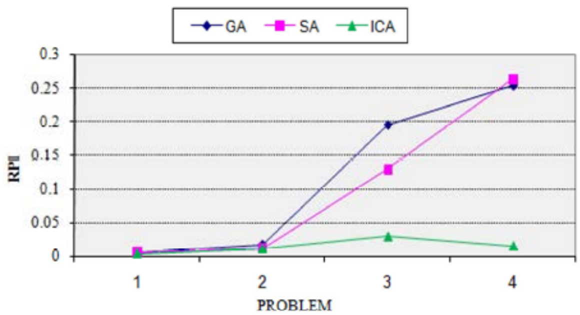

Fig. 4: The comparison of RPI algorithm ICA,GA,SA in problems 1,2,3 & 4

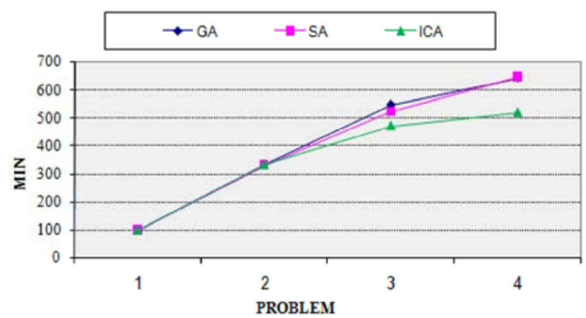

Fig. 5: The comparison of Min answer of algorithm ICA,GA,SA in problems 1,2,3, & 4

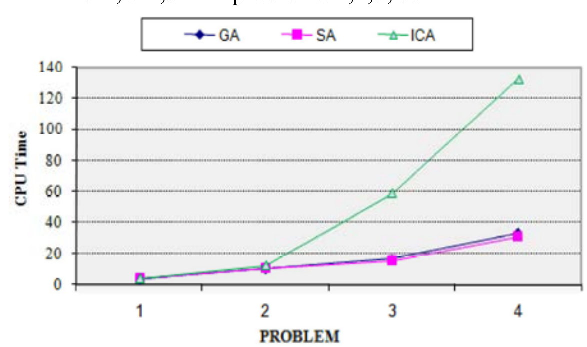

Fig. 6: The comparison of answering time of algorithm ICA, GA,SA in problems 1,2,3 & 4

Pictuer 3. Shows the average amounts chart LSD (trust space 65%) for different algorithms.

Also this chart clearly shows our claim on the basis of preference of algorithm ICT regarding to other algorithms.

As you see in picture 4, in these problem, the best way to solve is on the basis of standard of RPI algorithm in imperialistic comparison.

By Fig. 5 Clearly is recognizable that in these problems, the best method for solving on the basis of standard MIN answer is A algorithm.

And finally, in consideration to paicture6 from the standard of counting time, algorithm SA is better regarding to two other algorithms.

By all counting and resulting which did, briefly when the standard RPI and MIN answer be important, are have to use from Ida algorithm and when just the time of answer be important, we have to use SA algorithm in this paper defined one compounding model of controlling as set in periodical review and array theory group entrance in multi production mood and by the limitations of ware houses space, the allowed number of shortage, minimum serving level and shortage cost in delayed mood and demonstrated sale and also compounding mood.

The target of model is minimizing the ordering costs and keeping and shortage with getting the best amount of maximize as set about specified good.

For solving model has been used from genetic algorithm method, refrigeration simulation and imperialist comparison.

By comparing the result of these three methods and by using this software Minitab, the final results of time has been defined he standard of MIN and RPD answer is on the bas is of the imperialist compared method and when the time has been defined is on the bas is of using SA method.

Considering the shortage for ware house and consider to priority for costumers order and also considering to array system with limited waiting room and also fuzzy of entrance and exit rate can be a suitable a for next works.

CONCLUSION

The paper propounded a compounding model of stock controlling in periodical review and array theory of multitude entrance in the state of multi production considering to space limitation of warehouse, the least service and expected missing cost in the state of delay and lost sale and also the state of compound.

The object of model is ordering cost minimizing, maintaining and deficit along with the optimized extent of considered maximum stock. In order to solve the model has been used, genetic algorithm, simulation of refrigeration and imperialistic competition.

Considering to comparison of these three methods by MINITAB software, shows that the final consequence would relying on imperialistic competition when RPD and MIN is considered and on SA method when time is considered. Considering to deficit for warehouse, priority for order of customers, array system with limited waiting room and fuzziness of entrance and exit rate would be a suitable ground for coming works.

REFERENCES

1. Yasemin Arda and Jean-Claude Hennet, 2006. Inventory control in a multi-supplier system, Int. J. Production Economics, 104: 249-259.
2. Breuerb, L. and D. Baum, 2005. An Introduction to Queueing Theory and Matrix-Analytic Methods, Springer, Netherlands.
3. ElHafsi Mohsen, 2009. Optimal integrated production and inventory control of an assemble-to-order system with multiple non-unitary demand classes, European J. Operational Res., 194: 127-142.
4. Xiaoming Liu and Zhaotong Lian, 2008. Cost-effective inventory control in a value-added manufacturing system, European J. Operational Res.,
5. Taleizadeh, A., M.B. Aryanezhad and S.T.A. Niaki, 2008. Optimizing multi-product multi-constraint inventory control systems with stochastic replenishments, J. Applied Sciences 8: 1228-1234.
6. Attahiru, S. Alfa and Qi-Ming He, 2008. Algorithmic analysis of the discrete time queuing system, Performance Evaluation an International Journal.
7. Hill, R.M., 2007. Continuous-review, lost-sales inventory models with Poisson demand, a fixed lead time and no fixed order cost, European J. Operational Res., 176: 956-963.
8. Kiesmuller, G.P. and A.G. De Kok, 2006. The customer waiting time in an (R,s,Q) inventory system, Int. J. Production Economics, 104: 354-364.
9. Maiti, A.K., A.K. Bhunia and M. Maiti, 2006. An application of real-coded genetic algorithm (RCGA) for mixed integer non-linear programming in two-storage multi-item inventory model with discount policy, Applied Mathematics and Computation, 183: 903-915.

10. Stanislaw Bylka, 2005. Turnpike policies for periodic review inventory model withemergency orders, Int. J. Production Economics, 93-94: 357-373.
11. Eungab Kim, 2005. Optimal inventory replenishment policy for a queueing system with finite waiting room capacity, European J. Operational Res., 161: 256-274.
12. Ming Dong and F. Frank Chen, 2005. Performance modeling and analysis of integrated logistic chains: An analytic framework, European J. Operational Res., 162: 83-98.
13. Chingping Han and Montri Damrongwongsiri, 2005. Stochastic modeling of a two-echelon multiple sourcing supply chain system with genetic algorithm, J. Manufacturing Technology Management, 16, 1, ABI/INFORM Global, pp: 87.
14. Arslan, H., H. Ayhan and T.L. Olsen, 2001. Analytic models for when and how to expedite in make-to-order systems, IIE Transactions 33(11): 1019-1029.
15. Geremie, G. and M. Lawrence Wein, 2001. A simple and effective component procurement policy for stochastic assembly systems, Queuing Systems, 38: 221-248.
16. Koyanagi, J. and H. Kawai, 1997. An optimal maintenance policy for a queuing system server under periodic observation, International J. Reliability, Quality and Safety Engineering, 4: 357-367.
17. Bazzazi, M, N. Safaei and N. Javadian, 2008. A genetic algorithm to solve the storage space allocation problem in a container terminal, Computers & Industrial Engineering.