Volume 1, Number 2, 2012

International Journal of Research in Industrial Engineering

journal homepage: www.nvlscience.com/index.php/ijrie

Designing a Genetic Algorithm to Optimize Fulfilled Orders in Order Picking Planning Problem with Probabilistic Demand

M. Seyedrezaei[1,*], S.E. Najafi[1], A. Aghajani[2], H. Bagherzadeh Valami[3]

[1]Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[2]Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran
[3]Department of Applied Mathematics, Shahr-e-Rey Branch, Islamic Azad University, Tehran, Iran

A R T I C L E I N F O

Article history :
Received:
May 20, 2012
Revised:
August 10, 2012
Accepted:
September 25, 2012

Keywords :
Supply chain, Order picking problem, mathematical model, distribution center, fulfilled order, genetic algorithm.

A B S T R A C T

Distribution centers (DCs) play an important key role in supply chain. Delivering the right items to the right customers at the right time, at the right cost is a critical mission of the DCs. Today, customer satisfaction is an important factor for supplier companies in order to gain more profits. Optimizing the number of fulfilled orders (An order that the required quantity of all items in that order are available from the inventory and can be send to the customer) in a time period may lead to delay some major orders; and consequently lead to dissatisfaction of these customers, ultimately loss them and lead to lower profits. In addition, some inventory may remain in the warehouse in a time-period and over the time become corrupt. It also leads to reduce the benefit of supplier companies in the supply chain.
Therefore, in this paper, we will present a dynamic mathematical model to flow process /storage process of goods for order picking planning problem (OPP) in DCs. And we will optimize the number of fulfilled orders in this problem with regard to a) the coefficient of each customer, b) to meet each customer's needs in the least time c) probabilistic demand of customers, and d) taking inventory to send to customers at the earliest opportunity to prevent their decay. After presenting the mathematical model, we use Lingo software to solve small size problems. Complexity of the mathematical model will intensify by increasing the numbers of customers and products in distribution center, Therefore Lingo software will not able to solve these problems in a reasonable time. Therefore, we will develop and use a genetic algorithm (GA) for solving these problems.

1. Introduction

Order picking is a process by which products are retrieved from specified storage locations with respect to customer orders. Order picking is a labor-intensive task in warehousing, improving the performance of order picking generally can lead to a large amount of savings in warehousing costs [1]. The efficiency of order picking is dependable on factors such as the storage racks, DC layout and control mechanisms. The overall logistics service level also can

*Corresponding author
E-mail address: seyedrezai@yahoo.com

be improved through efficient warehousing operations. In an order picking operation, order pickers may pick one order at the time (single order picking). Warehouse (or DC) managers are interested in finding the most economical way of picking orders, which optimize the costs involved in terms of customer satisfactions or time. In addition, many developed researches considered demand in deterministic manner, whereas in the real world it is not deterministic because of changing in requirements and shorter product life cycles.

Therefore, in this study, we estimate the demand of each item types in each period according to the previous periods and probability distribution function and develop a mathematical model for OPP as the decision process to optimize fulfilled orders in a DC with respect to the perishable goods. Although this step is very important for large DCs, but developing dynamic mathematical models to multi time-period has not received attention from the academia and practitioners so far.

2. Literature review

In the following two subsections, a brief description of the work, which has been done to address the design models for a distribution center, researches, have been developed in Order picking system design models would be briefly discussed.

2.1.Designing models for a distribution center (warehouse)

Designing a DC (or a warehouse) is a complex task involving a number of design parameters. A designer is face with the challenge of how best to address all the designing issues without increasing the complexity of problem.

Several models have been presented in the literature to design a DC. Ashayeri and Gelders [2] review various analytical- and simulation-based approaches proposed for warehouse designing problems. They conclude that, in general, neither a pure analytical approach nor a pure simulation approach will lead to a practical designing method. As a potential solution, they suggest a two-step procedure for warehouse designing. As a first step of this procedure, analytical models could be used to quickly compare different designing alternatives and to select only a few feasible alternatives, thus, reducing the search space. As a second step, an elaborate simulation model can be built to include dynamics that could not be modeled using analytical approaches.

Gray et al. [3] propose a multi-stage hierarchical decision approach to model the composite design and operating problems for a typical order-consolidation warehouse. Annualized incremental initial costs plus the warehouse operating costs are minimized as an objective function. Van den Berg [4] presents a literature survey on methods and techniques for the planning and control of warehousing systems. Planning-related decisions include inventory management and storage location assignment, while control-related decisions include picker-routing, sequencing, scheduling, and order batching. Rouwenhorst et al. [5] present a reference framework and a classification of warehouse design and control problems. They define a warehouse design as a structured approach of decision making at a strategic, tactical, and operational level in an attempt to meet a number of well-defined performance criteria.

For efficient order picking planning, researchers have elaborated on warehouse layout, storage policy, and picking policy [7]. Warehouse layout problem addresses the shape of warehouse, shelf location, number of aisles, location, and number of starting/ending points of picking operation [8, 9]. Storage policy includes random storage, dedicated storage, class-based storage, and volume-based storage [7].

2.2.Order picking system designing models

Order picking has been identified as a major activity in a DC and the prime contributor to overall DC operating expenses. Several research contributions propose models for the design and selection of an appropriate orders picking system (OPS) - for a given application.

Yoon and Sharp [10] present a structured procedure for the analysis and design of order picking systems that considers interdependent relationships between different functional areas (e.g., receiving, picking, sorting, etc.) of OPS.

Ho and Tseng [11] have studied the order-batching methods for a DC with two cross-aisles; and conduct an extensive experiment over the combinations of nine seed-order selection rules and ten accompanying-order selection rules, along with two route-planning methods and two picking frequency distributions. Picker routing problem is to determine how many pickers to travel along what sequence in order to minimize the total travel distance. Many researchers have studied this problem. Among those, Hsieh and Tsai [12] conduct a simulation over a set of variables such as the number of cross-aisles, storage policy, order combination, and picking policy in order to find out an efficient design of the warehouse. Petersen and Aase [7] conduct a simulation study to compare the impact of picking policy, storage policy, and routing policy on the picking efficiency; and concluded that the total picking time can be reduced the most by order combination, followed by storage policy and routing policy. A recent review paper by Gu, Goetschalckx, and McGinnis [13] identifies order picking as one area of warehouse operations. They classify order picking into batching, sequencing and routing, and sorting, all of which belong to operational level, not planning level.

Chen and Wu [14] describe the development of an order-batching approach based on data mining and integer programming; and present an order-clustering model based on 0–1 integer programming to maximize the associations between orders within each batch. Daniels, Rummel, and Schantz [15] address the problem in which items may be stored in multiple locations so that order picking requires choosing a subset of the locations that store the required item to collect the required quantity. They formulate a model for simultaneously determining the assignment and sequencing decisions, and compare it to previous models for order picking. De Koster, van der Poort, and Wolters [16] propose two groups of heuristic algorithms for efficient order batching, the Seed algorithms, and the Time Savings algorithms, and evaluate the performance of the algorithms using two different routing strategies: The so-called S-shape and largest gap strategies. Won and Olafsson [17] formulate the batching and order-picking problem jointly as a combinatorial optimization problem; and evaluate the benefits of addressing the joint problem and propose simple but effective heuristics for its solution. Gademann and van de Velde [18] address problem of batching orders in a parallel-aisle warehouse to minimize the total traveling time needed to pick all items. They model the problem as a generalized set partitioning problem and present a column generation algorithm to solve its linear programming relaxation. For further research results on order picking, the interested readers may refer to [13].

Parikh [6] designed an order picking systems for DCs. In this paper, he focused on the following two keys in orders picking system-designing issues: configuration of the storage system and selection between batch and zone order picking strategies. The overall goal of his research is to develop a set of analytical models to help the designer in designing order-picking systems in a distribution center.

In distribution centers, some orders may not be fulfilled due to the shortage of inventory. Rim and Park [19] considered the case where an order is picked and shipped only when all items in the order is available in the inventory; and the unfulfilled orders are transferred to the next day with higher priority. Given daily inventory and a set of orders arrived during a day, the problem was to assign the inventory to the orders so as the order fill rate (OFR) to maximize. They formulated this problem as a linear programming problem. They also consider the weighted OFR to reflect the importance of each order. This problem do not dale with in a dynamic manner. This is a practical model but it does not take into consideration certain issues that are addressed within this present paper: forecasting the demand of each customer according to the demand function, capacity limitation of warehouse in DC, and considering perishable goods in planning horizon of multiple time-periods.

The proposed model belongs to the class of NP-complete problems. Therefore, we use genetic algorithm to solve this model.

3. Demand Forecasting Methods

Forecasting is an attempt to determine in advance the most likely outcome of an uncertain variable. Planning and controlling logistics systems need predictions for the level of future economic activities because of the time lag in matching supply to demand.

Throughout the literature review and mathematical models development, emphasis has been put upon the deterministic demand of items and stable production, whereas in the real world, demand is not deterministic and must be estimated according to the Probability and statistic methods. In this research, we supposed demand of items in orders are discreet and have Binomial distribution. The essential parameters to estimate the demand of items in an order are:

$\mathrm{P}_{ij}(t)$	Probability of demand item *j* in order *i* at time period *t*
$\mathrm{S}D_{ij}(t)$	Standard deviation of demand item *j* in order *i* at time period *t*
$\mathrm{E}D_{ij}(t)$	Mathematical expectation of demand item *j* in order *i* at time period *t* according to the demand function
n_{ij}	Quantity of demand item *j* in order customer *i* at pervious time periods
$1-\alpha$	Confidence interval
$Z_{1-\frac{\alpha}{2}}$	Is a point s.t $P(z > Z_{1-\frac{\alpha}{2}}) = \alpha/2$, Z is the normal distribution

The following relations calculate the mean and standard deviation of demand function

$$\mathrm{E}D_{\mathrm{ij}}(t) = n_{\mathrm{ij}} \times P_{\mathrm{ij}}(t) \qquad \forall i \in I, \forall j \in J \tag{1a}$$

$$\mathrm{S}D_{\mathrm{ij}}(t) = \sqrt{n_{\mathrm{ij}} \times P_{\mathrm{ij}}(t) \times (1 - P_{\mathrm{ij}}(t))} \qquad \forall i \in I, \forall j \in J \tag{1b}$$

According to the Central limit theorem, we can estimate the binomial distribution by normal distribution. We suppose confidence interval of item demand is 95%, so Inequality (1c) ensures a 95% Confidence interval for demand of items in each order.

$$\mathrm{E}D_{\mathrm{ij}}(t) - Z_{1-\frac{\alpha}{2}} \times \mathrm{S}D_{\mathrm{ij}}(t) \le D_{\mathrm{ij}}(t) \le \mathrm{E}D_{\mathrm{ij}}(t) + Z_{1-\frac{\alpha}{2}} \times \mathrm{S}D_{\mathrm{ij}}(t) \qquad \forall i \in I, \forall j \in J, \forall t \in T \tag{1c}$$

4. Mathematical model

Suppose there is a DC, which is supplied by Production centers and supplies many items to many retail stores. The customers (retailers) send orders to the DC once in every time-period. The orders-whose quantity is available from the current inventory for all items-are picked and shipped to the customers on the current time-period. The orders; which are not replied due to the shortage of at least one item in that order are carried forward to the next time-period (called back order).

Assumptions

The assumptions of the mathematical model are as following:

1. Demand of items in customer order is determined according to binomial distribution in each time-period.
2. Confidence interval for demand of items in customer order should be predetermined (95%).
3. Back-orders are allowed.
4. The order of retail centers is independent.
5. The product's life is specific.
6. Dividing of items in a customer order is not allowed.
7. The capacity of DC is limit and specific.
8. importance of each customer's order for the DC is definite and specific

Indexes

i	Index of orders (customers) (i=1,2,…,|I|)
j	Index of item types (j=1, 2,…,|J|)
t, q, k	Index t of time periods (t, q, k=1,2,…,|T|)
W_{i}	Weight of order (customer) i
$\mathrm{S}D_{\mathrm{ij}}(t)$	Standard deviation of demand function of product j for customer i in period t

$\mathrm{E}D_{\mathrm{ij}}(t)$ Mathematical expectation of demand item *j* in order *i* at time period *t* according to the demand function

Parameters

$L(j)$ Longevity of product *j* in terms of time period

$V(j)$ The space that occupied by each unit of product *j* in terms of m^3

M A large positive number

$V_{capacity}$ Capacity of distribution center

Model decision variables:

$R_{\mathrm{j}}(t)$ Inventory (remaining) of distribution center for item *j* at the end of time period *t*

$S_{\mathrm{j}}(t)$ Quantity of item *j*, must be supplied from manufacturing centers to meeting the demand of customers at time period *t*

$D_{\mathrm{ij}}(t)$ Estimated quantity of items *j* for demand of customer *i* at time period *t*

$Y_{\mathrm{ij}}(t,k)$ 1 If demand of customer *i* for product *j* in period *t* is meet period *k*; or 0 otherwise

$F_{\mathrm{i}}(t,k)$ 1 if order *i* that requested in time-period *t* is fulfilled in time-period *k*; or 0 otherwise

Mathematical model

$$Maximize\ Z = \sum_{i\in I}\sum_{t\in T}\sum_{k\geq t,k\in T}\left(\frac{W_{\mathrm{i}}}{\sum_{i\in I} W_{\mathrm{i}}}\times(T+t-k)\times F_{\mathrm{i}}(t,k)\right) \tag{2}$$

Subject to

$$\sum_{k\geq t,k\in T} Y_{\mathrm{ij}}(t,k)\leq 1 \qquad \forall i\in I,\forall j\in J,\forall t\in T \tag{3}$$

$$F_{\mathrm{i}}(t,k)\leq 1-\frac{(1-Y_{\mathrm{ij}}(t,k))}{M} \qquad \forall i\in I,\forall j\in J\ \ \forall t\in T \tag{4}$$

$$\sum_{i\in I}\sum_{t\in T}(F_{\mathrm{i}}(t,k)\times D_{\mathrm{ij}}(t))\leq S_{\mathrm{j}}(k)+R_{\mathrm{j}}(k-1) \qquad \forall j\in J\ \ \forall k\in T \tag{5}$$

$$R_{\mathrm{j}}(k)=S_{\mathrm{j}}(k)+R_{\mathrm{j}}(k-1)-\sum_{i\in I}\sum_{t\in T}(F_{\mathrm{i}}(t,k)\times D_{\mathrm{ij}}(t)) \qquad \forall j\in J\ \forall k\in T \tag{6}$$

$$R_{\mathrm{j}}(k)\leq\sum_{i\in I}\sum_{t\in T}\sum_{q\geq k+1,q\in T}^{k+L(j)-1}(F_{\mathrm{i}}(t,q)\times D_{\mathrm{ij}}(t)) \qquad \forall j\in J\ \ \forall k\in T \tag{7}$$

$$\sum_{j\in J}(S_{\mathrm{j}}(t)+R_{\mathrm{j}}(t-1)\times V_{\mathrm{j}})\leq V_{capacity} \qquad \forall t\in T \tag{8}$$

$$\mathrm{E}D_{\mathrm{ij}}(t)-Z_{1\text{-}\frac{\alpha}{2}}\times \mathrm{S}D_{\mathrm{ij}}(t)\leq D_{\mathrm{ij}}(t) \qquad \forall i\in I,\forall j\in J,\forall t\in T \tag{9}$$

$$D_{\mathrm{ij}}(t)\leq \mathrm{E}D_{\mathrm{ij}}(t)+Z_{1\text{-}\frac{\alpha}{2}}\times \mathrm{S}D_{\mathrm{ij}}(t) \qquad \forall i\in I,\forall j\in J,\forall t\in T \tag{10}$$

$$\sum_{k\geq t, k\in T} F_i(t,k) \leq 1 \qquad \forall i \in I,\ \forall t \in T \tag{11}$$

$$F_i(t,k) = 0 \qquad \forall i \in I, \forall j \in J, \forall t,k \in T, t > k \tag{12}$$

$$Y_{ij}(t,k) = 0 \qquad \forall i \in I, \forall j \in J, \forall t,k \in T, t > k \tag{13}$$

$$F_i(t,k) \in \{0,1\} \qquad \forall i \in I, \forall j \in J, \forall t,k \in T \tag{14}$$

$$Y_{ij}(t,k) \in \{0,1\} \qquad \forall i \in I, \forall j \in J, \forall t,k \in T \tag{15}$$

$$R_j(k) \geq 0 \quad and\ integer \qquad \forall j \in J, \forall k \in T \tag{16}$$

$$D_{ij}(t) \geq 0 \quad and\ integer \qquad \forall i \in I, \forall j \in J, \forall t \in T \tag{17}$$

$$S_j(k) \geq 0 \quad and\ integer \qquad \forall j \in J, \forall k \in T \tag{18}$$

The objective function (2) attempted to maximize the number of fulfilled orders with respect to the importance of customers and back-order coefficient. Term $w_i \big/ \sum_{i\in I} w_i$ shows importance of customer i to the other customers. This term ensures meeting demand of important customers in the higher priority. Coefficient $(T + t - k)$ guarantees assigning priority of back-orders. By optimizing of the objective function, satisfaction of customers will be increased because of customer importance and back-order coefficient. Constraint (3) shows items of an order in a time-period are supplied only once in planning horizon of multiple time-periods. Constraint (4) forces $F_i(t,k)$ to be zero if any one item of the order ***i*** in time-period ***t*** is not fulfilled by the inventory of time-period ***k***.

Constraint (5) means that the total assignment for item ***j*** in time-period ***k*** cannot exceed the inventory of the item in beginning of this time-period (the inventory of each time-period for each item ***j*** is the inventory carried over from the previous period, plus supplied amount of current period). Constraint (6) calculate remaining amount of item ***j*** at the end of each time-period. Constraint (7) guaranty remaining items in the end of each time-period in DC must be used before perishing. This constraint ensures assigning the items carried forward from previous periods to avoid perishing. Therefore, there is no perished item in DC in planning horizon of multiple time-periods and we do not allow items in DC to be perished. Constraint (8) specifies the DC capacity limitation. Constraint (9) and (10) ensure confidence interval for demand of items in each order. Constraint (11) ensures that an order of a customer in a time-period is supplied only once in planning horizon of multiple time-periods. Constraint (12) guaranty the orders of time period ***t*** can be fulfilled on the next time-periods not on the pervious time-periods(for example: the demanded order ***i*** in time period 3 can't be fulfilled in time-period 1 or 2, this order can be fulfilled in period 3 and future time-periods (4,5, …)). Constraint (13) restricts illogical meeting of items in an order. Constraints (14) - (18) are the logical binary and non-negativity integer requirements on the decision variables.

5. Genetic algorithm

In spite of other stochastic search methods, GA searches the feasibility space by setting of feasibility solutions simultaneously in order to find optimal or near-optimal solutions. The procedure is carried out by the use of genetic operations. GA was developed initially by Holland [20]. Goldberg [21] gave an interesting survey of some practical works carried out in this area.

In the next section, we describe an elitist genetic algorithm to solving our proposed mathematical model.

6. Elitist genetic algorithm for solving the OPP

In this section, an elitist genetic algorithm for solving the OPP is introduced. For designing the GA, some principle factors are described as follows:

6.1.Solution coding (chromosome structure)

The structure of the solution comprises of four matrixes (figure 5) explained below.

(1). Matrix $[F]_{\mathrm{I \times T}}$ (supplying time matrix) indicates supply time of orders in planning horizon of multiple time- periods. Each element of matrix [F] is determined by equation (19) which ***k*** is generated from the uniform distribution of U(t,T+1) (figure 1).

$$f_{\mathrm{it}} = k \qquad k \in (t, |T|+1) \qquad \forall i \in I,, \forall t \in T \tag{19}$$

$f_{\mathrm{it}} = k$ means the demand of customer ***i*** in time-period ***t*** is supplied in time period k_1.

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \cdots & f_{1k} & \cdots & f_{1T} \\ f_{21} & f_{22} & f_{23} \cdots & f_{2k} & \cdots & f_{2T} \\ f_{31} & f_{32} & f_{33} \cdots & f_{3k} & \cdots & f_{3T} \\ \vdots & \vdots & \vdots \cdots & \vdots & \cdots & \vdots \\ f_{i1} & f_{i2} & f_{i3} \cdots & f_{ik} & \cdots & f_{iT} \\ \vdots & \vdots & \vdots \quad \cdots & \vdots & \cdots & \vdots \\ f_{I1} & f_{I2} & f_{I3} \cdots & f_{Ik} & \cdots & f_{IT} \end{bmatrix}_{\mathrm{I \times T}}$$

Figure 1.Component of chromosome (supplying time matrix)

(2). Matrix [D] is a complex matrix and consists of genes relating to the forecasted amount of items for customers. Element d_{ij} in matrix $[D]_{I \times J}^{T}$ signifies forecasted demand of item ***j*** by customer ***i*** in time-period ***t***. each element of these matrixes are determined by equations (9) and (10). Figure 2 shows the structure of matrix [D].

$$D = [[D]_{I \times J}^{1} \; [D]_{I \times J}^{2} \; \ldots [D]_{I \times J}^{t} \; \ldots [D]_{I \times J}^{|T|}]$$

$$D = \left[\begin{bmatrix} d_{11} & d_{12} & \dots & d_{1J} \\ d_{21} & d_{22} & \dots & d_{2J} \\ \vdots & & & \vdots \\ d_{I1} & d_{I2} & \dots & d_{IJ} \end{bmatrix}^1 \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1J} \\ d_{21} & d_{22} & \dots & d_{2J} \\ \vdots & & & \vdots \\ d_{I1} & d_{I2} & \dots & d_{IJ} \end{bmatrix}^2 \dots \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1J} \\ d_{21} & d_{22} & \dots & d_{2J} \\ \vdots & & & \vdots \\ d_{I1} & d_{I2} & \dots & d_{IJ} \end{bmatrix}^t \dots \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1J} \\ d_{21} & d_{22} & \dots & d_{2J} \\ \vdots & & & \vdots \\ d_{I1} & d_{I2} & \dots & d_{IJ} \end{bmatrix}^T \right]$$

Figure 2. Component of chromosome (Demand matrix)

(3). Matrix $[S]_{J\times T}$ consists of genes, which indicate the quantity of items for meeting the demand of customers in planning horizon of time-periods. Each element of matrix $[S]_{J\times T}$ is specified by equation (20). Figure 3 shows the structure of matrix [S].

$$\boldsymbol{S} = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1T} \\ s_{21} & s_{22} & \dots & s_{2T} \\ \vdots & & & \vdots \\ s_{J1} & s_{J2} & \dots & s_{JT} \end{bmatrix}_{J\times T}$$

Figure 3. Component of chromosome (DC demand matrix)

$$\sum_{i\in I} D_t(i,j) - R_{jk-1} \le S_{jk} \qquad \forall j \in J,\ \forall k \in T, s.t \quad f_{it} = k \tag{20}$$

(4). Matrix $[R]_{J\times T}$ consists of genes (Figure 4) demonstrating the rest of items at the end of each time-period. Each element of this matrix is determined according to the equations (8) and (21).

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1T} \\ r_{21} & r_{22} & \dots & r_{2T} \\ \vdots & & & \vdots \\ r_{J1} & r_{J2} & \dots & r_{JT} \end{bmatrix}_{J\times T}$$

Figure 4. Component of chromosome (rest of items in DC)

$$R_{jk} = Max\{0, S_{jk} + R_{jk-1} - \sum_{i\in I} D_t(i,j)\} \qquad \forall j \in J,\ \forall k \in T, s.t \quad f_{it} = k \tag{21}$$

$$\begin{bmatrix} \mathbf{f}_{11} & \mathbf{f}_{12} & \mathbf{f}_{13} & \dots & \mathbf{f}_{1k} & \dots & \mathbf{f}_{1T} \\ \mathbf{f}_{21} & \mathbf{f}_{22} & \mathbf{f}_{23} & \dots & \mathbf{f}_{2k} & \dots & \mathbf{f}_{2T} \\ \mathbf{f}_{31} & \mathbf{f}_{32} & \mathbf{f}_{33} & \dots & \mathbf{f}_{3k} & \dots & \mathbf{f}_{3T} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ \mathbf{f}_{i1} & \mathbf{f}_{i2} & \mathbf{f}_{i3} & \dots & \mathbf{f}_{ik} & \dots & \mathbf{f}_{iT} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ \mathbf{f}_{I1} & \mathbf{f}_{I2} & \mathbf{f}_{I3} & \dots & \mathbf{f}_{Ik} & \dots & \mathbf{f}_{IT} \end{bmatrix} \left[\begin{bmatrix} \mathbf{d}_{11} & \mathbf{d}_{12} & \dots & \mathbf{d}_{1J} \\ \mathbf{d}_{21} & \mathbf{d}_{22} & \dots & \mathbf{d}_{2J} \\ \vdots & & & \vdots \\ \mathbf{d}_{I1} & \mathbf{d}_{I2} & \dots & \mathbf{d}_{IJ} \end{bmatrix}^1 \dots \begin{bmatrix} \mathbf{d}_{11} & \mathbf{d}_{12} & \dots & \mathbf{d}_{1J} \\ \mathbf{d}_{21} & \mathbf{d}_{22} & \dots & \mathbf{d}_{2J} \\ \vdots & & & \vdots \\ \mathbf{d}_{I1} & \mathbf{d}_{I2} & \dots & \mathbf{d}_{IJ} \end{bmatrix}^T \right] \begin{bmatrix} \mathbf{r}_{11} & \mathbf{r}_{12} & \dots & \mathbf{r}_{1T} \\ \mathbf{r}_{21} & \mathbf{r}_{22} & \dots & \mathbf{r}_{2T} \\ \vdots & & & \vdots \\ \mathbf{r}_{J1} & \mathbf{r}_{J2} & \dots & \mathbf{r}_{JT} \end{bmatrix} \begin{bmatrix} \mathbf{s}_{11} & \mathbf{s}_{12} & \dots & \mathbf{s}_{1T} \\ \mathbf{s}_{21} & \mathbf{s}_{22} & \dots & \mathbf{s}_{2T} \\ \vdots & & & \vdots \\ \mathbf{s}_{J1} & \mathbf{s}_{J2} & \dots & \mathbf{s}_{JT} \end{bmatrix}$$

Figure 5. Chromosome structure

6.2. Improved GA operators

In this paper, the chromosome structure is formed as a complex matrix. When the GA operators are applied, offspring chromosome may not be feasible solution, so these solutions are corrected to be feasible.

6.2.1. Crossover operator

The crossover operator produces children by exchanging information contained in the parents. In this paper, we adopt the standard one-point crossover, which randomly generates one-crossover points along the length of array. This crossover point divides each parent into two segments; subsequently, the offsprings are created by exchanging the places of the second segments. The crossover process is illustrated in figure (6a) and (6b).

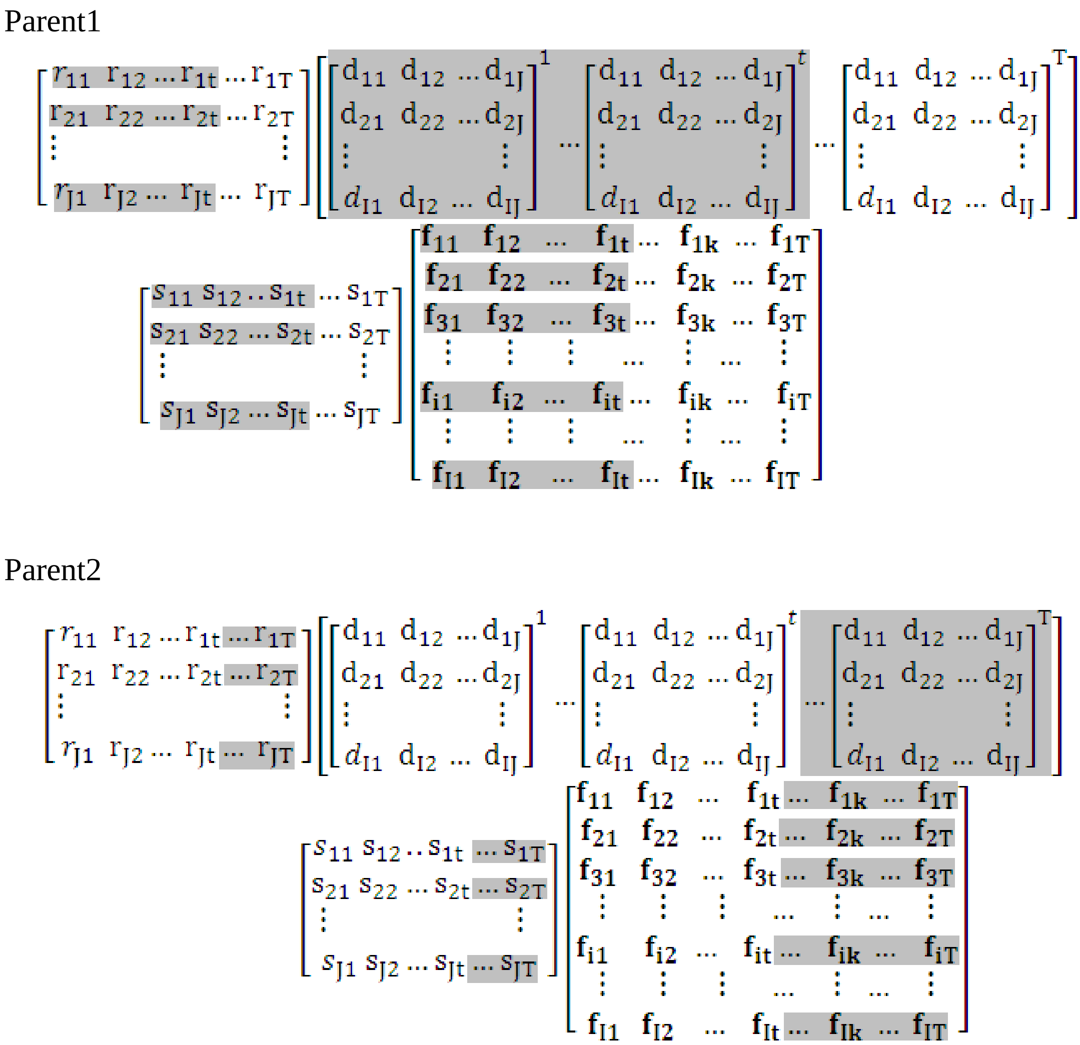

Figure 6a. Crossover operator (parents)

Offspring1

$$\begin{bmatrix} r_{11} & r_{12} \dots r_{1t} \dots r_{1T} \\ r_{21} & r_{22} \dots r_{2t} \dots r_{2T} \\ \vdots & \vdots \\ r_{J1} & r_{J2} \dots r_{Jt} \dots r_{JT} \end{bmatrix} \left[\begin{bmatrix} d_{11} & d_{12} & \dots d_{1J} \\ d_{21} & d_{22} & \dots d_{2J} \\ \vdots & & \vdots \\ d_{I1} & d_{I2} & \dots d_{IJ} \end{bmatrix}^{1} \dots \begin{bmatrix} d_{11} & d_{12} & \dots d_{1J} \\ d_{21} & d_{22} & \dots d_{2J} \\ \vdots & & \vdots \\ d_{I1} & d_{I2} & \dots d_{IJ} \end{bmatrix}^{t} \dots \begin{bmatrix} d_{11} & d_{12} & \dots d_{1J} \\ d_{21} & d_{22} & \dots d_{2J} \\ \vdots & & \vdots \\ d_{I1} & d_{I2} & \dots d_{IJ} \end{bmatrix}^{T} \right]$$

$$\begin{bmatrix} s_{11} & s_{12} \dots s_{1t} \dots s_{1T} \\ s_{21} & s_{22} \dots s_{2t} \dots s_{2T} \\ \vdots & \vdots \\ s_{J1} & s_{J2} \dots s_{Jt} \dots s_{JT} \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1t} \dots & f_{1k} & \dots & f_{1T} \\ f_{21} & f_{22} & \dots & f_{2t} \dots & f_{2k} & \dots & f_{2T} \\ f_{31} & f_{32} & \dots & f_{3t} \dots & f_{3k} & \dots & f_{3T} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ f_{i1} & f_{i2} & \dots & f_{it} \dots & f_{ik} & \dots & f_{iT} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ f_{I1} & f_{I2} & \dots & f_{It} \dots & f_{Ik} & \dots & f_{IT} \end{bmatrix}$$

Offspring2

$$\begin{bmatrix} r_{11} & r_{12} \dots r_{1t} \dots r_{1T} \\ r_{21} & r_{22} \dots r_{2t} \dots r_{2T} \\ \vdots & \vdots \\ r_{J1} & r_{J2} \dots r_{Jt} \dots r_{JT} \end{bmatrix} \left[\begin{bmatrix} d_{11} & d_{12} & \dots d_{1J} \\ d_{21} & d_{22} & \dots d_{2J} \\ \vdots & & \vdots \\ d_{I1} & d_{I2} & \dots d_{IJ} \end{bmatrix}^{1} \dots \begin{bmatrix} d_{11} & d_{12} & \dots d_{1J} \\ d_{21} & d_{22} & \dots d_{2J} \\ \vdots & & \vdots \\ d_{I1} & d_{I2} & \dots d_{IJ} \end{bmatrix}^{t} \dots \begin{bmatrix} d_{11} & d_{12} & \dots d_{1J} \\ d_{21} & d_{22} & \dots d_{2J} \\ \vdots & & \vdots \\ d_{I1} & d_{I2} & \dots d_{IJ} \end{bmatrix}^{T} \right]$$

$$\begin{bmatrix} s_{11} & s_{12} \dots s_{1t} \dots s_{1T} \\ s_{21} & s_{22} \dots s_{2t} \dots s_{2T} \\ \vdots & \vdots \\ s_{J1} & s_{J2} \dots s_{Jt} \dots s_{JT} \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1t} \dots & f_{1k} & \dots & f_{1T} \\ f_{21} & f_{22} & \dots & f_{2t} \dots & f_{2k} & \dots & f_{2T} \\ f_{31} & f_{32} & \dots & f_{3t} \dots & f_{3k} & \dots & f_{3T} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ f_{i1} & f_{i2} & \dots & f_{it} \dots & f_{ik} & \dots & f_{iT} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ f_{I1} & f_{I2} & \dots & f_{It} \dots & f_{Ik} & \dots & f_{IT} \end{bmatrix}$$

Figure 6b. (offspring) -Crossover operator

6.2.2. Mutation operator

For carrying out the mutation, three different operators are developed:

6.2.2.1.Single mutation

To implement this type of mutation, a customer in a single time period is randomly chosen, and then the mutation operator selects a different time-period, which demand of this customer can be supplied (illustrated in figure 7).

$$\begin{bmatrix} f_{11} & f_{12} & \dots & f_{1t} \dots & f_{1k} & \dots & f_{1T} \\ f_{21} & f_{22} & \dots & f_{2t} \dots & f_{2k} & \dots & f_{2T} \\ f_{31} & f_{32=3} & \dots & f_{3t} \dots & f_{3k} & \dots & f_{3T} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ f_{i1} & f_{i2} & \dots & f_{it} \dots & f_{ik} & \dots & f_{iT} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ f_{I1} & f_{I2} & \dots & f_{It} \dots & f_{Ik} & \dots & f_{IT} \end{bmatrix} \Rightarrow \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1t} \dots & f_{1k} & \dots & f_{1T} \\ f_{21} & f_{22} & \dots & f_{2t} \dots & f_{2k} & \dots & f_{2T} \\ f_{31} & f_{32=2} & \dots & f_{3t} \dots & f_{3k} & \dots & f_{3T} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ f_{i1} & f_{i2} & \dots & f_{it} \dots & f_{ik} & \dots & f_{iT} \\ \vdots & \vdots & \vdots & \dots & \vdots & \dots & \vdots \\ f_{I1} & f_{I2} & \dots & f_{It} \dots & f_{Ik} & \dots & f_{IT} \end{bmatrix}$$

Figure 7. Single mutation

6.2.2.2.Multi row mutation

The multi mutation performs the single mutation on all the orders of a picked customer in planning horizon of multiple time-periods (illustrated in figure 8).

$$\begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1t} & \cdots & f_{1k} & \cdots & f_{1T} \\ f_{21} & f_{22} & \cdots & f_{2t} & \cdots & f_{2k} & \cdots & f_{2T} \\ f_{31} & f_{32} & \cdots & f_{3t} & \cdots & f_{3k} & \cdots & f_{3T} \\ \vdots & \vdots & & \vdots & & \vdots & \cdots & \vdots \\ 1 & 2 & & t+3 & & k+0 & & T \\ \vdots & \vdots & & \vdots & \cdots & \vdots & \cdots & \vdots \\ f_{I1} & f_{I2} & \cdots & f_{It} & \cdots & f_{Ik} & \cdots & f_{IT} \end{bmatrix} \Rightarrow \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1t} & \cdots & f_{1k} & \cdots & f_{1T} \\ f_{21} & f_{22} & \cdots & f_{2t} & \cdots & f_{2k} & \cdots & f_{2T} \\ f_{31} & f_{32} & \cdots & f_{3t} & \cdots & f_{3k} & \cdots & f_{3T} \\ \vdots & \vdots & & \vdots & & \vdots & \cdots & \vdots \\ 2 & 3 & & t+1 & & k+1 & & T \\ \vdots & \vdots & & \vdots & \cdots & \vdots & \cdots & \vdots \\ f_{I1} & f_{I2} & \cdots & f_{It} & \cdots & f_{Ik} & \cdots & f_{IT} \end{bmatrix}$$

Figure 8. Multi row mutation

6.2.2.3.Multi column mutation

This type of mutation performs the single mutation on all the orders of a picked time-period (illustrated in figure 9).

$$\begin{bmatrix} f_{11} & f_{12=4} & \cdots & f_{1t} & \cdots & f_{1k} & \cdots & f_{1T} \\ f_{21} & f_{22=2} & \cdots & f_{2t} & \cdots & f_{2k} & \cdots & f_{2T} \\ f_{31} & f_{32=3} & \cdots & f_{3t} & \cdots & f_{3k} & \cdots & f_{3T} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \cdots & & \vdots \\ f_{i1} & f_{i2=3} & \cdots & f_{it} & \cdots & f_{ik} & \cdots & f_{iT} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \cdots & & \vdots \\ f_{I1} & f_{I2=5} & \cdots & f_{It} & \cdots & f_{Ik} & \cdots & f_{IT} \end{bmatrix} \Rightarrow \begin{bmatrix} f_{11} & f_{12=2} & \cdots & f_{1t} & \cdots & f_{1k} & \cdots & f_{1T} \\ f_{21} & f_{22=5} & \cdots & f_{2t} & \cdots & f_{2k} & \cdots & f_{2T} \\ f_{31} & f_{32=4} & \cdots & f_{3t} & \cdots & f_{3k} & \cdots & f_{3T} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \cdots & & \vdots \\ f_{i1} & f_{i2=2} & \cdots & f_{it} & \cdots & f_{ik} & \cdots & f_{iT} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \cdots & & \vdots \\ f_{I1} & f_{I2=2} & \cdots & f_{It} & \cdots & f_{Ik} & \cdots & f_{IT} \end{bmatrix}$$

Figure 9. Multi column mutation

6.3.Mating pool selection strategy

For creating the new generation, it is necessary to select some chromosomes (mating pool) with the latest fitness in the current generation for recombining or creating chromosomes related to the new generation. In this case, tournament selection, and Elitist selection are used in which the fitness of current generation chromosomes is calculated according to objective function. Figure (10) shows the procedure of mating pool selection strategy.

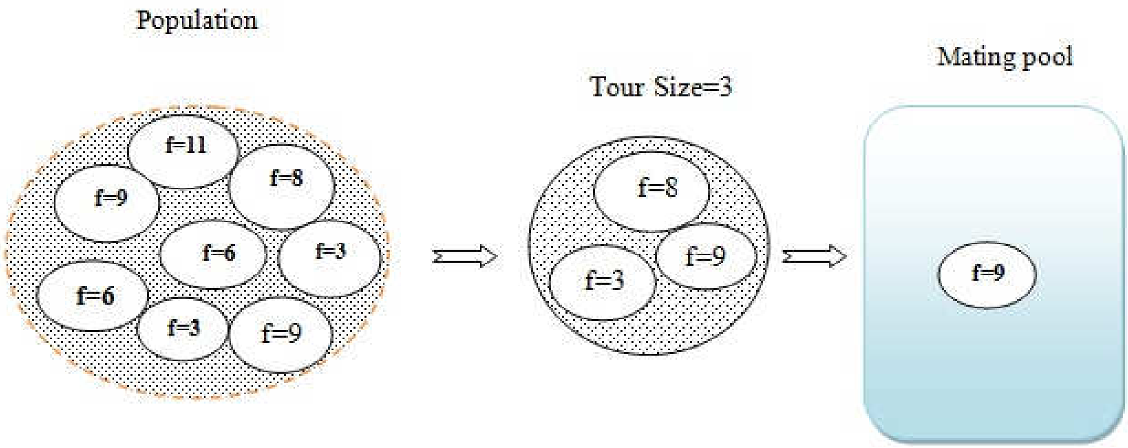

Figure 10. Mating pool selection strategy

6.4.Stopping criteria

The algorithm terminates, if the number of generations exceeds the specific number.

6.5. Elitist GA procedure

Figure (11) shows the flowchart of GA. The steps need to implement GA are as follow:
Step 1: Create a random feasible parent population P0 of population size, Set t=0.
*Step 2***:** Assign a fitness value to each solution in P0 according to the objective function
Step 3: Use tournament selection and Elitist selection with their rates to selecting chromosomes from P_t and creating mating pool members of population size.
*Step 4***:** generate a new population of population size as fallow:

4-1: chose tow solutions from mating pool with predefined crossover rate to generate offspring and add them to population of next generation (P_{t+1}).

4-2: mute each solution in mating pool with predefined mutation rate and add it to population of next generation (P_{t+1}).

4-3: select the rest of population size form mating pool according to the best objective function (fitness value) and copy them in to the population of next generation (P_{t+1}).

Step 5: If the stopping criterion is satisfied, stop and return the best solution of the generation.
Step 6: Set t =t+1, and go to *Step 3.*

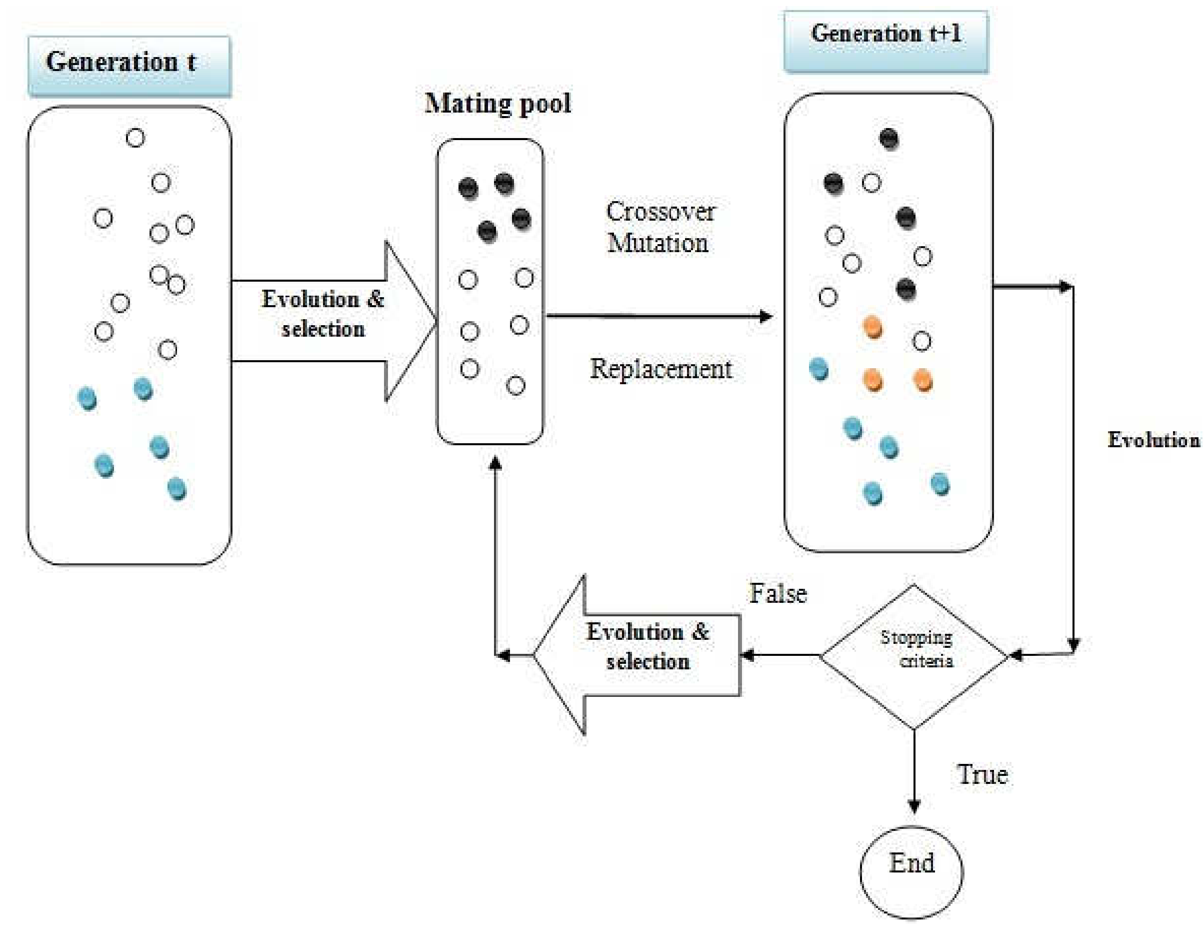

Figure 11. Flowchart of elitist GA

7. Computational results

In order to shows the efficiency of developed GA; we solve 10 test problems by this algorithm and Lingo 9 on a PC Pentium IV 3.06 GHz processor, 1024 MB RAM, and windows XP Professional Operating System. The GA was developed using Matlab7.0. The input parameters used in GA for all problems are showed in Table 1. The obtained results are shown in Table 2.

Table 1. Parameters used in GA

Iterations	Population size	Elitism	Mutation rate	Crossover rate
100	100	0.02	0.1	0.88

Table 2. Comparing result of GA and Lingo

Problem		Lingo			GA		
Ex. n	**I × J × T**	**Time (S)**	**No. fulfilled orders**	**Obj**	**Time (S)**	**No. fulfilled orders**	**Obj**
1	$\mathbf{4 \times 4 \times 3}$	**79**	**12**	**9**	**5.7**	**12**	**8.48**
2	$\mathbf{4 \times 5 \times 4}$	**179**	**20**	**16**	**5.3**	**16**	**14.27**
3	$\mathbf{8 \times 6 \times 4}$	**661**	**32**	**16**	**6.7**	**32**	**14.15**
4	$\mathbf{8 \times 8 \times 4}$	**982**	**31**	**15.97**	**10.5**	**32**	**13.99**
5	$\mathbf{8 \times 8 \times 5}$	**1010**	**38**	**24.92**	**13**	**40**	**21.99**
6	$\mathbf{8 \times 10 \times 7}$	**1659**	**56**	**48.96**	**15.1**	**56**	**43.28**
7	$\mathbf{12 \times 10 \times 7}$	**2258**	**84**	**48.97**	**15.3**	**84**	**43**
8	$\mathbf{13 \times 12 \times 7}$	**3911**	**91**	**48.91**	**18.5**	**91**	**43.2**
9	$\mathbf{13 \times 12 \times 12}$	**12173**	**153**	**143.2**	**28**	**156**	**145.6**
10	$\mathbf{18 \times 12 \times 12}$	**31133**	**197**	**160.3**	**35**	**212**	**178.5**

Problems (9) and (10) have been feasible after 12173 and 31133 seconds respectively. We interrupted running of these problems before obtaining global optimal solutions.

7.1.Comparison metrics

To compare the result obtained by two solving method, we define the three following metrics and compare the solutions according to the metrics.

1- *Elapsed CPU time*: this metric shows the elapsed time to solve the problems. By considering the result of Table 2 and this metric, the developed GA performs better than Lingo software.

2- *Number of fulfilled orders:* this metric demonstrate the number of fulfilled orders in planning horizon of multiple time-periods. According this metric, if number of fulfilled orders in a procedure is more another, the procedure performs better. Table 3 and Figure (8) compare the result obtained by GA and Lingo.

3- *Objective (fitness) function:* this metric represents fitness of each solution. In addition, this metric shows the primacy of solution procedure

According to the Comparison metrics and considering the results of Table 2, when the complexity of problems increase (because of excessive number of variables and constraints) lingo can' not solve these problems in reasonable time. However, the elapsed time to solve these problems by GA is very noticeable and quality of these solutions approximately is as same as Lingo procedure. Figures (12) and (13) show quality of solutions that have produced by GA and Lingo software. These result shows efficiency of developed GA.

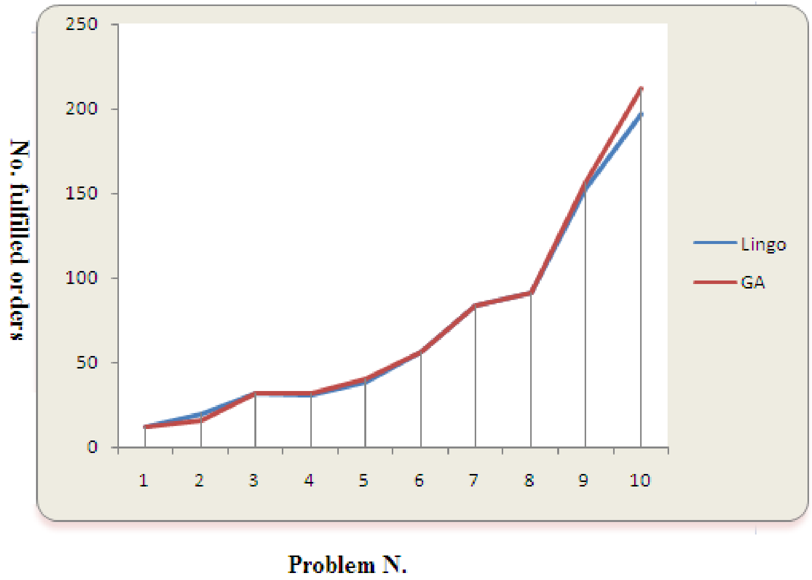

Figure 12. Resulted number of fulfilled orders in GA and Lingo

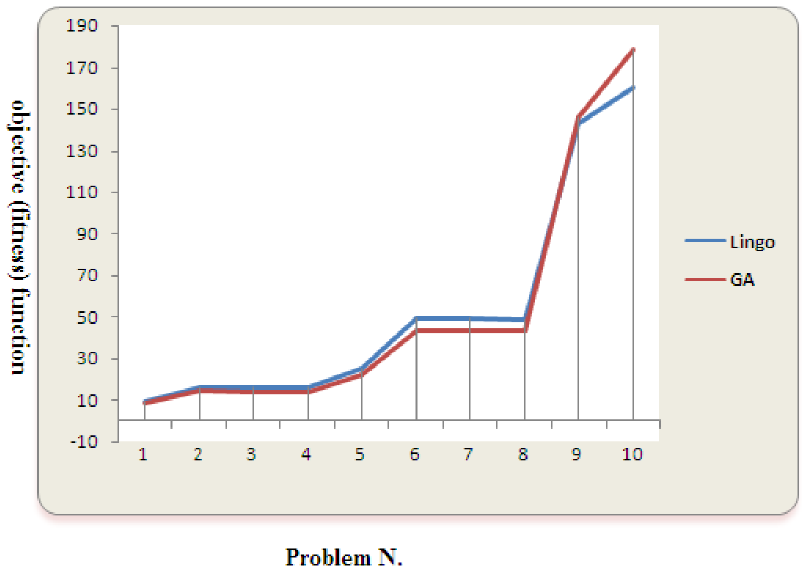

Figure 13. Quality (objective function) of obtained solutions according to the GA and Lingo

8. Conclusions

In this article, we have presented a new dynamic mathematical model for OPP with considering product life, customer importance, probabilistic demand, and back-order strategy to meeting demand of retailers in a DC. In addition, we have designed principles of GA to solve the proposed model because of its complexity. Finally, we have solved numerous examples to compare the result of GA and lingo. Numerical examples showed that the proposed GA is efficient and effective in searching for near optimal solutions. According to the comparing result and model's properties, we can discover using importance of this model; so using the factors considered in this paper helps to the DC manager system in getting satisfaction of customers in the competitive market. Minimizing item-handling cost in this model and using the other meta-heuristics to solve this problem are investigated as the next researches.

References

[1] Tompkins, J.A., White, J.A., Bozer, Y.A., Frazelle, E.H., Tanchoco, J.M.A. and Trevino, J. (1996), *Facilities Planning*, 2nd ed. Wiley, New York.

[2] Ashayeri, J. and Gelders, L.F. (1985), Warehouse design optimization, *European Journal of Operation Research*, Vol. 21, No. 3, pp. 285-294.

[3] Gray, A.E., Karmarkar, U.S. and Seidmann, A. (1992), Design and operation of an order-consolidation warehouse models and application, *European Journal of Operational Research*, Vol. 58, No. 1, pp. 14-36.

[4] Van den Berg, J.P. (1999), A literature survey on planning and control of warehousing systems, IIE Transactions, Vol. 31, No. 8, pp. 751-762.

[5] Rouwenhorst, B., Reuter, B., Stockrahm, V., van Houtum, G.J., Mantel, R.J. and Zijm, W.H.M. (2000), Warehouse design and control: Framework and literature review, *European Journal of Operational Research*, Vol. 122, No. 3, pp. 515-533.

[6] Parikh, P.J. (2006), *Designing Order Picking Systems for Distribution Centers*, Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Industrial and Systems Engineering.

[7] Petersen, C.G. and Aase, G. (2004), A comparison of picking, storage, and routing policies in manual order picking, *International Journal of Production Economics*, Vol. 92, No. 1, pp. 11–19.

[8] Ratliff, H.D. and Rosenthal, S. (1983), Order picking in a rectangular warehouse: A solvable case of the traveling salesman problem, *Operation Research*, Vol. 31, No. 3, pp. 507–521.

[9] Vaughan, T.S. and Petersen, C.G. (1999), The effect of cross aisles on order picking efficiency, *International Journal of Production Research*, Vol. 37, No. 4, pp. 887–897.

[10] Yoon, C.S. and Sharp, G.P. (1995), Example application of the cognitive design procedure for an order pick system: Case study, *European Journal of Operational Research*, Vol. 87, No. 2, pp. 223-246.

[11] Ho, Y.C. and Tseng, Y.Y. (2006), A study on order-batching methods of order-picking in a distribution centre with two cross-aisle, *International Journal of Production Research*, Vol. 44, No. 17, pp. 3391–3417.

[12] Hsieh, L. and Tsai, L. (2006), The optimum design of a warehouse system on order picking efficiency, *International Journal of Advanced Manufacturing Technology*, Vol. 28, No. 5/6, pp. 626–637.

[13] Gu, J., Goetschalckx, M. and McGinnis, L.F. (2007), Research on warehouse operation: A comprehensive review, *European Journal of Operational Research*, Vol. 177, No. 1–21.

[14] Chen, M.C. and Wu, H.P. An association-based clustering approach to order batching considering customer demand patterns, *Omega*, Vol. 33, No. 4, pp. 333–343.

[15] Daniels, R.L., Rummel, J.L. and Schantz, R. (1998), A model for warehouse order picking, *European Journal of Operational Research*, Vol. 105, pp. 1–17.

[16] De Koster, M.B.M., van der Poort, E.S. and Wolters, M. (1999), Efficient order batching methods in warehouses, *International Journal of Production Research*, Vol. 37, No. 7, pp. 1479–1504.

[17] Won, J. and Olafsson, S. (2005), Joint order batching and order picking in warehouse operations, *International Journal of Production Research*, Vol. 43, No. 7, pp. 1427–1442.

[18] Gademann, N. and van de Velde, S. (2005), Order batching to minimize total travel time in a parallel-aisle warehouse, IIE Transactions, Vol. 37, No. 1, pp. 63–75.

[19] Rim, S.C. and Park, I.S. (2008), Order picking plan to maximize the order fill rate, *International Journal of Computers & Industrial Engineering*, Vol. 55, pp. 557–566.

[20] Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbon.

[21] Goldberg, D.E. (1989), *Genetic Algorithm in Search Optimization, and Machine Learning*, Addison, Wesley, Massachusetts.