A Parameter-Tuned Genetic Algorithm for Multi-Product Economic Production Quantity Model with Space Constraint, Discrete Delivery Orders and Shortages

Seyed Taghi Akhavan Niaki, Ph.D., Professor
Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran
Phone: +98 21 66165740, Fax: +98 21 66022702, e-mail: niaki@sharif.edu

Jalil Aryan, M.Sc. Student
Department of Management and Accounting, Shahid Beheshti University, Tehran, Iran
Phone: +98 (21) 29902383, Fax: +98 (21) 22431843, e-mail: jalil.1361@yahoo.com

Seyed Hamid Reza Pasandideh, Ph.D., Assistant Professor
Department of Management and Accounting, Shahid Beheshti University, Tehran, Iran
Phone: +98 (21) 29902383, Fax: +98 (21) 22431843, e-mail: SHR_pasandideh@sbu.ac.ir

Abstract
In this paper, a multi-product economic production quantity problem with limited warehouse-space is considered in which the orders are delivered discretely in the form of multiple pallets and the shortages are completely backlogged. We show that the model of the problem is a constrained nonlinear integer program and propose a genetic algorithm to solve it. Moreover, design of experiments is employed to calibrate the parameters of the algorithm for different problem sizes. At the end, a numerical example is presented to demonstrate the application of the proposed methodology.

Keywords: Genetic algorithm; inventory management; economic production quantity; discrete delivery; shortage; backorder; design of experiments
1. Introduction and literature review

The economic production quantity (EPQ) is one of the most applicable models in production and inventory control environments. This model can be considered as an extension to the well-known economic order quantity (EOQ) model that was introduced by Harris [1] in 1913. Regardless of the simplicity of EOQ and EPQ, they are still applied industry-wide today [2].

In spite of such acceptance, some practitioners and researchers have questioned its practical applications due to several unrealistic assumptions regarding model input parameters. These parameters are setup costs, holding costs and demand rate. For example Woolsey [3] severely critiqued the use of the EOQ model, arguing that the assumptions (i.e. constant demand, constant carrying capacity, constant price, unlimited storage capacity, and paying for the price of items as soon as they are received) necessary to justify the use of this model are not met in real world environment. This has motivated many researchers to modify the EPQ model to match real-life situations. Chang et al. [4] developed an EOQ model for deteriorating items, in which the supplier provides a permissible delay to the purchaser if the order quantity is greater than or equal to a predetermined quantity. Li et al. [5] developed EPQ-based models with planned backorders to evaluate the impact of the postponement strategy on a manufacturer in a supply chain. They derived the optimal total average costs per unit time for producing and keeping end-products in a postponement system and a non-postponement system, respectively.

Another key assumption of both the basic EOQ and EPQ models is that stock-outs are not permitted. In inventory systems, when a shortage occurs, unfilled demands become either backorders or lost sales. If all customers cancel their orders and turn them to other suppliers lost sales will result (see [6] for an instance.) However, when all customers are willing to wait for delivery a full backorder occurs. Wee et al. [7] developed an optimal inventory model for items with imperfect quality and shortage backordering. Many researchers have studied
inventory models with partial backordering which is a mixture of back orders and lost sales. San Jose et al. [8] studied an inventory model with partial backlogging, where unsatisfied demand is partially backlogged according to an exponential function.

In recent years, several researchers have applied genetic algorithms (GAs) as an optimization technique to solve the production/inventory problems. For example, Zhao and Wang [9] developed an EOQ model for multi-item and multi-storehouse with limited funds, limited storage capacity and stochastic demand. In order to solve the model, they provided a hybrid genetic algorithm that combines self-adapting crossover operator and mutation operator. There are several interesting and relevant papers related to the application of GA in inventory problems such as Stockton and Quinn [10], Mondal and Maiti [11], Hou et al. [12], Gupta et al. [13], Lotfi [14], Pal et al. [15], and Taleizadeh et al. [16,17,18,19].

Pasandideh and Niaki [20] developed a multi-product EPQ model with limited warehouse space. They assumed that the orders may be delivered discretely in the form of multiple pallets. In their models shortages and delays were not permitted. Under these conditions, they formulated the problem as a non-linear integer-programming model and proposed a genetic algorithm to solve it.

In this paper, we extend Pasandideh and Niaki [20] model to include shortages. We assume that all shortages are completely backordered. Other conditions and constraints are the same as their work.

The rest of the paper is organized as follows. The problem along with its assumptions is defined in Section 2. In section 3 the problem is mathematically formulated. After a brief introduction in section 4, a genetic algorithm is proposed to solve the model. In Section 5, design of experiments (DOE) is used to analyze the performance of the proposed GA. This analysis is important in improving the performances of the proposed GA by identifying the optimal values of its control parameters. Section 6 includes results of applying the proposed
genetic algorithm to a numerical example. The conclusion and recommendations for future research are given in section 7.

2. Problem definition

Consider a production company that works with a supplier. The situations by which the company and the supplier interact with each other are defined as follow:

   a) The supplier produces all of the demanded products with known and constant rates.
   b) The demand of each product in the company is known with a constant rate.
   c) The supplier sends the orders to the company by pallets.
   d) The company pays the transportation cost of each pallet.
   e) The company determines the capacity of each pallet and the number of shipments.
   f) The warehouse space of the company for all products is limited.
   g) The setup and holding costs are known.
   h) Shortages are allowed and unsatisfied demands are fully backlogged.

The problem is to determine the order quantity, the pallet capacity, the number of shipments for each product, and the maximum shortage level of each product such that the total inventory cost is minimized while the constraints are satisfied.

3. Problem modeling

In order to mathematically formulate the problem we take advantage of the classical EPQ model and extend it to the problem at hand. We note that while in the classical EPQ model orders are produced by the supplier with constant and continuous rates, are shipped to the customer, and there is no limitation on the warehouse space, for the problem at hand the deliveries are made in the form of several discrete pallets and the space is limited.
In order to model the problem, first we define the parameters and the variables in Section 3.1. Then, we pictorially demonstrate the situation by inventory graphs in Section 3.2. Different costs are derived in Section 3.3. Finally, we present the model of the problem in Section 3.4.

3.1. Variables and parameters

For products \( i = 1,...,n \), we define the variables and the parameters of the model as follows:

\[ D_i \quad \text{Demand rate} \]
\[ P_i \quad \text{Production rate} \]
\[ Q_i \quad \text{Order quantity} \]
\[ T_i \quad \text{Cycle time} \]
\[ T_{p_i} \quad \text{Effective production time per cycle} \]
\[ T_{d_i} \quad \text{Non effective production (down) time per cycle} \]
\[ t_i \quad \text{Time between two consecutive pallet shipments} \]
\[ k_i \quad \text{Pallet capacity} \]
\[ m_i \quad \text{Number of shipments per cycle} \]
\[ B_i \quad \text{Maximum shortage (backorder) level} \]
\[ I_{\max} \quad \text{Maximum on-hand inventory level} \]
\[ f_i \quad \text{Space occupied by each unit} \]
\[ d_i \quad \text{Transportation cost per shipment} \]
\[ c_i \quad \text{Providence cost per unit} \]
\( A_i \)  Setup cost per cycle
\( b_i \)  Backordering cost per unit
\( h_i \)  Holding cost per unit
\( TT_i \)  Total transportation costs per year
\( TP_i \)  Total providence costs per year
\( TS_i \)  Total setup costs per year
\( TB_i \)  Total backordering costs per year
\( TH_i \)  Total holding costs per year
\( TC \)  Total costs of all products per year.
\( n \)  Number of products
\( f \)  Available warehouse space for all products

### 3.2. Inventory graph

The situation of the inventory problem of this research is similar to the one of EPQ model; the differences are in the delivery types and the shortages. In this paper, an order of the \( i^{th} \) product, after being produced by the supplier, will be delivered to the company in \( m_i \) pallets each with capacity of \( k_i \). A graph of the inventory position of product \( i \) over time is illustrated in Figure (1).

**Insert Figure (1) about here**

In Figure (1), each jump in \( T_{d_i} \) section shows a delivery of a pallet to the company with capacity of \( k_i \). During \( T_{d_i} \) and \( T_{p_i} \) sections the company consumes the delivered products at constant rate. When the inventory position is positive, there is inventory on hand
and the company incurs holding cost. When the inventory is negative, the demand is backordered and the company incurs backorder cost. In order to calculate the holding and backordering costs, we need to know the number of jumps in $T_d$ section. The total number of jumps is $m_i$. Accordingly, $Q_i = m_i k_i$ is correct.

Referring to Figure (1), during the interval $(0, 2t)$ the supplier faces backorder cost only. However, while during interval $(2t, 5t)$ the supplier faces both holding and backordering costs, in interval $(5t, 7t)$ he faces only holding costs. Let $Z_1$ be the number of jumps in interval $[0, 2t]$ and $Z_2$ be the number of the jumps in interval $[0, 5t]$. Then, it turns out that

$$Z_1 = \left\lfloor \frac{B - Dt}{k - Dt} \right\rfloor$$

(1)

$$Z_2 = \left\lfloor \frac{B}{k - Dt} \right\rfloor$$

(2)

Hence, the number of the jumps during interval $[0, 2t]$ is $Z_1$, during interval $[2t, 5t]$ it is $Z_2 - Z_1$ and in interval $[5t, 7t]$ it is $m_i - 1 - z_2$. For instance in Figure (1), $Z_1 = 2$, $Z_2 = 5$ and $m_i = 8$.

### 3.3. Costs calculations

The total annual cost of all products ($TC$) is the sum of total transportation costs

$$(\sum_{i=1}^{n} TT_i), \text{ total providence costs } (\sum_{i=1}^{n} TP_i), \text{ total setup costs } (\sum_{i=1}^{n} TS_i), \text{ total backordering cost}$$

$$(\sum_{i=1}^{n} TB_i) \text{ and total holding costs } (\sum_{i=1}^{n} TH_i) \text{ of all products per year. Therefore,}$$

$$TC = \sum_{i=1}^{n}(TT_i + TP_i + TS_i + TB_i + TH_i)$$

(3)
3.3.1 Total transportation costs ($TT_i$)

As the transportation cost depends on the number of shipments, it will be $m_i d_i$ in each cycle. In addition, the number of cycles for each product is given by $D_i / Q_i$. Thus

$$TT_i = m_i d_i \frac{D_i}{Q_i} = \frac{Q_i}{k_i} d_i \frac{D_i}{Q_i} = d_i \frac{D_i}{k_i} ; \quad i = 1, 2, \ldots, n \quad (4)$$

3.3.2 Total providence costs ($TP_i$)

Since the annual rate of demand for each product is known, total providence cost of product $i$ is obtained as

$$TS_i = c_i D_i ; \quad i = 1, 2, \ldots, n \quad (5)$$

3.3.3 Total setup costs ($TS_i$)

As the total setup cost of product $i$ depends on the number of cycles, it can be obtained by

$$TS_i = A_i \frac{D_i}{Q_i} ; \quad i = 1, \ldots, n \quad (6)$$

3.3.4 Total backordering cost ($TB_i$)

In order to model the backordering cost, we split the backorder areas of Figure (1) into three sections. The first section corresponds to interval $(0, 2t)$ which is made up of a collection of trapezoids located in the left side of each period. The second section is located on the interval $(2t, 5t)$ and is made up of a collection of triangles. The last section corresponds to the areas of triangles in the right side of each period.
For the first section, let $AZLS_i(j)$ represent the area of trapezoid $j$ for the $i^{th}$ product.

Then for the areas of trapezoid 1 and 2 we have

$$AZLS_i(1) = \int_0^t (-Dx - B + k) \, dx = -\frac{1}{2}Dt^2 + (k - B)t$$  \hspace{1cm} (7)

$$AZLS_i(2) = \int_t^{2t} (-Dx - B + 2k) \, dx = -\frac{3}{2}Dt^2 + (2k - B)t$$  \hspace{1cm} (8)

Hence, in general, the area of trapezoid $j$ of product $i$ can be obtained by:

$$AZLS_i(j) = -\frac{2j}{2}Dt^2 + (jk - B)t \quad ; \quad i = 1, \ldots, n \quad j = 1, \ldots, Z_1$$  \hspace{1cm} (9)

Since this section is located below the x-axis, we must multiply it by $-1$. Hence,

$$AZLS_i(j) = \frac{2j}{2}Dt^2 - (jk - B)t \quad ; \quad i = 1, \ldots, n \quad j = 1, \ldots, Z_1$$  \hspace{1cm} (10)

Furthermore, if $TAZLS_i$ denotes the total area of the trapezoids to the left side of the periods, then

$$TAZLS_i = \sum_{j=1}^{Z_1} \frac{2j}{2}Dt^2 - (jk - B)t \quad ; \quad i = 1, \ldots, n$$  \hspace{1cm} (11)

or

$$TAZLS_i = Z_1(Bt - \frac{Dt^2}{2}) + (Dt^2 - kt)Z_1 \frac{Z_1 + 1}{2} \quad ; \quad i = 1, 2, \ldots, n$$  \hspace{1cm} (12)

Figure (1) shows that the area of the second section is the sum of $Z_2 - Z_1$ similar triangles. Let $ATLS_i(j)$ denote the area of triangle $j$ of product $i$. Then, for the areas of triangle 1 and 2 we have

$$ATLS_i(1) = \frac{[-B + (k - Dt)]^2}{2D} \quad ; \quad i = 1, \ldots, n$$  \hspace{1cm} (13)
\[ ATLS_i(2) = \left[ \frac{-B + 2(k - Dt)}{2D} \right]^2 \quad ; \quad i = 1, \ldots, n \] (14)

So, in general, \( ATLS_i(j) \) can be obtained by

\[ ATLS_i(j) = \left[ \frac{-B + j(k - Dt)}{2D} \right]^2 \quad ; \quad i = 1, \ldots, n \] (15)

Moreover, if \( TATLS \) represents the total area of the triangles to the left of the periods, then

\[
TATLS_i = \sum_{j=Z_{i-1}+1}^{Z_i} \left[ \frac{-B + j(k - Dt)}{2D} \right]^2 \quad ; \quad i = 1, \ldots, n
\] (16)

\[
TATLS_i = \sum_{j=1}^{Z_i} \left[ \frac{-B + j(k - Dt)}{2D} \right]^2 - \sum_{j=1}^{Z_i} \left[ \frac{-B + j(k - Dt)}{2D} \right]^2 \quad ; \quad i = 1, \ldots, n
\] (17)

\[
TATLS_i = \frac{(Z_2 - Z_1)B^2}{2D} + \frac{(Z_2(Z_2+1)(2Z_2+1) - Z_1(Z_1+1)(2Z_1+1))(k - Dt)^2}{12D} + \frac{(Z_1(Z_1+1) - Z_2(Z_2+1))B(k - Dt)}{2D} \quad ; \quad i = 1, 2, \ldots, n
\] (18)

The third section of the backorder area is the area of the triangle to the right of \( T_{pi} \).

Let \( ATRS_i \) be the area of this triangle. Then, it can be calculated by

\[ ATRS_i = \frac{B^2}{2D} \] (19)

Finally, let the total backorder area of the \( i^{th} \) product be

\[ TBA_i = AZLS_i + ATLS_i + ATRS_i \quad ; \quad i = 1, 2, \ldots, n \] (20)

Then, the total annual backorder cost of the \( i^{th} \) product will be

\[ TB_i = \frac{D_i}{Q_i} b_i TBA_i \] (21)

which can be obtained by Equation (22).
Total holding cost ($TH_i$)

Employing the same approach used to calculate $TB_i$, in order to model $TH_i$, we split the inventory area into three sections. The first section is located above the interval $(2t, 5t)$ and is made up of $Z_2 - Z_1$ triangles. Let $AT_i(j)$ be the area of triangle $j$ of product $i$. Then, in general we have

$$AT_i(j) = \frac{[-B + j(k - Dt) + Dt]^2}{2D}; \quad i = 1, \ldots, n \tag{23}$$

If $TAT_i$ denotes the total area of the triangles for product $i$, then

$$TAT_i = \sum_{j=Z_i+1}^{Z_2} \frac{[-B + j(k - Dt) + Dt]^2}{2D}; \quad i = 1, \ldots, n \tag{24}$$

$$TAT_i = \frac{(Z_2 - Z_i)(Dt - B)^2}{2D} + \frac{(Z_2(Z_2 + 1)(2Z_2 + 1) - Z_i(Z_i + 1)(2Z_i + 1))(k - Dt)^2}{12D} + \frac{(Z_2(Z_2 + 1) - Z_i(Z_i + 1))(Dt - B)(k - Dt)}{2D}; \quad i = 1, 2, \ldots, n \tag{25}$$

The second section is located above the interval $(5t, 7t)$ and is made up of $m - 1 - Z_2$ trapezoids. Let $AZ_i$ represents the area of trapezoid $j$ of product $i$. Then, after simplifications the total trapezoidal area ($TAZ_i$) becomes

$$\frac{b_i(D_i - Z_i(B_i t_i - \frac{D_i t_i^2}{2}) + b_i(D_i t_i^2 - k t_i)Z_i Z_1 + 1}{2} +$$

$$b_i \left(\frac{Z_2 - Z_i}{2Q_i}\right) B_i^2 + b_i \left(\frac{Z_2(Z_2 + 1)(2Z_2 + 1) - Z_i(Z_i + 1)(2Z_i + 1))}{12Q_i}\right)(k_i - D_i t_i)^2 +$$

$$b_i \left(\frac{Z_i(Z_i + 1) - Z_2(Z_2 + 1)B_i(k_i - D_i t_i)}{2Q_i}\right) + b_i \frac{B_i^2}{2Q_i} \tag{22}$$
\[
TAZ_i = \sum_{j=Z_i+1}^{\max_{j}^1 - 1} - \frac{2j - 1}{2}Dt^2 + (jk - B)t \quad ; \quad i = 1, \ldots, n
\] (26)

\[
TAZ_i = \sum_{j=1}^{\max_{j}^1} - \frac{2j - 1}{2}Dt^2 + (jk - B)t - \sum_{j=1}^{\max_{j}^1} - \frac{2j - 1}{2}Dt^2 + (jk - B)t \quad ; \quad i = 1, \ldots, n
\] (27)

\[
TAZ_i = (m - 1 - Z_j)(\frac{Dt^2}{2} - Bt) + (kt - Dt^2)\left(m \frac{m - 1}{2} - Z_j \frac{Z_j + 1}{2}\right) \quad ; \quad i = 1, \ldots, n
\] (28)

The third section covers the triangle to the left of \(T_{p_i}\). Let \(ALT_i\) be the area of this triangle. Then it can be modeled as

\[
ALT_i = \frac{r^2}{2D} = \frac{\left(Q \left(1 - D/\rho\right) - B\right)^2}{2D} \quad ; \quad i = 1, \ldots, n
\] (29)

Finally, let the total holding cost area be

\[
THA_i = TAT_i + TAZ_i + ALT_i \quad ; \quad i = 1, \ldots, n
\] (30)

Then, the total annual holding cost of the \(i^{th}\) product will be

\[
TH_i = \frac{D}{Q_i} h_i THA_i \quad ; \quad i = 1, \ldots, n
\]

which can be obtained using Equation (31).

\[
TH_i = h_i \frac{(Z_2 - Z_1)(D_i t_i - B_i)^2}{2Q_i} + h_i \frac{(Z_2(Z_2+1)(2Z_2+1) - Z_1(Z_1+1)(2Z_1+1))(k_i - D_i t_i)^2}{12Q_i} + h_i \frac{(Z_2(Z_2+1) - Z_1(Z_1+1))(D_i t_i - B_i)(k_i - D_i t_i)}{2Q_i} + h_i \frac{D_i}{Q_i} (m_i - 1 - Z_2)\frac{(D_i t_i^2 - B_i)^2}{2} + h_i \frac{D_i}{Q_i} (k_i t_i - D_i t_i^2)\left(m_i \frac{m_i - 1}{2} - Z_2 \frac{Z_2 + 1}{2}\right) + h_i \frac{\left(Q_i \left(1 - D_i/\rho_i\right) - B_i\right)^2}{2Q_i}
\] (31)
3.3.6 Special cases

Before presenting the final inventory model, two special cases that may occur must be considered. Since $m_i, k_j$, and $B_i$ are the decision variables of the model, their optimal values are needed to be obtained. For any $k_i \in N$ and $B_i \in N$, we have $Z_1 \leq Z_2$. In case of $Z_2 < m_i - 1$ no change in the model is required. However, if $Z_2 \geq m_i - 1$ or $Z_1 \geq m_i - 1$, then Equation (1) and Equation (2) are not valid anymore. In these cases, we need to have $Z_1 = m_i - 1$ or $Z_2 = m_i - 1$. Therefore, Equations (1) and (2) are changed to (32) and (33), respectively.

$$Z_1 = \text{Min} \left( \frac{B_i - D_j t_j}{k_i - D_j t_j}, m_i - 1 \right)$$ \hspace{1cm} (32)

$$Z_2 = \text{Min} \left( \frac{B_i}{k_i - D_j t_j}, m_i - 1 \right)$$ \hspace{1cm} (33)

The second special case occurs when $Z_i \leq m_i - 1$. In this situation $I_{\text{max}} \leq 0$ and the model always faces stock-out. Figure (2) shows the inventory graph of this case.

Insert Figure (2) about here

In this situation, if Equation (22) is used to calculate the backordering cost, the area of the triangle ADE (see Figure 2) is computed twice; once in computation of the area of triangle ABC and once in the calculation of the area under line FG (see Figure 2). Thus, one of them needs to be eliminated. Let $S_d$ be the area of the triangle ADE. Then

$$S_d = \frac{I_{\text{max}}^2}{2D} = \left( \frac{Q \left( 1 - \frac{D}{P} \right) - B}{2D} \right)^2$$ \hspace{1cm} (34)
As we can see, Equation (34) is the same as Equation (29). Therefore, if \( I_{\text{max}} > 0 \) then we must multiply Equation (32) by \( h_i \) and add the result to the other costs. However, if \( I_{\text{max}} \leq 0 \) then we must multiply Equation (33) by \( b_i \) and subtract the result from the other costs. We note that in a specific problem one of either \( I_{\text{max}} > 0 \) or \( I_{\text{max}} \leq 0 \) can occur. Hence, if \( y_i \); \( i = 1, 2 \) denotes these two situations, then we have

\[
y_i = \begin{cases} 
  h_i \left( I_{\text{max}} \right)^2 & \text{if } I_{\text{max}} > 0 \\
  -b_i \left( I_{\text{max}} \right)^2 & \text{if } I_{\text{max}} \leq 0
\end{cases}
\]  

(35)

3.4. The total annual costs (TC)

Now, based on Equations (4), (5), (6), (22), (31) and (35), the total annual cost of all products is obtained using Equation (36).

\[
TC = \sum_{i=1}^{n} (TT_i + TP_i + TS_i + TB_i + TH_i)
\]

\[
= \sum_{i=1}^{n} \left( d_i \frac{D_i}{k_i} + c_i D_i + A_i \frac{D_i}{Q_i} + (b_i Z_1 - h_i (m_i - 1 - Z_2)) \frac{D_i}{Q_i} \left( B_i t_i - \frac{D_i t_i^2}{2} \right) + \right. \\
\left. \left( b_i Z_1 \frac{Z_1 +1}{2} - h_i \left( m_i \frac{m_i -1}{2} - Z_2 \frac{Z_2 +1}{2} \right) \right) \frac{D_i}{Q_i} \left( D_i t_i^2 - k_i t_i \right) + \right. \\
\left. \frac{(Z_2 - Z_1)}{2Q_i} \left( b_i B_i^2 + h_i (D_i t_i - B_i)^2 \right) + \right. \\
\left. (b_i + h_i) \left( Z_2(Z_2 +1)(2Z_2 +1) - Z_1(Z_1 +1)(2Z_1 +1) \right) (k_i - D_i t_i)^2 \right) + \right. \\
\left. \left( h_i D_i t_i - (b_i + h_i)B_i \right) \left( Z_2(Z_2 +1) - Z_1(Z_1 +1) \left( k_i - D_i t_i \right) + b_i \frac{B_i^2}{2Q_i} + y_i \right) \right)
\]  

(36)
3.5. Problem formulation

The objective of the model is to determine the optimum values of $Q, m, k$ and $B$ such that the total annual cost is minimized and the following constraints are satisfied:

1. The warehouse space to store the products is limited
2. The number of shipments must be within its boundaries

Hence, the problem can be formulated as

$$
\text{Min } TC = \sum_{i=1}^{n} \left\{ d_i \frac{D_i}{k_i} + c_i D_i + A \frac{D_i}{Q_i} + (b_i Z_i - h_i (m_i - 1 - Z_i)) \frac{D_i}{Q_i} \left( B_i t_i - \frac{D_i t_i^2}{2} \right) + 
\left( b_i Z_i + \frac{1}{2} - h_i \left( m_i - \frac{1}{2} - Z_i \frac{1}{2} \right) \right) \frac{D_i}{Q_i} (D_i t_i^2 - k_i t_i) + 
\frac{(Z_2 - Z_1)}{2Q_i} (b_i B_i^2 + h_i (D_i t_i - B_i)^2) + 
(b_i + h_i) \frac{(Z_2(Z_2 + 1)(2Z_2 + 1) - Z_i(Z_i + 1)(2Z_i + 1))(k_i - D_i t_i)^2}{12Q_i} + 
(h_i D_i t_i - (b_i + h_i)B_i) \frac{(Z_2(Z_2 + 1) - Z_i(Z_i + 1))(k_i - D_i t_i)}{2Q_i} + b_i \frac{B_i^2}{2Q_i} + y_i \right\}
$$

s.t.:

$$
\sum_{i=1}^{n} f_i Q_i \leq f
$$

$$
Q_i = m_i k_i , \quad i = 1, \ldots, n
$$

$$
Z_1 = \left| \frac{B_i - D_i t_i}{k_i - D_i t_i} \right| , \quad Z_2 = \left| \frac{B_i}{k_i - D_i t_i} \right|
$$

$$
y_i = \begin{cases} 
  h_i \frac{(I_{\text{max}})^2}{2D_i} & \text{if } I_{\text{max}} > 0 \\
  -b_i \frac{(I_{\text{max}})^2}{2D_i} & \text{if } I_{\text{max}} \leq 0 
\end{cases}
$$
\[ I_{\text{max}} = Q_i (1 - \frac{D_i}{p_j}) - B_i \]

\[ m_i, k_i, B_j, Q_i \text{ Integer} \]  \hspace{1cm} (37)

Since the model in (37) is a constrained nonlinear integer program, in the next section a genetic algorithm will be proposed to solve it.

4. Genetic Algorithms

Genetic algorithms (GAs) are stochastic search algorithms based on the mechanism of natural selection and natural genetics. The basic concepts of GAs were introduced by Holland in 1975 [21]. Since then growing interest can be observed on GAs due to their simplicity as algorithms and their ability to discover good solutions quickly for complex searching and optimization problems. They mimic the process of natural selection and are based on Darwin’s “survival of the fittest” principles. A GA normally starts with a set of potential solutions called population. Each Individual in the population (called chromosome) represents a possible solution to the problem at hand. 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, some chromosomes are selected from the current generation as parents to generate offsprings. Parents are combined using the crossover and mutation operators to produce offspring. The algorithm applies genetic operators until a stopping criterion is satisfied.

In the next subsections, we demonstrate the steps required to solve the model given in Equation (37) by a genetic algorithm.
4.1. Chromosome Representation

The first step of developing a GA is to encode the problem’s variables as a finite-length string called chromosome. Traditionally, chromosomes are a simple binary string. This simple representation is not well suited for combinatorial problems; therefore a chromosome consisting of integers is a solution in this paper. In the GA method, we select a two-dimensional structure to represent a solution. This matrix has three rows and \( n \) columns. The three elements of each column show the number of shipped pallets, the capacities of the pallets and the maximum shortage level for each product. In addition, the first row of the matrix shows the number of shipped pallets, the second row shows the capacities of pallets and the third one shows the maximum shortage level for all products. Figure (3) presents the general form of a chromosome.

Insert Figure (3) about here

4.2. Initial population

A GA requires a population of potential solutions of the given problem to be initialized. The initial population of individuals is randomly generated by a number of chromosomes (population size or \( \text{PopSize} \)). If \( X_c \) denotes a chromosome in the population, then it can be obtained by:

\[
X_c = \begin{bmatrix}
\alpha \times m_1^{\text{up}} & \alpha \times m_2^{\text{up}} & \ldots & \alpha \times m_n^{\text{up}} \\
\alpha \times k_1^{\text{up}} & \alpha \times k_2^{\text{up}} & \ldots & \alpha \times k_n^{\text{up}} \\
\alpha \times B_1^{\text{up}} & \alpha \times B_2^{\text{up}} & \ldots & \alpha \times B_n^{\text{up}}
\end{bmatrix} ; \quad c = 1, 2, \ldots, \text{PopSize}
\]

where \( \alpha \) is a random number generated uniformly from \([0,1]\) and \( m_i^{\text{up}}, k_i^{\text{up}}, B_i^{\text{up}} \) are the upper bounds of \( m_i, k_i, \) and \( B_i, \) respectively.
4.3. Constraint-handling and fitness evaluation

After getting a population of potential solutions, we need to evaluate how good they are. This evaluation is achieved through the computation of the cost associated to each chromosome, using the fitness function. Since the offspring produced by the GA operators are likely to be infeasible, the main problem in applying genetic algorithms to solve a constrained problem is how to deal with constraints. The most common approach in the GA community to handle constraint is to use penalties. The idea of this method is to transform a constrained optimization problem into an unconstrained one by adding or multiplying a certain value to/by the objective function based on the amount of constraint violation presented in a certain solution [22]. In this study, we use the additive form of the penalty function (\( Pen(S) \)) and the fitness function (\( fitn(S) \)) with the following form:

\[
fitn(S) = f(S) + Pen(S) \\
Pen(S) = 0 \quad \text{if } S \text{ is feasible} \\
Pen(S) > 0 \quad \text{otherwise}
\]  

(38)

where \( f(S) \) is the objective function in Equation (37) and \( S \) represents a solution.

4.4. Selection

In each generation, a set of offspring chromosomes will be generated through a recombination process. The selection operator is responsible for the choice of which individual, and how many copies of it, will be selected as parent chromosome. An individual is selected if it has a high fitness value, and the choice is biased towards the fittest members. In recent years, many selection methods have been compared and examined. Common types of them are as follows [23]:

- Roulette wheel selection.
- \((\mu + \lambda)\) - Selection.
• Tournament selection.
• Steady-state reproduction.
• Ranking and scaling.
• Sharing.

A “roulette wheel selection” procedure has been applied for the selection operator of this research. This selection approach is based on the concept of selection probability for each individual proportional to the fitness value. For individual $k$ with fitness $f_k$, its selection probability, $p_k$, is calculated as follows:

$$p_k = \frac{f_k}{\sum_{j=1}^{PopSize} f_j} \quad (39)$$

Then a biased roulette wheel is made according to these probabilities. The selection process is based on spinning the roulette wheel $PopSize$ times. The individuals selected from the selecting process are then stored in a mating pool. Moreover, in order to prevent losing the best-found solution, a simple elitist strategy is also used. It always copies the best chromosome of each generation to the next generation without any modification.

4.5. Crossover

The crossover operator is the basic operator of producing new chromosomes in a GA. It operates on two parent solutions with probability $P_c$ and generates offspring by recombining both parent solution features. In this operation, an expected $P_c \times PopSize$ number of chromosomes will take part. Let us assume that $X = (x^1_1, x^1_2, \ldots, x^1_n)$ and
are the two chromosomes that have been selected for the crossover operation. Then, the two produced offspring, \( Y_k = (y_1^k, y_2^k, \ldots, y_n^k), k = 1, 2, \) are

\[
y_i^k = \begin{cases} 
\alpha_i \left( x_i^1 + x_i^2 \right)/2 & \text{if a random digit is 0} \\
\alpha_i \left( x_i^1 + x_i^2 \right)/2 + (1 - \alpha_i) \left( x_i^1 + x_i^2 \right) & \text{if a random digit is 1}
\end{cases}
\]

where \( \alpha_i \) is a random number from \([0,1]\).

### 4.6. Mutation

Mutation is the second operation in the GA methods to explore new solutions that increase the population diversity. This operation consists of randomly altering the value of each element of the chromosome according to the mutation probability \( P_m \). It can avoid the answer becoming trapped in local optima and ensure the accessibility of the entire solution space. It is usually performed with low probability; otherwise it would defeat the order building being generated through selection and crossover. Here, we use the non-uniform mutation whose action is dependent on the age of the population. If \( X = (x_1, x_2, \ldots, x_n) \) is a chromosome and \( x_i \in [l_i, u_i] \) is a gene to be mutated, then the resulting gene \( x_i' \) is obtained by:

\[
x_i' = \begin{cases} 
x_i + \Delta(t, u_i - x_i) & \text{if a random digit is 0} \\
x_i - \Delta(t, x_i - l_i) & \text{if a random digit is 1}
\end{cases}
\]

where \( t \) is the current generation number and \( l_i \) and \( u_i \) are the lower and the upper bounds of \( x_i \), respectively. This operation returns a value in the range \([0, y]\) such that the probability of returning a number close to zero increases as \( t \) increases. This property causes the operator to
search uniformly in the initial stages (when \( t \) is small) and locally at the final stages. In our study, we have taken \( \Delta(k, y) \) as follows:

\[
\Delta(t, y) = y \left( 1 - r \left( \frac{t}{T} \right)^b \right)
\]

where \( r \) is a random number from [0,1], \( T \) is the maximum generation number, and \( b \) is a parameter provided by the user that determines the degree of non-uniformity.

### 4.7. Stopping criteria

The processes of generating new chromosomes and searching for better solutions are continued until a stopping criterion is satisfied. Different criteria used in literature are: (1) the process can be stopped after a fixed number of generations, or (2) when any significant improvement in the solution ceases to occur. In this paper, the proposed GA is run for a fixed number of generations.

### 5. Setting the parameters of the genetic algorithm

One important decision to make when implementing a genetic algorithm is how to set the parameters values. Modifying the parameter settings or the components of a genetic algorithm may greatly impact the algorithm’s performance [23,24]. Conventionally, setting GA parameters relies on a trial and-error procedure. However, this procedure cannot determine optimal GA parameter settings and consumes considerable time.

In this paper, we employed the Response Surface Methodology (RSM) to optimize GA parameters via systematic experiments. RSM is a collection of mathematical and statistical techniques that are useful for the modeling and analysis of problems in which a response of interest is influenced by several variables and the objective is to optimize this response [25].
In most of the RSM problems, the form of relationship between the response and the independent variable is unknown. Thus the first step in RSM is to find a suitable approximation for the true functional relationship between the response \( Y \) and set of independent variables. Usually a second order model is utilized in response surface methodology [25]. There are many designs available for fitting a second-order model. The most popular one is the central composite design (CCD). It consists of \( 2^k \) factorial points, \( n_c \) central points, and \( 2k \) axial points. In CCD, the model parameters are estimated using a second-order model of the form:

\[
Y = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{i=1}^{k} \beta_i^2 X_i^2 + \sum_{i \geq j} \sum_{i=1}^{k} \beta_{ij} X_i X_j
\] (40)

where \( Y \) is the expected value of the response variable, \( \beta_0, \beta_i, \beta_{ij} \) are the model parameters, \( X_i \) and \( X_j \) are the input variables that affect the response \( Y \), and \( k \) is the number of factors being studied.

In this research, the factors that affect the response are the population size \( (\text{PopSize}) \), the maximum number of generations \( (\text{MaxGen}) \), the crossover probability \( (P_c) \), the mutation probability \( (P_m) \) and one control factor: the problem size \( (\text{ProS}) \). The selected design factors each with three levels are listed in Table (1). The selected optimum parameters are the ones with the best fitness value obtained by GA.

| Insert Table (1) about here |

Since there are five factors, a fractional factorial design with \( 2^{5-1} \) factorial points, \( 2 \times 5 \) axial points and 7 central points requiring thirty three experiments is performed. The design matrix of the selected central composite design along with the experimental results is
shown in Table (2). The last column of Table (2) represents the best fitness value for each problem obtained in the last generation of GA.

5.1. Results

In order to evaluate the GA parameters three different problem instances of size 5, 10 and 15 products are generated. The data of these examples are given in Table (3). In order to fit the data by a regression model, the experimental data were analyzed with the aid of the Design-Expert software (Version 7.1.6, State Ease, MN, USA). Second-order coefficients were generated by regression with stepwise elimination. Response was first fitted to the factors by multiple regressions. The quality of fit was evaluated by the coefficients of determination \( R^2 \) and the analysis of variances (ANOVA). The insignificant coefficients were eliminated on the basis of their p-values. The regression coefficients, standard error, \( p \)-values, and determination coefficients \( R^2 \) are presented in Table (4).

The ANOVA for fitness is shown in Table (5). This analysis was carried out for a level of significance of 5%, that is, for a level of confidence of 95%. The \( R^2 \) value of 99.98% and the F-value for the regression was significant at a level of 5% \( (p < 0.05) \), while
the lack of fit was not significant at the 5% level \((p>0.05)\), indicating the good predictability of the model.

**5.2. Discussion**

The estimated regression function of the model fitness that is given in equation (41) not only shows that the most important GA parameters are \(P_r\), \(P_o\), and \(P_m\), but also indicates significant negative linear effects for \(P_r\), \(PopSize\), \(MaxGen\) and \(P_o\) on \(TC\). That is, the higher \(P_r\), \(PopSize\), \(MaxGen\) and \(P_o\), the lower the \(TC\) will be. Furthermore, the effect of \(P_m\) on \(TC\) is positive. Hence, the lower the mutation probability, the lower the total annual cost of all products will be.

\[
Fitness = 53091.54 - 4915.36ProS - 62.59PopSize - 27.52MaxGen - 13848.43P_r + 9424.27P_m + 653.96(ProS)^2 + 0.01(MaxGen)^2 + 336.19(ProS)(P_r) - 621.36(ProS)(P_m) + 0.04(PopSize)MaxGen + 37.48(PopSize)(P_r) + 7.89(MaxGen)(P_r) - 3.12(MaxGen)(P_m)
\]  

(41)

Since the most significant GA parameters are defined so far, the next step is to determine the best values of these parameters that lead to the best value of the fitness function. To do so, we must set the control factor (\(ProS\)) equal to the size of problem at hand and the levels of other factors must be set at the values given in Table (1). The optimized GA parameters for 5 examples with different problem sizes are shown in Table (6). The results in Table (6) indicate that as the problem becomes larger, a lower probability of crossover and a higher mutation probability are required to reach the best fitness.
In the next section a numerical example is given to demonstrate the applicability of the proposed parameter-tuned GA.

6. A numerical example

To demonstrate the application of the proposed methodology, a simple numerical example is coded by Microsoft Visual C# 2005 and is tested under the operation environment of Windows XP Professional with Pentium III 800MHz, 320MB of RAM.

Consider a multi-product inventory control problem with ten products and general data given in Table (3) (the first ten products). In this example, \( f = 25000 \) and the initial parameters of GA (PopSize, MaxGen, \( P_c \), and \( P_m \)) were set according to Table (6). The optimal solution of this problem, gained by the proposed algorithm, is as follows:

\[
\begin{bmatrix}
4 & 3 & 2 & 1 & 1 & 1 & 9 & 1 & 1 & 28 \\
80 & 48 & 149 & 200 & 334 & 232 & 26 & 64 & 157 & 149 \\
0 & 17 & 26 & 3 & 6 & 3 & 2 & 0 & 7 & 6
\end{bmatrix}, Fitness = 54681
\]

This solution indicates that for example for product two, the company needs to order 3 pallets, each with capacity of 48, in which 17 units of shortage is permitted. Furthermore, based on the fitness values, the graph of the convergence path is presented in Figure (4).
7. Conclusion

In this paper a multi-product EPQ model with limited warehouse space was presented in which the orders were delivered discretely in the form of multiple pallets. Moreover, it was assumed that the shortages were allowed and were completely backlogged. Under these conditions, the problem was formulated as a non-linear integer-programming model and a parameter-tuned genetic algorithm was proposed to solve it.

8. References

[1]. Harris FW. How many parts to make at once. *Factory, the Magazine of Management* 1913; **10**: 135–136.


[7]. Wee HM, Yu J, Chen MC. Optimal inventory model for items with imperfect quality and shortage backordering. *Omega* 2007; **35**: 7-11.


[21]. Holland JH. *Adoption in neural and artificial systems*. The University of Michigan Press 1975, Ann Arbor, Michigan, USA.


Figure (1). The inventory graph of the problem

Figure (2). Behavior of the inventory level when $I_{\text{max}} \leq 0$
\[
\begin{pmatrix}
  m_i \\
  k_i \\
  B_i
\end{pmatrix}
\begin{bmatrix}
  m_1 & m_2 & \ldots & m_n \\
  k_1 & k_2 & \ldots & k_n \\
  B_1 & B_2 & \ldots & B_n
\end{bmatrix}
\]

Figure (3). Structure of a chromosome

Figure (4). The graph of the convergence path
Table (1). The GA input parameter levels of the factorial design

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Min</th>
<th>Medium</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProS</td>
<td>5</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>PopSize</td>
<td>20</td>
<td>110</td>
<td>200</td>
</tr>
<tr>
<td>MaxGen</td>
<td>50</td>
<td>525</td>
<td>1000</td>
</tr>
<tr>
<td>$P_c$</td>
<td>0.1</td>
<td>0.55</td>
<td>1</td>
</tr>
<tr>
<td>$P_m$</td>
<td>0.1</td>
<td>0.55</td>
<td>1</td>
</tr>
</tbody>
</table>

Table (2). Design matrix of the central composite design

<table>
<thead>
<tr>
<th>Std. Order</th>
<th>Run Order</th>
<th>ProS</th>
<th>PopSize</th>
<th>MaxGen</th>
<th>$P_c$</th>
<th>$P_m$</th>
<th>Fitness</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>1</td>
<td>15</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>112752</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>15</td>
<td>200</td>
<td>50</td>
<td>1</td>
<td>0.1</td>
<td>115406</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>15</td>
<td>200</td>
<td>50</td>
<td>0.1</td>
<td>1</td>
<td>116198</td>
</tr>
<tr>
<td>29</td>
<td>4</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>57366</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>15</td>
<td>200</td>
<td>1000</td>
<td>1</td>
<td>1</td>
<td>114452</td>
</tr>
<tr>
<td>28</td>
<td>6</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>57307</td>
</tr>
<tr>
<td>31</td>
<td>7</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>57186</td>
</tr>
<tr>
<td>19</td>
<td>8</td>
<td>10</td>
<td>20</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>57732</td>
</tr>
<tr>
<td>23</td>
<td>9</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.1</td>
<td>0.55</td>
<td>56848</td>
</tr>
<tr>
<td>24</td>
<td>10</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>1</td>
<td>0.55</td>
<td>56069</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>5</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>33301</td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>5</td>
<td>200</td>
<td>1000</td>
<td>1</td>
<td>0.1</td>
<td>33313</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>15</td>
<td>200</td>
<td>1000</td>
<td>0.1</td>
<td>0.1</td>
<td>111537</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>10</td>
<td>200</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>55063</td>
</tr>
<tr>
<td>27</td>
<td>15</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>56574</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>5</td>
<td>200</td>
<td>50</td>
<td>0.1</td>
<td>0.1</td>
<td>34389</td>
</tr>
<tr>
<td>5</td>
<td>17</td>
<td>5</td>
<td>20</td>
<td>1000</td>
<td>0.1</td>
<td>0.1</td>
<td>33304</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>5</td>
<td>20</td>
<td>50</td>
<td>1</td>
<td>0.1</td>
<td>34574</td>
</tr>
<tr>
<td>1</td>
<td>19</td>
<td>5</td>
<td>20</td>
<td>50</td>
<td>0.1</td>
<td>1</td>
<td>50709</td>
</tr>
<tr>
<td>14</td>
<td>20</td>
<td>15</td>
<td>20</td>
<td>1000</td>
<td>1</td>
<td>0.1</td>
<td>114271</td>
</tr>
<tr>
<td>32</td>
<td>21</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>56704</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>15</td>
<td>20</td>
<td>50</td>
<td>0.1</td>
<td>0.1</td>
<td>126141</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>10</td>
<td>110</td>
<td>1000</td>
<td>0.55</td>
<td>0.55</td>
<td>56835</td>
</tr>
<tr>
<td>25</td>
<td>24</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.1</td>
<td>57032</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
<td>5</td>
<td>200</td>
<td>1000</td>
<td>0.1</td>
<td>1</td>
<td>33302</td>
</tr>
<tr>
<td>10</td>
<td>26</td>
<td>15</td>
<td>20</td>
<td>50</td>
<td>1</td>
<td>1</td>
<td>119563</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>1</td>
<td>56916</td>
</tr>
<tr>
<td>30</td>
<td>28</td>
<td>10</td>
<td>110</td>
<td>525</td>
<td>0.55</td>
<td>0.55</td>
<td>56540</td>
</tr>
<tr>
<td>11</td>
<td>29</td>
<td>5</td>
<td>200</td>
<td>50</td>
<td>1</td>
<td>1</td>
<td>35799</td>
</tr>
<tr>
<td>21</td>
<td>30</td>
<td>10</td>
<td>110</td>
<td>50</td>
<td>0.55</td>
<td>0.55</td>
<td>62609</td>
</tr>
<tr>
<td>6</td>
<td>31</td>
<td>15</td>
<td>20</td>
<td>1000</td>
<td>0.1</td>
<td>1</td>
<td>112367</td>
</tr>
<tr>
<td>13</td>
<td>32</td>
<td>5</td>
<td>20</td>
<td>1000</td>
<td>1</td>
<td>1</td>
<td>33364</td>
</tr>
</tbody>
</table>
Table (3). Data of the examples

<table>
<thead>
<tr>
<th>Product</th>
<th>$D_i$</th>
<th>$p_i$</th>
<th>$c_i$</th>
<th>$A_i$</th>
<th>$d_i$</th>
<th>$h_i$</th>
<th>$f_i$</th>
<th>$b_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>190</td>
<td>30</td>
<td>50</td>
<td>45</td>
<td>2</td>
<td>3</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>150</td>
<td>170</td>
<td>17</td>
<td>80</td>
<td>15</td>
<td>4</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>240</td>
<td>320</td>
<td>25</td>
<td>45</td>
<td>40</td>
<td>1</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>240</td>
<td>270</td>
<td>22</td>
<td>38</td>
<td>27</td>
<td>5</td>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>360</td>
<td>450</td>
<td>35</td>
<td>55</td>
<td>21</td>
<td>2</td>
<td>1</td>
<td>49</td>
</tr>
<tr>
<td>6</td>
<td>200</td>
<td>250</td>
<td>37</td>
<td>50</td>
<td>35</td>
<td>10</td>
<td>2</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>120</td>
<td>42</td>
<td>50</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>120</td>
<td>170</td>
<td>28</td>
<td>72</td>
<td>10</td>
<td>2</td>
<td>4</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>250</td>
<td>350</td>
<td>17</td>
<td>60</td>
<td>10</td>
<td>5</td>
<td>7</td>
<td>38</td>
</tr>
<tr>
<td>10</td>
<td>400</td>
<td>420</td>
<td>10</td>
<td>57</td>
<td>25</td>
<td>1</td>
<td>8</td>
<td>46</td>
</tr>
<tr>
<td>11</td>
<td>260</td>
<td>360</td>
<td>32</td>
<td>40</td>
<td>28</td>
<td>3</td>
<td>4</td>
<td>19</td>
</tr>
<tr>
<td>12</td>
<td>320</td>
<td>450</td>
<td>45</td>
<td>45</td>
<td>10</td>
<td>4</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>370</td>
<td>500</td>
<td>50</td>
<td>55</td>
<td>17</td>
<td>7</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>14</td>
<td>290</td>
<td>320</td>
<td>37</td>
<td>30</td>
<td>28</td>
<td>6</td>
<td>9</td>
<td>36</td>
</tr>
<tr>
<td>15</td>
<td>160</td>
<td>190</td>
<td>13</td>
<td>30</td>
<td>30</td>
<td>3</td>
<td>8</td>
<td>22</td>
</tr>
</tbody>
</table>

Table (4). Multiple regression analysis for fitness

<table>
<thead>
<tr>
<th>Term</th>
<th>Coefficient</th>
<th>SE Coef</th>
<th>t-value</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>53091.54</td>
<td>1136.29</td>
<td>46.724</td>
<td>0.000</td>
</tr>
<tr>
<td>Main (linear) effects</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProS</td>
<td>-4915.36</td>
<td>239.63</td>
<td>-20.513</td>
<td>0.000</td>
</tr>
<tr>
<td>PopSize</td>
<td>-62.59</td>
<td>2.88</td>
<td>-21.758</td>
<td>0.000</td>
</tr>
<tr>
<td>MaxGen</td>
<td>-27.52</td>
<td>1.51</td>
<td>-18.198</td>
<td>0.000</td>
</tr>
<tr>
<td>Pc</td>
<td>-13848.43</td>
<td>835.7</td>
<td>-16.571</td>
<td>0.000</td>
</tr>
<tr>
<td>Pm</td>
<td>9424.27</td>
<td>749.13</td>
<td>12.58</td>
<td>0.000</td>
</tr>
<tr>
<td>Squared effects</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProS×ProS</td>
<td>653.96</td>
<td>11.68</td>
<td>56.007</td>
<td>0.000</td>
</tr>
<tr>
<td>MaxGen×MaxGen</td>
<td>0.01</td>
<td>0</td>
<td>10.43</td>
<td>0.000</td>
</tr>
<tr>
<td>Interaction effects</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProS×Pc</td>
<td>336.19</td>
<td>60.61</td>
<td>5.547</td>
<td>0.000</td>
</tr>
<tr>
<td>ProS×Pm</td>
<td>-621.36</td>
<td>60.61</td>
<td>-10.252</td>
<td>0.000</td>
</tr>
<tr>
<td>PopSize×MaxGen</td>
<td>0.04</td>
<td>0</td>
<td>13.058</td>
<td>0.000</td>
</tr>
<tr>
<td>PopSize×Pc</td>
<td>37.48</td>
<td>3.37</td>
<td>11.133</td>
<td>0.000</td>
</tr>
<tr>
<td>MaxGen×Pc</td>
<td>7.89</td>
<td>0.64</td>
<td>12.367</td>
<td>0.000</td>
</tr>
<tr>
<td>MaxGen×Pm</td>
<td>-3.12</td>
<td>0.64</td>
<td>-4.903</td>
<td>0.000</td>
</tr>
</tbody>
</table>

S = 545.499  PRESS = 21149803  R-Sq = 99.98%  R-Sq(pred) = 99.93%  R-Sq(adj) = 99.97%
Table (5). Analysis of variance for fitness

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Seq SS</th>
<th>Adj SS</th>
<th>Adj MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>13</td>
<td>32033884613</td>
<td>3203884613</td>
<td>246144970</td>
<td>8280.91</td>
<td>0.000</td>
</tr>
<tr>
<td>Linear</td>
<td>5</td>
<td>29090394427</td>
<td>890773399</td>
<td>178154680</td>
<td>598.7</td>
<td>0.000</td>
</tr>
<tr>
<td>Square</td>
<td>2</td>
<td>2762770529</td>
<td>2762770529</td>
<td>138135265</td>
<td>4642.3</td>
<td>0.000</td>
</tr>
<tr>
<td>Interaction</td>
<td>6</td>
<td>180719656</td>
<td>180719656</td>
<td>30119943</td>
<td>101.22</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>18</td>
<td>5356251</td>
<td>5356251</td>
<td>297570</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lack-of-Fit</td>
<td>13</td>
<td>4630146</td>
<td>4630146</td>
<td>356165</td>
<td>2.45</td>
<td>0.165</td>
</tr>
<tr>
<td>Pure Error</td>
<td>5</td>
<td>726105</td>
<td>726105</td>
<td>145221</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>31</td>
<td>32039240864</td>
<td>32039240864</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table (6). Optimal values of GA factors

<table>
<thead>
<tr>
<th>Example</th>
<th>ProS</th>
<th>PopSize</th>
<th>MaxGen</th>
<th>P_c</th>
<th>P_m</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>189</td>
<td>946</td>
<td>0.95</td>
<td>0.12</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>193</td>
<td>751</td>
<td>0.95</td>
<td>0.13</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>175</td>
<td>469</td>
<td>0.81</td>
<td>0.18</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>189</td>
<td>660</td>
<td>0.77</td>
<td>0.34</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>170</td>
<td>842</td>
<td>0.51</td>
<td>0.96</td>
</tr>
</tbody>
</table>