International Journal of the Physical Sciences Vol. 7(2), pp. 262 - 272, 9 January, 2012
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.1592

Full Length Research Paper

Optimizing a multi-product and multi-supplier the economic production quantity model using genetic algorithm

Goran Ćirović[1], Dragan Pamučar[2]*, Dragoljub Sekulović[2] and Boban Đorović[2]

[1]The Belgrade University College of Civil Engineering and Geodesy, 11000 Belgrade, Serbia.
[2]Military Academy, University of Defence, Pavla Jurisica Sturma 33, 11000 Belgrade, Serbia.

Accepted 12 December, 2011

In order to make the economic production quantity (EPQ) model more applicable to real-world production and inventory control problems, in this paper, we expand this model by assuming that some imperfect items of different product types are being produced such that reworks are allowed. In addition, we may have more than one product and supplier along with warehouse space and budget limitation. We show that the model of the problem is a constrained non-linear integer program and propose a genetic algorithm (GA) 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.

Key words: Genetic algorithm, optimizing, design of experiments, imperfect items.

INTRODUCTION

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 (Harris, 1913). Regardless of the simplicity of EOQ and EPQ, they are still applied in industry-wide today (Jamal et al., 2004).

Traditional EPQ models assume that a production process always produces parts with perfect quality. However, process failures are a fact of any workplace. It is more realistic to assume that production is sometimes imperfect. Such a production process is called imperfect production (Salameh and Jaber, 2000). Rosenblatt and Lee (1986) investigated the influence of process deterioration on optimal EPQ.

A multiproduct single-machine system on EPQ problem in which the production defective rates of all items are random variables and all defective items are assumed to be scrapped (rework is not allowed) is considered. Besides, the productions of all items are performed on a single machine such that there is a limited capacity, and that shortages are allowed and are considered to be partially backordered. Furthermore, the service level is another constraint of the system (Taleizadeh et al., 2010). Goyal et al. (2003) developed a simple approach for determining an optimal integrated vendor-buyer inventory policy for an item with imperfect quality.

Hou and Lin (2004) studied the effect of an imperfect production process on the optimal production run length when capital investment in process quality improvement is adopted.

Leung (2007) proposed an EPQ model with a flexible and imperfect production process. He formulated this inventory decision problem using geometric programming. Recently, Hou (2007) considered an EPQ model with imperfect production processes, in which the setup cost and process quality are functions of capital expenditure.

Specifically, Hsu and Yu (2009) investigate an EOQ model with imperfective items under a one-time-only sale, where the defectives can be screened out by a 100% screening process and then can be sold in a single batch by the end of the 100% screening process. In real manufacturing environments, producing defective items is inevitable. These defective items must be rejected,

*Corresponding author. E-mail: dpamucar@gmail.com. Tel: +38164 23 77 908, +38111 360 31 35.

repaired, reworked, or, if they have reached the customer, refunded. In all cases, substantial costs are incurred. Therefore, it is more appropriate to take the quality related cost into account in determining the optimal ordering policy. In the literature, Porteus (1986) initially studied the effect of process deterioration on the optimal production cycle time. Tapiero (1987) links optimal quality inspection policies and the resulting improvements in the manufacturing cost.

With the assumption that imperfect items are removed from inventory, the production cycles are no longer identical (rather renewable points are defined at the beginning of every cycle), and the expected cost depends on both the mean and variance (second moment) of the amount of imperfect quality items in a lot. One contribution of this paper is deriving an expression for the variance of the number of imperfect quality items resulting from a two-state Markov process and incorporating this in the expected cost function (Maddah et al., 2010).

Wahab and Jaber (2010) present models based on Salameh and Jaber (2000), Maddah and Jaber (2008) and Jaber et al. (2008) with different holding costs for the good items and defective items.

An EPQ model is considered, where demand of the item is fuzzy random in nature with known probability distribution and the production process is assumed to be not 100% perfect, that is, a fraction of the produced items are defective. Further, it is assumed that the defective items are sold at a reduced price and the selling price of fresh units is taken as a mark up over the unit production cost. The model is formulated to maximize the expected average profit. Since demand is fuzzy random in nature, expected profit is a fuzzy number (Bag et al., 2009).

In practice, as production defective items are inevitable, reworking process is often done. Gopalan and Kannan (1995) wrote: ‘‘All over the world, industries are concentrating in making quality an inherent in their products. In spite of these efforts, rework is becoming an unavoidable factor in many production systems. For example, glass manufacturing, food processing, etc.’’

An integrated EPQ model that incorporates EPQ and maintenance programs is presented. This model considers the impact of restoration action such as imperfect repair; rework and preventive maintenance (PM) on the damage of a deteriorating production system (Liao et al., 2009). Jaber et al. (2009) investigate production processes that generate defects requiring rework. It does this by using an EOQ model with entropy costs. Chiu et al. (2007) extend the prior works (Chung, 1997; Chiu, 2003) and study the optimal run time problem of EPQ model with scrap, reworking of defective items and stochastic breakdowns. Since little attention was taken into the area of investigating joint effects of the aforementioned practical situations on optimal production run time, this paper intends to serve this purpose.

In recent years, several researchers have applied genetic algorithms (GAs) as an optimization technique to solve the production/inventory problems. For example. Rezaei and Davoodi (2008) introduce imperfect items and storage capacity in the lot sizing with supplier selection problem and formulate the problem as a mixed integer programming (MIP) model. Then the model is solved with a GA. There are several interesting and relevant papers related to the application of GA in inventory problems such as Stockton and Quinn (1993), Mondal and Maiti (2002), Hou et al. (2007), Gupta et al. (2009), Lotfi (2006), Pal et al. (2009) and Taleizadeh et al. (2009a, 2009b).

PROBLEM DEFINITION

Consider a production company that works with more suppliers to produce more products. All of the produced items are inspected to be classified as perfect, imperfect (defective but repairable) and scrap (defective and not repairable) products. The situations by which the company and the supplier interact with each other are defined as follows:

1) Required time of the inspection task is zero.
2) All imperfect products are reworked to be perfect and the scrap products are sold with the reduced price.
3) The work in process inventory (WIP) consists of three types of materials around the manufacturing machines: raw material, perfect products and imperfect products.
4) The budget and warehouse space of the company for all products is limited.
5) Shortage and delay are not allowed.
6) All parameters such as the demand rate, the rate of imperfect and scrap items production, the setup cost, etc. are all known and deterministic.
7) Transportation cost is fraction of raw material cost per unit.

The objective is to determine the order quantity of the products for each supplier that the total inventory cost is minimized while the constraints are satisfied. It means what products order, in what quantities and with which suppliers.

PROBLEM MODELING

In order to mathematically formulate the problem at hand, we take advantage of the classical EPQ model and extend it to the problem to contain the perfect, the imperfect and the scrap items along with the warehouse and budget capacity. In order to model the problem, firstly, we define the parameters and the variables. Secondly, we pictorially demonstrate the situation by inventory graphs. Thirdly, we derive different costs. Finally, we present the model of the problem.

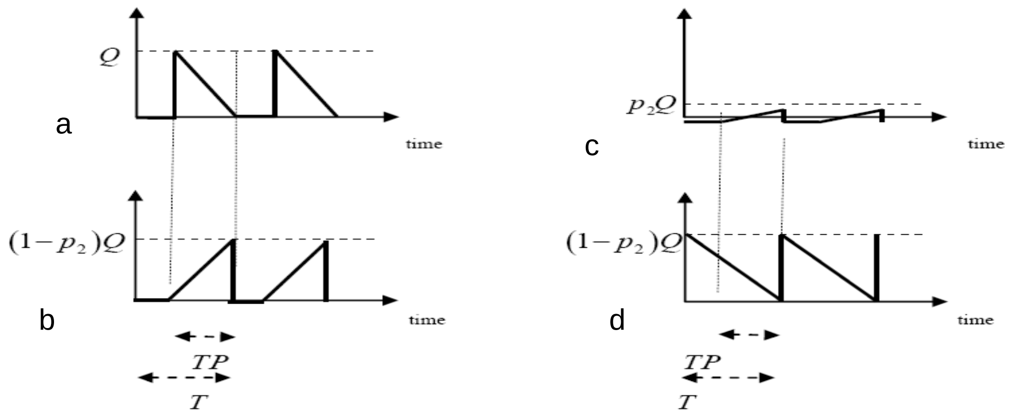

Figure 1. (a), Inventory graphs of row material; (b), perfect products; (c), scrap items; (d), the product in the warehouse.

Variables and parameters

For products $j = 1,..., m$ and suppliers $i = 1,..., n$, we define the variables and the parameters of the model as follows:

n, Number of suppliers; *m*, number of products; Q_{ij}, order quantity; P_{ij}, production rate; D_{ij}, demand rate; A_{ij}, setup cost per cycle; h_{ij}, holding cost rate; M_{ij}, the cost of raw material per unit; S_{ij}, setup time; m_{ij}, machining time per unit; R_{ij}, production cost rate per unit time; c_{ij}, average production cost per unit; v_{ij}, average value added per unit; $\overline{W}_{ij}$, average investment per unit of WIP; $\overline{I}_{ij}$, average amount of warehouse inventory; p_{1ij}, imperfect production percentage; p_{2ij}, scrap production percentage; s_{1ij}, perfect production cost; s_{2ij}, scrap production cost; T_{ij}, cycle time; TP_{ij}, total time per cycle to produce; t_{ij}, average production time per unit; *α*, fractional of raw material cost for transportation cost; f_{ij}, required space per perfect unit; *F*, total available warehouse space for all products; C_{ij}, providence cost per unit; *X*, maximum capital; TC_{Pij}, total procurement cost; TC_{Oij}, total set up cost; TC_{Iij}, total inspection cost; TC_{Tij}, total transportation cost; TC_{WIPij}, total holding cost for WIP; TC_{Hij}, total holding cost for perfect products; *TC*, total annual cost of all products.

Inventory graph

In order to calculate all inventory costs, it is necessary to survey the WIP and warehouse inventory. For the problem at hand, the graph of raw material quantity versus time is demonstrated in Figure 1a. In addition, the graphs of the perfect and scrap WIP inventory versus time are illustrated in Figures 1b and c, respectively. In this problem, the rate of demand is constant and hence graph of the final product quantity in the warehouse is similar to the EOQ model and is given in Figure 1d.

Costs calculations

The total annual cost of all products (*TC*) is the sum of total procurement cost ($\sum_{i=1}^{n}\sum_{j=1}^{m} TC_{Pij}$), total set up cost ($\sum_{i=1}^{n}\sum_{j=1}^{m} TC_{Oij}$), total inspection cost ($\sum_{i=1}^{n}\sum_{j=1}^{m} TC_{Iij}$), total transportation cost ($\sum_{i=1}^{n}\sum_{j=1}^{m} TC_{Tij}$), total holding cost for WIP inventory ($\sum_{i=1}^{n}\sum_{j=1}^{m} TC_{WIPij}$) and the total holding cost for warehouse inventory ($\sum_{i=1}^{n}\sum_{j=1}^{m} TC_{Hij}$) for all products. Therefore,

$$TC = \sum_{i=1}^{n}\sum_{j=1}^{m}(TC_{Pij} + TC_{Oij} + TC_{Iij} + TC_{Tij} + TC_{WIPij} + TC_{Hij}) \quad (1)$$

In any cycle, since the set up time, the production time and the reworking time are equal to S_{ij}, $m_{ij}Q_{ij}$ and $m_{ij}(p_{1ij}Q_{ij})$, respectively, the total time to produce product *j*, (TP_{ij}), is given in (2).

$$TP_{ij} = S_{ij} + m_{ij}Q_{ij} + m_{ij}(p_{1ij}Q_{ij}) = S_{ij} + m_{ij}Q_{ij}(1 + p_{1ij}) \quad (2)$$

Hence, the average production time for each unit of product *j* is:

$$t_{ij} = \frac{TP_{ij}}{Q_{ij}} = \frac{S_{ij}}{Q_{ij}} + m_{ij}(1 + p_{1ij}) \quad (3)$$

Based on R_{ij} which is the rate of production cost per unit time, v_{ij} and c_{ij} are obtained as:

$$v_{ij} = R_{ij}t_{ij} = R_{ij}(\frac{S_{ij}}{Q_{ij}} + m_{ij}(1 + p_{1ij})) \quad (4)$$

$$c_{ij} = M_{ij} + v_{ij} = M_{ij} + R_{ij}\left(\frac{S_{ij}}{Q_{ij}} + m_{ij}(1 + p_{1ij})\right) \quad (5)$$

Since delays are not allowed, the supply and the demand quantities are equal and we have:

$$(1 - p_{2ij})Q_{ij} = D_{ij}T_{ij} \Rightarrow T_{ij} = \frac{(1 - p_{2ij})Q_{ij}}{D_{ij}} \quad (6)$$

As s_{1ij} and s_{2ij} represent the price of the perfect and the scrap items, respectively, the average revenue in unit time is obtained as:

$$TR_{ij} = \frac{(1 - p_{2ij})Q_{ij}s_{1ij} + p_{2ij}Q_{ij}s_{2ij}}{T_{ij}} = D_{ij}s_{1ij} + \frac{p_{2ij}}{1 - p_{2ij}}D_{ij}s_{2ij} \quad (7)$$

Note that for the problem at hand the revenue in unit time does not depend on the lot size.

Based on (2) to (6), the inventory costs of (1) are calculated as follows:

Total procurement cost (TC_{Pij})

Since the annual rate of demand for each product is known, the total procurement cost for product *j* per unit time is obtained as:

$$TC_{Pij} = \frac{m_{ij}Q_{ij}}{T_{ij}} = \frac{m_{ij}D_{ij}}{(1 - p_{2ij})} \quad (8)$$

Total setup cost (TC_{Oij})

For each product, the setup process accrues only once and hence the set up cost per unit time of the j^{th} product can be obtained as:

$$TC_{Oij} = \frac{A_{ij}}{T_{ij}} = \frac{A_{ij}D_{ij}}{Q_{ij}(1 - p_{2ij})} \quad (9)$$

Total inspection cost (TC_{Iij})

Assuming 100% inspection and that all of the imperfect products transform to perfect ones after reworks, the inspection of each product occurs once and its associated cost per unit time is obtained as:

$$TC_{Iij} = \frac{I_{ij}Q_{ij}}{T_{ij}} = \frac{I_{ij}D_{ij}}{1 - p_{2ij}} \quad (10)$$

Total transportation cost (TC_{Tij})

Here, we assume the transportation cost related to fractional of raw material cost. Thus:

$$TC_{Tij} = \alpha(1 - p_{2ij})M_{ij}Q_{ij} \quad (11)$$

Total holding cost for work in process (WIP) (TC_{WIPij})

In order to calculate the holding cost of WIP inventory of the j^{th} product, since $\bar{W}_{ij}$ denotes the average investment per unit of WIP inventory (including raw materials, perfect and imperfect items) and h_{ij} represents the holding cost rate of the j^{th} product, then:

$$TC_{WIPij} = h_{ij}\bar{W}_{ij} \quad (12)$$

The average raw material inventory of each product is the total amount of raw materials (the surface under its corresponding inventory graph) divided by the cycle time. Accordingly, the average investment value of the raw material is obtained by the product of the average raw material inventory and the price per unit of the raw material. The average investment value of the perfect and imperfect products can be calculated similarly. Hence, the average investment value per unit of the WIP inventory of product *j* is given in (13).

$$\bar{W}_{ij} = \frac{\frac{1}{2}Q_{ij}TP_{ij}}{T_{ij}}M_{ij} + \frac{\frac{1}{2}(1 - p_{2ij})Q_{ij}TP_{ij}}{T_{ij}}c_{ij} + \frac{\frac{1}{2}p_{2ij}Q_{ij}TP_{ij}}{T_{ij}}c_{ij} = \frac{1}{2}\frac{Q_{ij}TP_{ij}}{T_{ij}}(M_{ij} + c_{ij}) = \frac{D_{ij}}{2(1 - p_{2ij})}[S_{ij} + m_{ij}(1 + p_{1ij})Q_{ij}][2M_{ij} + \frac{R_{ij}S_{ij}}{Q_{ij}} + R_{ij}m_{ij}(1 + p_{1ij})] \quad (13)$$

Hence, based on (12) and (13), the average holding cost of the WIP inventory of product *j* is:

$$TC_{WIPij} = \frac{h_{ij}D_{ij}}{2(1 - p_{2ij})}[S_{ij} + m_{ij}(1 + p_{1ij})Q_{ij}][2M_{ij} + \frac{R_{ij}S_{ij}}{Q_{ij}} + R_{ij}m_{ij}(1 + p_{1ij})] \quad (14)$$

Total holding cost for perfect products (TC_{Hij})

In order to calculate the holding cost of the warehouse inventory, we first need to estimate the average warehouse inventory. Figure 1d, we have:

$$\bar{I}_{ij} = \frac{\frac{1}{2}Q_{ij}(1 - p_{2ij})T_{ij}}{T_{ij}} = \frac{1}{2}Q_{ij}(1 - p_{2ij}) \quad (15)$$

Hence, using (5) and (15), the holding cost of the warehouse inventory for product *j* becomes:

$$TC_{Hij}=h_{ij}c_{ij}\bar{I}_{ij}=\frac{1}{2}h_{ij}\{M_{ij}+R_{ij}[\frac{S_{ij}}{Q_{ij}}+m_{ij}(1+p_{1ij})]\}Q_{ij}(1-p_{2ij}) \quad (16)$$

Finally, the total annual inventory cost of all products described in (1) is given in (17).

$$TC=\sum_{i=1}^{n}\sum_{j=1}^{m}(TC_{Pij}+TC_{Oij}+TC_{Iij}+TC_{Tij}+TC_{WIPij}+TC_{Hij})=$$
$$\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\begin{array}{l}\frac{m_{ij}D_{ij}}{(1-p_{2ij})}+\frac{A_jD_{ij}}{Q_{ij}(1-p_{2ij})}+\frac{I_{ij}D_{ij}}{(1-p_{2ij})}+\alpha(1-p_{2ij})M_{ij}Q_{ij}+\\ \frac{h_{ij}D_{ij}}{2(1-p_{2ij})}\left[S_{ij}+m_{ij}(1+p_{1ij})Q_{ij}\right]\left[2M_{ij}+\frac{R_{ij}S_{ij}}{Q_{ij}}+R_{ij}m_{ij}(1+p_{1ij})\right]+\\ h_{ij}\{M_{ij}+R_{ij}\left[\frac{S_{ij}}{Q_{ij}}+m_{ij}(1+p_{1ij})\right]\}\frac{1}{2}Q_{ij}(1-p_{2ij})\end{array}\right] \quad (17)$$

Problem formulation

The objective of the model is to determine the optimum value of Q_{ij} for each supplier 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) Budget limitation

Hence, the problem can be formulated as:

$$Min\ TC=\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\begin{array}{l}\frac{m_{ij}D_{ij}}{(1-p_{2ij})}+\frac{A_jD_{ij}}{Q_{ij}(1-p_{2ij})}+\frac{I_{ij}D_{ij}}{(1-p_{2ij})}+\alpha(1-p_{2ij})M_{ij}Q_{ij}+\\ \frac{h_{ij}D_{ij}}{2(1-p_{2ij})}\left[S_{ij}+m_{ij}(1+p_{1ij})Q_{ij}\right]\left[2M_{ij}+\frac{R_{ij}S_{ij}}{Q_{ij}}+R_{ij}m_{ij}(1+p_{1ij})\right]+\\ h_{ij}\{M_{ij}+R_{ij}\left[\frac{S_{ij}}{Q_{ij}}+m_{ij}(1+p_{1ij})\right]\}\frac{1}{2}Q_{ij}(1-p_{2ij})\end{array}\right]$$

such that:

$$\sum_{j=1}^{m}\sum_{i=1}^{n}(1-p_{2ij})f_{ij}Q_{ij}\le F$$

$$\sum_{j=1}^{m}\sum_{i=1}^{n}(1-p_{2ij})C_{ij}Q_{ij}\le X$$

$$Q_{ij}\succ 0;\ j=1...m, i=1...n \quad (18)$$

Since the model in (18) is a constrained non-linear integer program, in the study a GA will be proposed to solve it.

$$\begin{bmatrix} 15 & 10 & 67 & 20 & 38 \\ 12 & 7 & 32 & 45 & 50 \end{bmatrix}$$

Figure 2. Structure of a chromosome.

GENETIC ALGORITHM

GAs mimic the evolutionary process by implementing a “survival of the fittest’’ strategy. GAs are probabilistic search and optimization techniques guided by the principles of evolution and natural genetics. This method provides global near-optimal solutions of an objective or fitness function by striking a remarkable balance between exploration and exploitation in complex, large, and multi-modal landscapes.

A more complete discussion of GAs including extensions to the general algorithm and related topics can be found in books by Davis (1991), Goldberg (1989), Holland (1975) and Michalewicz (1994).

In each evolution step of a standard GA, a new population is created from the preceding one using the selection, crossover and mutation operators that are explained below.

In the study, we demonstrate the steps required to solve the model given in (18) by a GA.

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 *n* rows and *m* columns. The elements of each column show the number of products. In addition, the elements of each row show the number of suppliers. For example, when we have 2 suppliers and 5 products, the chromosome matrix is 2×5. Figure 2 presents a typical form of a chromosome.

Initialization of the population

For any GA, it is necessary to initialize the population. The most common method is to randomly generate solutions for the entire population. Since GAs iteratively improves existing solutions the beginning population can be seeded by the decision maker with individuals from other algorithms or from an existing solution to the problem. The remainder of the population is then seeded with randomly generated solutions.

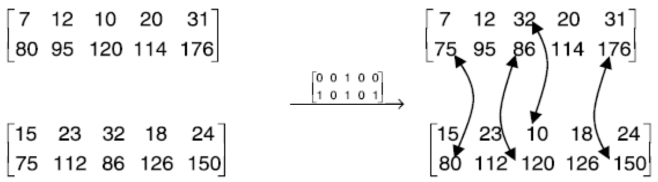

Figure 3. An example of a crossover operation.

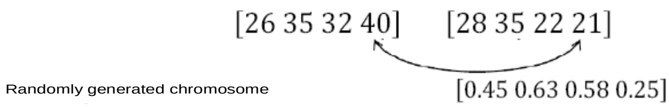

Figure 4. A graphical representation of the mutation operator.

In this research, the initial population is randomly generated regarding the population sizes that vary between 1 and maximum order.

Constraint-handling and fitness evaluation

The fitness value is a measure of the goodness of a solution with respect to the original objective and the ‘‘amount of infeasibility’’. The fitness function is formed by adding a penalty to the original objective function in MIP model (Homaifar et al., 1994; Michalewicz and Schoenauer, 1996).

Since finding a feasible solution (which will satisfy all constraints in MIP model) is a major problem and has difficulty in finding even one feasible solution, we use penalty approach to decrease fitness of infeasible solutions and toward the feasible region.

In this study, we use the additive form of the penalty function (*Pen(S)*) and the fitness function (*fitn(S)*) with the following form:

$$fitn(S) = f(S) + Pen(S)$$
$$Pen(S) = 0 \quad if\ S\ is\ feasible \qquad (19)$$
$$Pen(S) > 0 \quad otherwise$$

Where, *f(S)* is the objective function in (18) and *S* represents a solution. In this approach, we search for the solution that minimizes *fitn(S)*.

Selection operator

The selection of parents to produce successive generations plays an extremely important role in the GA. The goal is to allow the fittest individuals to be selected more often to reproduce. However, all individuals in the population have a chance of being selected to reproduce the next generation. Each individual is assigned a probability of being selected with better individuals having larger probabilities. There are several schemes for determining and assigning the selection probability, for example, roulette wheel selection, tournament selection and its extensions scaling techniques and ranking methods (Goldberg, 1989; Michalewicz, 1994).

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:

$$p_k = \frac{f_k}{\sum_{k=1}^{PopSize} f_k} \qquad (20)$$

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 in which the best chromosome of each generation is always copied to the next generation without any modification. This selection approach causes the algorithm to converge faster.

Genetic operators

There are two types of operators involved in the GA proposed: mutation and crossover.

In this research, we use single point crossover with different values of the P_c parameter ranging between 0.45 and 0.85. We note that an infeasible chromosome that does not satisfy the constraints of the models in (18) does not move to the new population. Figure 3 demonstrates the crossover operation.

In this research, different values between 0.05 and 0.35 are chosen as different values of P_m. We note that an infeasible chromosome that does not satisfy the constraints of the models in (18) does not move to the new population. Figure 4 shows an example of the mutation operator for four products in which P_m is chosen to be 0.35.

Stopping criteria

The last step in the methodology is to check if the method has found a solution that is good enough to meet the user’s expectations. Stopping criteria is a set of conditions such that when the method satisfies them, a good solution is obtained. In this paper, the proposed GA is run for a fixed number of generations.

Table 1. The GA input parameter levels of the factorial design.

Parameter	Min	Max
Max Gen	100	500
N	20	60
P_c	0.45	0.85
P_m	0.05	0.35

Table 2. Data of the example.

Product	D_1	D_2	A_1	A_2	M_1	M_2	S_1	S_2	m_1	m_2	p_{11}	p_{12}	p_{21}	p_{22}	R_1	R_2	h_1	h_2	I_1	I_2	f_1	f_2	C_1	C_2
1	20	22	21	28	8	10	0.017	0.02	0.01	0.03	0.24	0.1	0.05	0.1	15	10	0.1	0.38	15	6	15	9	55	36
2	22	22	26	18	9	5	0.01	0.014	0.14	0.38	0.15	0.25	0.1	0.02	14	15	0.5	0.10	12	5	18	15	35	26
3	19	28	23	31	5	7	0.02	0.018	0.15	0.2	0.25	0.09	0.01	0.06	13	9	0.98	0.28	14	7	13	32	40	29
4	30	19	31	29	9	6	0.04	0.01	0.1	0.1	0.15	0.26	0.06	0.09	12	11	0.52	0.3	12	10	16	8	60	23
5	21	20	18	21	7	4	0.013	0.011	0.18	0.19	0.1	0.19	0.06	0.12	15	9	0.24	0.16	7	12	10	16	45	28
6	27	24	29	25	11	9	0.01	0.01	0.01	0.25	0.12	0.16	0.09	0.05	12	13	0.35	0.28	9	10	8	14	65	34
7	30	21	26	22	8	10	0.011	0.024	0.1	0.23	0.21	0.15	0.05	0.07	14	8	0.16	0.24	26	10	21	10	80	26
8	20	20	21	26	8	8	0.01	0.032	0.11	0.16	0.26	0.13	0.05	0.045	15	12	0.25	0.32	12	14	14	8	35	27
9	20	17	28	24	10	9	0.01	0.04	0.05	0.25	0.18	0.21	0.08	0.065	12	9	0.42	0.18	11	4	16	15	50	25
10	24	18	27	15	5	8	0.05	0.098	0.11	0.25	0.16	0.14	0.11	0.08	11	11	0.25	0.2	13	8	13	17	75	24

SETTING THE PARAMETERS OF THE GENETIC ALGORITHM

Response surface methodology (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. Central composite design (CCD) is a response surface method that allows one to keep the size and complexity of the design low and simultaneously obtain some protection against curvature as described by Montgomery (2001).

One important decision to make when implementing a GA is how to set the parameters values. In order to satisfy this condition, a CCD is selected. Since there are four factors, a fractional factorial design with 2^4 factorial points, 2×4 axial points and seven central points, requiring 31 experiments are required.

In this research, the factors that affect the response are the population size (*PopSize*), the maximum number of generations (*MaxGen*), the crossover probability (P_c) and the mutation probability (P_m).

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.

Genetic algorithm parameters results

In order to evaluate the GA parameters, an example with parameters (F=10000, X=150000, α=0.1, n=2 and m=10) is presented. The data of this example is given in Table 2.

The design matrix of the selected CCD along with the experimental results is shown in Table 2. The PtType column of Table 3 represents the type of the design points (‘‘-1” for the axial points, ‘‘0” for the central points and ‘‘1” for the factorial points). The last column of Table 3 represents the best fitness value for each problem obtained in the last generation of GA. In order to fit the data to a regression model, an independent analysis for

Table 3. Design matrix of the central composite design.

Std. order	Run order	PtType	Blocks	*PopSize*(*N*)	*MaxGen*	P_c	P_m	Fitness
21	1	-1	1	40	300	0.100	0.550	29286
4	2	1	1	50	400	0.325	0.325	29184
24	3	-1	1	40	300	0.550	1.000	29105
30	4	0	1	40	300	0.550	0.550	29141
15	5	1	1	30	400	0.775	0.775	29048
31	6	0	1	40	300	0.550	0.550	29080
2	7	1	1	50	200	0.325	0.325	29524
12	8	1	1	50	400	0.325	0.775	29032
20	9	-1	1	40	500	0.550	0.550	29117
7	10	1	1	30	400	0.775	0.325	29328
18	11	-1	1	60	300	0.550	0.550	29120
16	12	1	1	50	400	0.775	0.775	29092
14	13	1	1	50	200	0.775	0.775	29251
9	14	1	1	30	200	0.325	0.775	29147
22	15	-1	1	40	300	1.000	0.550	29217
25	16	0	1	40	300	0.550	0.550	29227
13	17	1	1	30	200	0.775	0.775	29174
10	18	1	1	50	200	0.325	0.775	29267
11	19	1	1	30	400	0.325	0.775	29090
19	20	-1	1	40	100	0.550	0.550	29749
26	21	0	1	40	300	0.550	0.550	29088
5	22	1	1	30	200	0.775	0.325	29731
3	23	1	1	30	400	0.325	0.325	29242
29	24	0	1	40	300	0.550	0.550	29240
1	25	1	1	30	200	0.325	0.325	29700
6	26	1	1	50	200	0.775	0.325	29225
23	27	-1	1	40	300	0.550	0.100	29536
28	28	0	1	40	300	0.550	0.550	29145
17	29	-1	1	20	300	0.550	0.550	29406
27	30	0	1	40	300	0.550	0.550	29244
8	31	1	1	50	400	0.775	0.325	29164

response, fitness, is required. Second-order coefficients were generated by regression with stepwise elimination. The first step is to identify significant factors in the model, for that purposes a multiple regression analysis and an analysis of variance (ANOVA) are developed for both responses. The ANOVA is a partition of the total variability into its component parts. The regression coefficients, standard error, *p*-values and coefficient of determination (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 89.76% 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. It should be noted that the high value of R^2 is due to the fact that all sources of randomness in a GA (the population size, the maximum number of generations, the crossover probability and the mutation probability) are taken into account and that a second-order model was used to model the performance.

Discussion

Based on the results of Table 4, the estimated regression function is:

$$\begin{aligned} FitnessFunction &= 29166.7 - 53.9N - 129.3MaxGen - 13.1P_c \\ &- 119.1P_m + 15N^2 + 57.6MaxGen^2 + 12.2P_c^2 + 29.4P_m^2 \\ &+ 15.5N \times MaxGen - 23.6N \times P_c + 67.9N \times P_m \\ &+ 21.3MaxGen \times P_c + 42.7MaxGen \times P_m + 14.4P_c \times P_m \end{aligned} \quad (21)$$

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

Table 4. Multiple regression analysis for fitness.

Term	Coefficient	SE Coef	*t*-Value	*p*-Value
Constant	29166.7	32.77	890.136	0.000
Main (linear) effects				
PopSize	-53.9	17.70	-3.045	0.008
MaxGen	-129.3	17.70	-7.306	0.000
P_c	-13.1	17.70	-0.743	0.469
P_m	-119.1	17.70	-6.731	0.000
Squared effects				
PopSize×*PopSize*	15.0	16.21	0.928	0.367
MaxGen×*MaxGen*	57.6	16.21	3.556	0.003
$P_c \times P_c$	12.2	16.21	0.755	0.461
$P_m \times P_m$	29.4	16.21	1.816	0.088
Interaction effects				
PopSize×*MaxGen*	15.5	21.67	0.714	0.486
PopSize×P_c	-23.6	21.67	-1.087	0.293
PopSize×P_m	67.9	21.67	3.134	0.006
MaxGen×P_c	21.3	21.67	0.982	0.341
MaxGen×P_m	42.7	21.67	1.971	0.066
$P_c \times P_m$	14.4	21.67	0.663	0.517
S = 86.6923	*PRESS* = 561011	*R-Sq* = 89.76%		

Table 5. Analysis of variance for fitness.

Source	DF	Seq SS	Adj SS	Adj MS	*F*	*P*
Regression	14	1054135	1054135	75295	10.02	0.000
Linear	4	815512	815512	203878	27.13	0.000
Square	4	112362	112362	28091	3.74	0.025
Interaction	6	126261	126261	21044	2.80	0.047
Residual error	16	120249	120249	7516		
Lack-of-fit	10	90327	90327	9033	1.81	0.241
Pure error	6	29922	29922	4987		
Total	30	1174384				

function. The estimated regression function (the objective function) that needs to be minimized along with the constraints within the GA parameter ranges is solved by Lingo software. Table 6 shows the optimum results.

In the next section a numerical example is given to demonstrate the applicability of the proposed parameter-tuned GA.

A NUMERICAL EXAMPLE

Consider a multi-product inventory control model with ten products and general data are given in Table 2. In this example, F=10000, X=150000, α=0.1 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:

$$\begin{bmatrix} 44 & 55 & 40 & 55 & 48 & 52 & 56 & 47 & 53 & 60 \\ 55 & 44 & 58 & 43 & 51 & 46 & 42 & 52 & 45 & 38 \end{bmatrix}$$

Fitness, 29122

This solution indicates that for example, order quantity for product five by supplier two is 51 or order quantity for product ten by supplier one is 60. Furthermore, based on the fitness values, the graph of the convergence path is presented in Figure 5.

CONCLUSIONS

In this paper, a multi-product and multi-supplier EPQ model with limited warehouse space and budget were presented in which defective items and reworking are considered. Under these conditions, the problem was

Table 6. The Lingo optimum solution.

N*	***MaxGen****	**P_c***	**P_m***	**Cost**
60	500	0.85	0.35	14896990

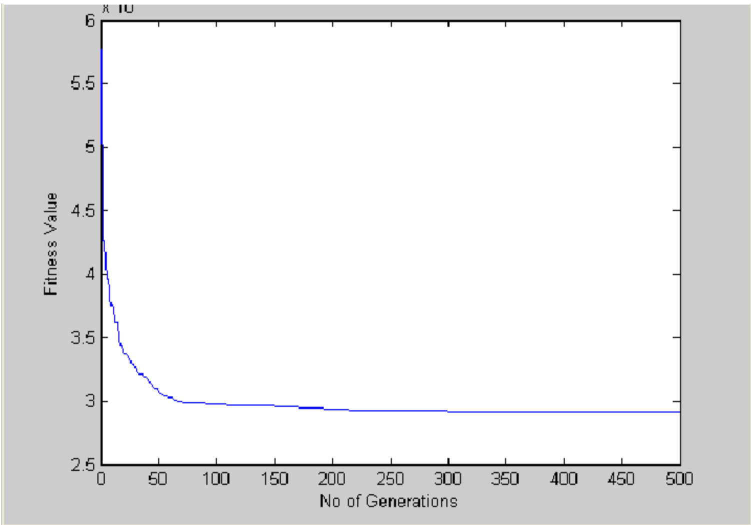

Figure 5. The graph of the convergence path.

formulated as a non-linear integer-programming model and a parameter tuned GA was proposed to solve it. At the end, a numerical example was presented to demonstrate the application of proposed methodology. In this example, the optimum values of the GA parameters were obtained using RSM.

REFERENCES

Bag S, Chakraborty D, Roy AR (2009). A production inventory model with fuzzy random demand and with flexibility and reliability considerations. Comput. Ind. Eng., 56: 411-416.

Chiu SW, Wang SL, Chiu YSP (2007). Determining the optimal run time for EPQ model with scrap, rework, and stochastic breakdowns. Eur. J. Oper. Res., 180: 664-676.

Chiu YP (2003). Determining the optimal lot size for the finite production model with random defective rate, the rework process, and backlogging. Eng. Optimiz., 35 (4): 427-437.

Chung KJ (1997). Bounds for production lot sizing with machine breakdowns. Comput. Ind. Eng., 32: 139-144.

Davis L (1991). *Handbook of genetic algorithms*. New York. Van Nostrand Reinhold.

Goldberg DE (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley. Reading. MA, pp. 159-161.

Gopalan MN, Kannan S (1995). Expected number analysis of a two-server queuing network subject to inner-stage inspection and rework. Comput. Oper. Res., 22: 935–946.

Goyal SK, Huang CK, Chen KC (2003). A simple integrated production policy of an imperfect item for vendor and buyer. Prod. Plan. Control., 14(7): 596–602.

Gupta RK, Bhunia AK, Goyal SK (2009). An application of genetic algorithm in solving an inventory model with advanced payment and interval valued inventory costs. Math. Comput. Model., 49:893–905.

Harris FW (1913). How many parts tomake at once. Factory Mag. Manage., 10: 135-136.

Holland JH (1975). *Adoption in neural and artificial systems.* Ann Arbor, Michigan (USA). The University of Michigan Press, pp. 351-355.

Homaifar A, Lai SHV, Qi X (1994). Constrained optimization via genetic algorithms. Simulation. 62(4): 242–254.

Hou CI, Lo CY, Leu JH (2007). Use genetic algorithm in production and inventory strategy. *In: Proceedings of the 2007 IEEE international conference on industrial engineering and engineering management (IEEM) conference.* Article number 4419335: 963–972.

Hou KL (2007). An EPQ model with setup cost and process quality as functions of capital expenditure. Appl. Math. Model., 31: 10-17.

Hou KL, Lin LC (2004). Optimal production run length and capital investment with an imperfect production processes. J. Comput. Sys. Sci. Int., 35: 133-137.

Hsu WKK, Yu HF (2009). EOQ model for imperfective items under a one-time only discount, Omega-Int. J. Manage. S., 37: 1018-1026.

Jaber MY, Bonney M, Moualek I (2009). An economic order quantity model for an imperfect production process with entropy cost, Int. J. Prod. Econ., 118: 26-33.

Jaber MY, Goyal SK, Imran M (2008). Economic production quantity for items with imperfect quality subjected to learning effects. Int. J. Prod. Econ., 115: 143-150.

Jamal AMM, Sarker BR, Mondal S (2004). Optimal manufacturing batch size with rework process at a single-stage production system. Comput. Indus. Eng., 47: 77–89.

Leung KNF (2007). A generalized geometric-programming solution to An economic production quantity model with flexibility and reliability considerations. Eur. J. Oper. Res., 176 (1): 240--251.

Liao GL, Chen YH, Sheu SH (2009). Optimal Economic Production Quantity Policy for Imperfect Process with Imperfect Repair and Maintenance. Eur. J. Oper. Res., 195: 348–357.

Lotfi G (2006). Applying genetic algorithms to dynamic lot sizing with batch ordering. Comput. Indust. Eng., 51: 433–444.

Maddah B, Jaber MY (2008). Economic order quantity for items with imperfect quality: Revisited. Int. J.Prod. Econ., 112: 808–815.

Maddah B, Moussawi L, Jaber MY (2010). Lot sizing with a Markov production process and imperfect items scrapped. Int. J. Prod. Econ., 124: 340–347.

Michalewicz Z (1994). *Genetic Algorithms + Data Structures = Evolution Programs*. AI Series, Springer, New York, pp. 264-266.

Michalewicz Z, Schoenauer M (1996). Evolutionary algorithms for constrained parameter optimization problems. Evol. Comput. J., 4 (1): 1-32.

Mondal S, Maiti M (2002). Multi-item fuzzy EOQ models using genetic algorithm. Comput. Indust. Eng., 44:105–117.

Montgomery DC (2001). *Design and analysis of experiments*. New York (USA). John Wiley and Sons, pp. 311-315.

Pal S, Maiti MK, Maiti M (2009). An EPQ model with price discounted promotional demand in an imprecise planning horizon via genetic algorithm. Comput. Indust. Eng., 57:181-187.

Porteus EL (1986). Optimal lot sizing process quality improvement and setup cost reduction. Oper. Res., 34: 137-144.

Rezaei J, Davoodi M (2008). A deterministic, multi-item inventory model with supplier selection and imperfect quality. Appl. Math. Model., 32: 2106–2116.

Rosenblatt MJ, Lee HL (1986). Economic production cycles with imperfect production process, IIE Trans., 18: 48–55.

Salameh MK, Jaber MY (2000). Economic production quantity model for items with imperfect quality. Int. J.Prod. Econ., 64: 59–64.

Stockton DJ, Quinn L (1993). Identifying economic order quantities using genetic algorithms. Int. J. Oper. Prod. Manage., 13: 92-103.

Taleizadeh AA, Niaki STA, Aryanezhad MB (2009a) A hybrid method of pareto, topsis and genetic algorithm to optimize multiproduct multi-constraint inventory systems with random fuzzy replenishment. J. Math. Comput. Model., 49:1044–1057.

Taleizadeh AA, Niaki STA, Hosseini V (2009b). Optimizing multiproduct multi-constraint bi-objective newsboy problem with discount by a hybrid method of goal programming and genetic algorithm. J. Eng. Optim., 41:437-457.

Taleizadeh AA, Niaki STA, Najafi AA (2010). Multiproduct single machine production system with stochastic scrapped production rate, partial backordering and service level constraint. J. Comput. Appl. Math., 233: 1834-1849.

Tapiero C (1987). Production learning and quality control. IIE Trans., 19: 362-370.

Wahab MIM, Jaber MY (2010). Economic order quantity model for items with imperfect quality, different holding costs, and learning effects: A note. Comput. Indust. Eng., 58: 186–190.