

NAUČNI ČLANCI SCIENTIFIC PAPERS

OPTIMIZING MODELS FOR PRODUCTION AND INVENTORY CONTROL USING A GENETIC ALGORITHM

Pamučar S. *Dragan*, Đorović D. *Boban*, University of Defence,
Military Academy, Deanery, Belgrade

FIELD: Mathematics (Operations Research)

ARTICLE TYPE: Original Scientific Paper

Summary:

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 being produced such as 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 to solve it. Moreover, a design of experiments is employed to calibrate the parameters of the algorithm for different problem sizes. In the end, a numerical example is presented to demonstrate the application of the proposed methodology.

Key words: Genetic algorithm, economic production quantity, optimizing.

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 [1]. Regardless of the simplicity of EOQ and EPQ, they are still applied industry-wide today [2].

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 [3]. Rosenblatt and Lee investigated the influence of process deterioration on optimal EPQ [4].

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 [5]. Goyal et al developed a simple approach for determining an optimal integrated vendor-buyer inventory policy for an item with imperfect quality [6].

Hou and Lin studied the effect of an imperfect production process on the optimal production run length when capital investment in process quality improvement is adopted [7].

Leung proposed an EPQ model with a flexible and imperfect production process [8]. He formulated this inventory decision problem using geometric programming. Recently, Hou considered an EPQ model with imperfect production processes, in which the setup cost and process quality are functions of capital expenditure [9,10].

Specifically, Hsu and Yu investigate an EOQ model with imperfect 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 [11].

In real manufacturing environments, producing defective items is inevitable. These defective items must be rejected, 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 initially studied the effect of process deterioration on the optimal production cycle time [12]. Tapiero links optimal quality inspection policies and the resulting improvements in the manufacturing cost [13].

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 [14].

Wahab and Jaber present models [15] based on [3], [16] and [17] with different holding costs for 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, i.e. 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 [18].

The optimization of choice by the application of fuzzy logic in practice has been applied many times [19, 20, 21, 22]

In practice, as the production of defective items is inevitable, reworking process is often done. Gopalan and Kannan wrote [23]: "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 as in 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 PM on the damage of a deteriorating production system [24]. Jaber et al investigate the production processes that generate defects requiring rework [25]. It does this by using an EOQ model with entropy costs. Chiu et al [26] extend the prior works [27, 28] 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 introduce imperfect items and storage capacity in the lot sizing with a supplier selection problem and formulate the problem as a mixed integer programming model [29]. Then the model is solved with a Genetic Algorithm. There are several interesting and relevant papers related to the application of GA in inventory problems such as [30, 31, 9, 10, 32, 33, 34].

Pasandideh et al developed a multi-product EPQ model with limited warehouse space. In addition, they assumed that the rate of imperfect and scrap items is known, and reworking makes the imperfect products perfect. Under these conditions, they formulated the problem as a non-linear integer-programming model and proposed a genetic algorithm to solve it [35].

In this paper, we extend the Pasandideh model assuming a multi-supplier. Also we consider the transportation cost and budget limitation in our model. Other conditions and constraints are the same as in 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, a design of experiments (DOE) is used to analyze the performance of the proposed GA. This analysis is important to improve the performance of the proposed GA by identifying the optimal values of its control parameters. Section 6 includes the results of applying the proposed genetic algorithm to a numerical example. The conclusion and recommendations for future research are given in Section 7.

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:

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

The objective is to determine the order quantity of the products for each supplier while minimizing the total inventory cost and keeping the constraints satisfied. It means to determine products ordered in certain quantities through certain 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 perfect, imperfect and scrap items along with the warehouse and budget capacity. In order to model the problem, first we define the parameters and the variables. Then we demonstrate the situation by inventory graphs and different costs and finally we present the model of the problem.

Variables and parameters

For products $j = 1, \dots, m$ and suppliers $i = 1, \dots, n$, 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,
\bar{W}_{ij}	Average investment per unit of WIP,
\bar{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	Fraction 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 work in process and warehouse inventory. For the problem at hand, the graph of raw material quantity versus time is demonstrated in Fig. 1. In addition, the graphs of the perfect and scrap WIP inventory versus time are illustrated in Figs. 2 and 3, respectively. In this problem, the rate of demand is constant and hence a graph of the final product quantity in the warehouse is similar to the EOQ model and is given in Fig. 4.

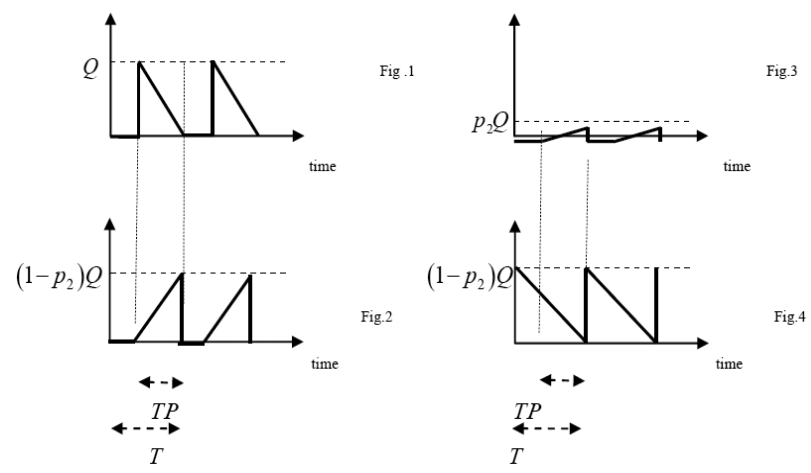


Figure 1-4. Inventory graphs of row material (Figure 1), perfect products (Figure 2), scrap items (Figure 3) and the product in the warehouse (Figure 4)

Costs calculations

The total annual cost of all products (TC) is the sum of total procurement cost (TC_{Pij}), total set up cost (TC_{Oij}), total inspection cost (TC_{Iij}), total transportation cost (TC_{Tij}), total holding cost for WIP inventory (TC_{PWij}) and the total holding cost for warehouse inventory (TC_{Hij}) for all products. In other words, we have:

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

In any cycle, since the set up time, the production time and the re-working time are equal to S_{ij} , $m_{ij}Q_{ij}$ and $m_{ij}(p_{1ij}Q_{ij})$, respectively, the total time to produce the 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 the 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:

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

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

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.

Now, 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 the 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 a fraction of the raw material cost. Thus

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

Total holding cost for WIP (TC_{WIPij})

In order to calculate the holding cost of the WIP inventory of the j^{th} product, since \bar{W}_{ij} denotes the average investment per unit of the 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 perfect and imperfect products can be calculated similarly. Hence, the average investment value per unit of the WIP inventory of the product j is given in (13).

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

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

$$TC_{WIPj} = \frac{h_j D_{ij}}{2(1-p_{2ij})} [S_{ij} + m_j(1+p_{1ij})Q_{ij}] \left[2M_{ij} + \frac{R_j S_{ij}}{Q_{ij}} + R_j m_j(1+p_{1ij}) \right] \quad (14)$$

Total holding cost for perfect products (TCH_{ij})

In order to calculate the holding cost of the warehouse inventory, we first need to estimate the average warehouse inventory. Regarding Fig. 4, 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 the product j becomes

$$TC_{Hij} = h_j c_{ij} \bar{I}_{ij} = \frac{1}{2} h_j \left\{ M_{ij} + R_j \left[\frac{S_{ij}}{Q_{ij}} + m_j(1+p_{1ij}) \right] \right\} Q_{ij} (1-p_{2ij}) \quad (16)$$

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

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

$$= \sum_{i=1}^n \sum_{j=1}^m \left[\begin{aligned} & \frac{m_j D_{ij}}{(1-p_{2ij})} + \frac{A_j D_{ij}}{Q_{ij}(1-p_{2ij})} + \frac{I_j D_{ij}}{(1-p_{2ij})} + \alpha(1-p_{2ij})M_{ij}Q_{ij} \\ & + \frac{h_j D_{ij}}{2(1-p_{2ij})} [S_{ij} + m_j(1+p_{1ij})Q_{ij}] \\ & \times \left[2M_{ij} + \frac{R_j S_{ij}}{Q_{ij}} + R_j m_j(1+p_{1ij}) \right] \\ & + h_j \left\{ M_{ij} + R_j \left[\frac{S_{ij}}{Q_{ij}} + m_j(1+p_{1ij}) \right] \right\} \frac{1}{2} Q_{ij} (1-p_{2ij}) \end{aligned} \right] \quad (17)$$

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 and
2. Budget limitation.

Hence, the problem can be formulated as:

$$MinTC = \sum_{i=1}^n \sum_{j=1}^m \left[\begin{aligned} & \frac{m_{ij}D_{ij}}{(1-p_{2ij})} + \frac{A_{ij}D_{ij}}{Q_{ij}(1-p_{2ij})} + \frac{I_{ij}D_{ij}}{(1-p_{2ij})} \\ & + \alpha(1-p_{2ij})M_{ij}Q_{ij} + \frac{h_{ij}D_{ij}}{2(1-p_{2ij})} [S_{ij} + m_{ij}(1+p_{1ij})Q_{ij}] \\ & \times \left[2M_{ij} + \frac{R_{ij}S_{ij}}{Q_{ij}} + R_{ij}m_{ij}(1+p_{1ij}) \right] + h_{ij} \\ & \times \{M_{ij} + R_{ij} \left[\frac{S_{ij}}{Q_{ij}} + m_{ij}(1+p_{1ij}) \right] \} \frac{1}{2} Q_{ij}(1-p_{2ij}) \end{aligned} \right] \\ st.:$$

$$\sum_{j=1}^m \sum_{i=1}^n (1-p_{2ij})f_{ij}Q_{ij} \leq F \quad (18)$$

$$\sum_{j=1}^m \sum_{i=1}^n (1-p_{2ij})C_{ij}Q_{ij} \leq X \quad (19)$$

$$Q_{ij} > 0; j = 1...m, i = 1...n \quad (20)$$

Since the model in (18), (19) and (20) is a constrained non-linear integer program, in the next section a genetic algorithm will be proposed to solve it.

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 [36], Goldberg [37], Holland [38] and Michalewicz [39].

In each evolution step of a standard genetic algorithm, a new population is created from the preceding one using the selection, crossover and mutation operators that explain these in the following.

In the next subsections, we demonstrate the steps required to solve the model given in (18), (19) and (20) by a genetic algorithm.

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.

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

Figure 5. Structure of a chromosome

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 . Fig. 5 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.

In this research, the initial population is randomly generated regarding the population sizes that vary between 1 and a 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 the MIP model [40, 41].

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

$$\begin{aligned} fitn(S) &= f(S) + Pen(S) \\ Pen(S) &= 0 \quad \text{if } S \text{ is feasible} \\ Pen(S) &> 0 \quad \text{otherwise} \end{aligned} \quad (21)$$

Where, $f(S)$ is the objective function in (18), (19) and (20) 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, e.g. roulette wheel selection, tournament selection and its extensions scaling techniques and ranking methods [37] and [39].

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:

$$p_k = \frac{f_k}{\sum_{k=1}^{PopSize} f_k} \quad (22)$$

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 genetic algorithm proposed: mutation and crossover.

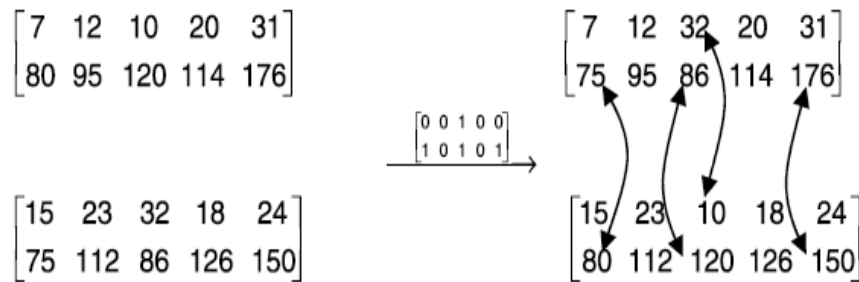


Figure 6. An example of a crossover operation

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 (18), (19) and (20) does not move to the new population. Fig. 6 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), (19) and (20) does not move to the new population. Fig. 7 shows an example of the mutation operator for four products in which P_m is chosen to be 0.35.

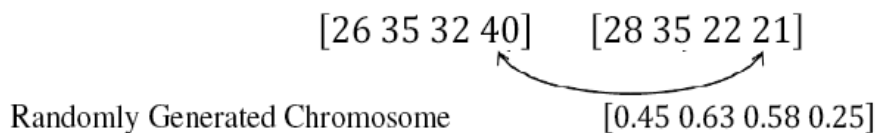


Figure 7. A graphical representation of the mutation operator

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.

Setting the Parameters of the Genetic Algorithm

The response surface methodology 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. The 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 [42].

One important decision to make when implementing a genetic algorithm is how to set the parameters values.

Table 1

The GA input parameter levels of the factorial design

<i>Parameters</i>	<i>Min</i>	<i>Max</i>
<i>Max Gen</i>	100	500
<i>N</i>	20	60
<i>P_c</i>	0.45	0.85
<i>P_m</i>	0.05	0.35

In order to satisfy this condition a central composite design 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 thirty one 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.

GA parameters results

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

Data of the example

Table 2

Pro-duct	D₁	D₂	A₁	A₂	M₁	M₂	S₁	S₂	m₁	m₂	p₁₁	p₁₂
1.	20	22	21	28	8	10	0.017	0.02	0.01	0.03	0.24	0.1
2.	22	22	26	18	9	5	0.01	0.014	0.14	0.38	0.15	0.25
3.	19	28	23	31	5	7	0.02	0.018	0.15	0.2	0.25	0.09
4.	30	19	31	29	9	6	0.04	0.01	0.1	0.1	0.15	0.26
5.	21	20	18	21	7	4	0.013	0.011	0.18	0.19	0.1	0.19
6.	27	24	29	25	11	9	0.01	0.01	0.01	0.25	0.12	0.16
7.	30	21	26	22	8	10	0.011	0.024	0.1	0.23	0.21	0.15
8.	20	20	21	26	8	8	0.01	0.032	0.11	0.16	0.26	0.13
9.	20	17	28	24	10	9	0.01	0.04	0.05	0.25	0.18	0.21
10.	24	18	27	15	5	8	0.05	0.098	0.11	0.25	0.16	0.14
Pro-duct	p₂₁	p₂₂	R₁	R₂	h₁	h₂	l₁	l₂	f₁	f₂	C₁	C₂
1.	0.05	0.1	15	10	0.1	0.38	15	6	15	9	55	36
2.	0.1	0.02	14	15	0.5	0.10	12	5	18	15	35	26
3.	0.01	0.06	13	9	0.98	0.28	14	7	13	32	40	29
4.	0.06	0.09	12	11	0.52	0.3	12	10	16	8	60	23
5.	0.06	0.12	15	9	0.24	0.16	7	12	10	16	45	28
6.	0.09	0.05	12	13	0.35	0.28	9	10	8	14	65	34
7.	0.05	0.07	14	8	0.16	0.24	26	10	21	10	80	26
8.	0.05	0.045	15	12	0.25	0.32	12	14	14	8	35	27
9.	0.08	0.065	12	9	0.42	0.18	11	4	16	15	50	25
10.	0.11	0.08	11	11	0.25	0.2	13	8	13	17	75	24

The design matrix of the selected central composite design along with the experimental results is shown in Table 2. The PTYPE 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 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 this purpose, a multiple regression analysis and an analysis of variance (ANOVA) are developed for both responses.

Table 3

Design matrix of the central composite design

<i>Std. Order</i>	<i>Run Order</i>	<i>Pt Type</i>	<i>Blocks</i>	<i>Pop Size(N)</i>	<i>Max Gen</i>	<i>P_c</i>	<i>P_m</i>	<i>Fitness</i>
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

The analysis of the variance 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.

Multiple regression analysis for fitness

Table 4

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 × P_c	12.2	16.21	0.755	0.461
P_m × 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 × P_m	14.4	21.67	0.663	0.517

 $S = 86.6923$ $PRESS = 561011$ $R-Sq = 89.76\%$

The ANOVA for fitness is shown in Table 5.

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				

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 algorithm (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:

$$\begin{aligned}
 \text{FitnessFunction} = & 29166.7 - 53.9N - 129.3\text{MaxGen} - 13.1P_c - 119.1P_m \\
 & + 15N^2 + 57.6\text{MaxGen}^2 + 12.2P_c^2 + 29.4P_m^2 + 15.5N \\
 & \times \text{MaxGen} - 23.6N \times P_c + 67.9N \times P_m + 21.3\text{MaxGen} \\
 & \times P_c + 42.7\text{MaxGen} \times P_m + 14.4P_c \times P_m
 \end{aligned} \tag{23}$$

Since the most significant GA parameters have been 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. 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.

Table 6

The Lingo optimum solution

N*	MaxGen*	P _c *	P _m *	Cost
60	500	0.85	0.35	14896990

A Numerical Example

Let us consider a multi-product inventory control model with ten products and general data 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; the order quantity for product five by supplier two is 51 or the 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 Fig. 8.

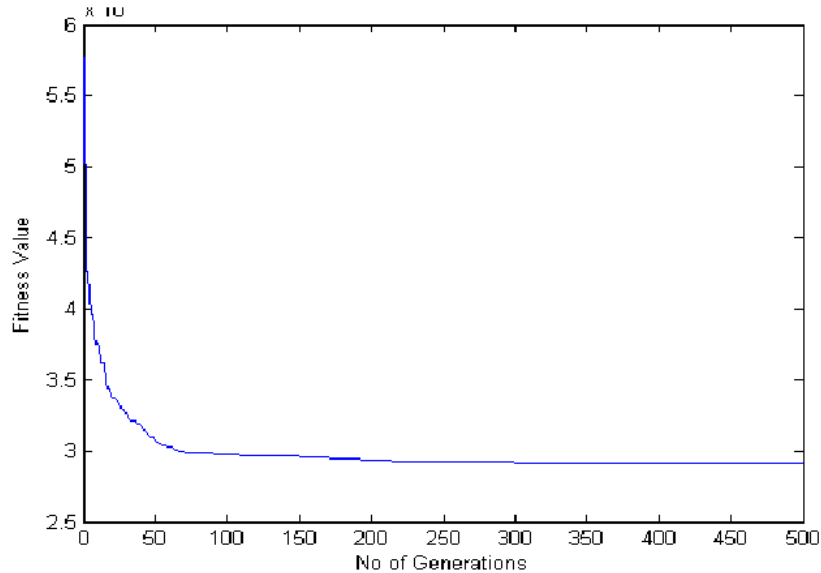


Figure 8. The graph of the convergence path

Conclusions

In this paper a multi-product and multi-supplier EPQ model with limited warehouse space and budget was presented, with defective items and reworking being considered. 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. In the end, a numerical example was presented to demonstrate the application of the proposed methodology. In this example, the optimum values of the GA parameters were obtained using the RSM.

References

- [1] Harris, F. W., How many parts to make at once, *Factory Management*, 10, pp. 135–146, 1913.
- [2] Jamal, A. M. M, Sarker, B. R., Mondal, S., Optimal manufacturing batch size with rework process at a single-stage production system, *Computers and Industrial Engineering*, 47, pp. 77–89, 2004.
- [3] Salameh, M. K., Jaber, M. Y., Economic production quantity model for items with imperfect quality, *International Journal of Production Economics*, 64, pp. 59–64, 2000.
- [4] Rosenblatt, M. J., Lee, H. L., Economic production cycles with imperfect production process, *IIE Transactions*, 18, pp. 48–55, 1986.

- [5] Taleizadeh, A. A., Niaki, S. T.A., Najafi, A. A., Multiproduct single machine production system with stochastic scrapped production rate, partial backordering and service level constraint, *Journal of Computational and Applied Mathematics*, 233, pp. 1834-1849, 2010.
- [6] Goyal, S. K., Huang, C. K., Chen, K. C., A simple integrated production policy of an imperfect item for vendor and buyer, *Production Planning and Control*, 14 (7), pp. 596–602, 2003.
- [7] Hou, K. L., Lin, L. C., Optimal production run length and capital investment with an imperfect production processes, *International Journal of System Sciences*, 35, pp. 133-137, 2004.
- [8] Leung, K. N. F., A generalized geometric-programming solution to An economic production quantity model with flexibility and reliability considerations, *European Journal of Operational Research*, 176 (1), pp. 240–251, 2007.
- [9] Hou, C. I., Lo, C. Y., Leu, J. H., Use genetic algorithm in production and inventory strategy. In: Proceedings of the 2007 IEEE international conference on industrial engineering and engineering management (IEEM) conference, pp. 963–967.
- [10] Hou, K. L., An EPQ model with setup cost and process quality as functions of capital expenditure, *Applied Mathematical Modeling*, 31, pp. 10-17, 2007.
- [11] Hsu, W. K. K. Yu, H. F., EOQ model for imperfective items under a one-time only discount, *Omega*, 37, pp. 1018-1026, 2009.
- [12] Porteus, E. L., Optimal lot sizing process quality improvement and setup cost reduction, *Operations Research*, 34, pp. 137-144, 1986.
- [13] Tapiero, C., Production learning and quality control, *IIE Transactions*, 19, pp. 362-370, 1987.
- [14] Maddah, B., Moussawi, L., Jaber, M. Y., Lot sizing with a Markov production process and imperfect items scrapped, *International Journal of Production Economics*, 124, 340–347, 2010.
- [15] Wahab, M. I. M., Jaber, M. Y., Economic order quantity model for items with imperfect quality, different holding costs, and learning effects: A note, *Computers & Industrial Engineering*, 58, pp. 186–190, 2010.
- [16] Maddah, B., Jaber, M. Y., Economic order quantity for items with imperfect quality:Revisited, *International Journal of Production Economics*, 112, pp. 808–815, 2008.
- [17] Jaber, M. Y., Goyal, S. K., Imran, M., Economic production quantity for items with imperfect quality subjected to learning effects, *International Journal of Production Economics*, 115, 143–150, 2008.
- [18] Bag, S., Chakraborty, D., Roy, A. R., A production inventory model with fuzzy random demand and with flexibility and reliability considerations, *Computers & Industrial Engineering*, 56, pp. 411–416, 2009.
- [19] Božanić, D., Pamučar, D., Evaluating locations for river crossing using fuzzy logic, *Military Technical Courier/Vojnotehnički glasnik*, Vol. 58, No. 1, pp. 129-145, ISSN 0042-8469, UDC 623+355/359, Ministarstvo odbrane Republike Srbije, Belgrade, 2010.
- [20] Đorović, B., Ljubojević, S., Dimić, S., Optimizing transportation processes using the route model, *Military Technical Courier/Vojnotehnički glasnik*, Vol. 53, No. 1, pp. 76-85, ISSN 0042-8469, UDC 623+355/359, Ministarstvo odbrane Republike Srbije, Belgrade, 2005.

[21] Ljubojević, S., Possibilities of implementation of contemporary traffic management systems in the defense system, *Military Technical Courier/Vojnotehnički glasnik*, Vol. 55, No. 4, pp. 463-474, ISSN 0042-8469, UDC 623+355/359, Ministarstvo odbrane Republike Srbije, Belgrade, 2007.

[22] Pamučar, D., Using fuzzy logic and neural networks during decision making proces in transport, *Military Technical Courier/Vojnotehnički glasnik*, Vol. 58, No. 3, pp 125-143, ISSN 0042-8469, UDC 623+355/359, Ministarstvo odbrane Republike Srbije, rade, 2010.

[23] Gopalan, M. N., Kannan, S., Expected number analysis of a two-server queuing network subject to inner-stage inspection and rework, *Computers and Operations Research*, 22, pp. 935–946, 1995.

[24] Liao, G. L., Chen, Y. H., Sheu S. H., Optimal Economic Production Quantity Policy for Imperfect Process with Imperfect Repair and Maintenance, *European Journal of Operational Research*, 195, pp. 348–357, 2009.

[25] Jaber, M. Y., Bonney, M., Moualek, I., An economic order quantity model for an imperfect production process with entropy cost, *International Journal of Production Economics*, 118, pp. 26–33, 2009.

[26] Chiu, S. W., Wang, S-L., Chiu, Y-S. P., Determining the optimal run time for EPQ model with scrap, rework, and stochastic breakdowns, *European Journal of Operational Research*, 180, pp. 664–676, 2007.

[27] Chung, K. J., Bounds for production lot sizing with machine breakdowns, *Computers & Industrial Engineering*, 32, pp. 139–144, 1997.

[28] Chiu, Y. P., Determining the optimal lot size for the finite production model with random defective rate, the rework process and backlogging, *Engineering Optimization* 35 (4), 427–437, 2003.

[29] Rezaei, J., Davoodi, M., A deterministic, multi-item inventory model with supplier selection and imperfect quality, *Applied Mathematical Modelling*, 32, pp. 2106–2116, 2008.

[30] Stockton, D. J., Quinn, L., Identifying economic order quantities using genetic algorithms, *International Journal of Operations & Production Management*, 13, pp. 92–103, 1993.

[31] Mondal, S., Maiti, M., Multi-item fuzzy EOQ models using genetic algorithm, *Computers & Industrial Engineering*, 44, pp. 105–117, 2002.

[32] Gupta, R. K., Bhunia, A. K., Goyal, S. K., An application of genetic algorithm in solving an inventory model with advanced payment and interval valued inventory costs, *Mathematical and Computer Modelling*, 49, pp. 893–905, 2009.

[33] Lotfi, G., Applying genetic algorithms to dynamic lot sizing with batch ordering, *Computers & Industrial Engineering*, 51, pp. 433–444, 2006.

[34] Pal, S., Maiti, M. K., Maiti, M., An EPQ model with price discounted promotional demand in an imprecise planning horizon via genetic algorithm, *Computers & Industrial Engineering*, 57, pp. 181–187, 2009.

[35] Pasandideh, S. H. R., Niaki, S. T. A., Yeganeh, J. A., A parameter-tuned genetic algorithm for multi-product economic production quantity model with space constraint, discrete delivery orders and shortages, *Advances in Engineering Software*, 2009.

- [36] Davis, L., *Handbook of genetic algorithms*. New York, Van Nostrand Reinhold, 1991.
- [37] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [38] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs*, AI Series, Springer, New York, 1994.
- [39] Holland, J.H., *Adoption in neural and artificial systems*, Ann Arbor, Michigan (USA): The University of Michigan Press, 1975.
- [40] Michalewicz, Z., Schoenauer, M., Evolutionary algorithms for constrained parameter optimization problems, *Evolutionary Computation*, 4 (1), pp. 1–32, 1996.
- [41] Montgomery, D. C., *Design and analysis of experiments*, New York (USA), John Wiley and Sons, 2001.
- [42] Homaifar, A., Lai, S. H-V., Qi, X., Constrained optimization via genetic algorithms, *Simulation*, 62 (4), pp. 242–254, 1994.
- [43] Pasandideh, S. H. R., Niaki, S. T. A., Mirhosseyni, S. S., A parameter-tuned genetic algorithm to solve multi-product economic production quantity model with defective items, rework, and constrained space, *The International Journal of Advanced Manufacturing Technology*, 49, pp. 827–837, 2010.

OPTIMIZACIJA MODELA ZA PROIZVODNJU I KONTROLU ZALIHA PRIMENOM GENETSKIH ALGORITAMA

OBLAST: matematika (operaciona istraživanja)
VRSTA ČLANKA: originalni naučni članak

Sažetak:

U ovom radu izvršeno je proširenje (optimizacija) modela za ekonomsko upravljanje proizvodnjom i skladištenjem dobara. Model je proširen imajući u vidu heterogenost proizvoda i dobavljača, kao i ograničenja u vidu magacinskog prostora i budžetska ograničenja. U radu je pokazano da opisani problem ima nelinearna ograničenja i predloženo je rešenje primenom genetskih algoritama. Na kraju rada prikazan je numerički primer koji je obrađen na osnovu opisane metodologije.

Uvod i pregled literature

Ovaj model se može smatrati kao proširenje poznatog modela Ekonomska količina naloga (EKN) [1]. Bez obzira na jednostavnost EKN i EPK modela, oni se i dalje primenjuju u industriji [2].

Tradicionalni EPK modeli pretpostavljaju da proizvodni proces proizvodi proizvode savršenog kvaliteta. Međutim, greške koje se javljaju tokom proizvodnog procesa su činjenice koje su neizostavne za svako radno mesto. Mnogo je realnije pretpostaviti da je proizvodni proces ponekad nesavršen tj. praćen je greškama. Takav proces proizvodnje naziva se nesavršen proces proizvodnje [3].

Definisanje problema

U radu je razmatran problem u kojem kompanija proizvodi više proizvoda za više dobavljača. Svi proizvodi koji se proizvedu svrstavaju se kao savršeni, nesavršeni (defektni, ali se mogu popraviti) i potpuno nesavrše-

ni (defektni, ali se ne mogu popraviti) proizvoda. Cilj je da se utvrdi količina proizvoda koju zahteva svaki snabdevač, a da se istovremeno ukupna količina zaliha se smanjuje uz zadovoljenje svih ograničenja.

Modeliranje problema

Da bi se matematički formulisao problem, uzeli smo prednost klasičnog EPK model i proširili ga na problem koji sadrži savršene, nesavršene i potpuno nesavršene proizvode zajedno sa ograničenjima u pogledu skladišta i budžetkih kapaciteta. Da bi se modelirao problem, prvo su definisani parametri i promenljive. Različiti troškovi su izvedeni u narednom odeljku, a na kraju je predstavljen model problema.

Troškovi kalkulacije

Ukupan godišnji trošak proizvoda predstavlja zbir ukupnih troškova nabavki, ukupni troškovi inspekcija, ukupni transportni troškovi, ukupni troškovi za VIP zalihe i ukupni troškovi održavanja skladišnog prostora za sve proizvode. Drugim rečima, imamo:

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

Formulacija problema

Cilj modela jeste da se odredi optimalna vrednost za svakog dobavljača, tako da se ukupan godišnji trošak svede na minimum, uz zadovoljenje sledećih ograničenja:

1. Ograničenje magacinskog prostora za skladištenje proizvoda i
2. Budžetska ograničenja.

Dakle, problem može biti formulisan kao:

$$\text{MinTC} = \sum_{i=1}^n \sum_{j=1}^m \left[\frac{m_j D_{ij}}{(1-p_{2ij})} + \frac{A_j D_{ij}}{Q_{ij}(1-p_{2ij})} + \frac{I_j D_{ij}}{(1-p_{2ij})} + \alpha(1-p_{2ij})M_{ij}Q_{ij} + \frac{h_j D_{ij}}{2(1-p_{2ij})} [S_{ij} + m_j(1+p_{1ij})Q_{ij}] \right. \\ \left. \times \left[2M_{ij} + \frac{R_j S_{ij}}{Q_{ij}} + R_j m_j(1+p_{1ij}) \right] + h_{ij} \right. \\ \left. \times \left\{ M_{ij} + R_j \left[\frac{S_{ij}}{Q_{ij}} + m_j(1+p_{1ij}) \right] \right\} \frac{1}{2} Q_{ij}(1-p_{2ij}) \right]$$

gde je:

$$\sum_{j=1}^m \sum_{i=1}^n (1-p_{2ij}) f_{ij} Q_{ij} \leq F \quad (18)$$

$$\sum_{j=1}^m \sum_{i=1}^n (1-p_{2ij}) C_{ij} Q_{ij} \leq X \quad (19)$$

$$Q_{ij} > 0; j = 1 \dots m, i = 1 \dots n \quad (20)$$

Genetski algoritam

Genetski algoritmi (GA) imitiraju evolutivni proces primenom strategije koja se često naziva „opstanak najsposobnijih jedinki“. Genetski algoritam je heuristički metod optimizacije koji je zasnovan na evoluciji živih organizama. Genetski algoritmi predloženi su od strane Johna H. Hollanda još u ranim sedamdesetim godinama prošlog veka. Osnovna je ideja ovih algoritama je kopiranje rada prirodnog evolucijskog procesa. Optimalna rješenja zadatih problema nije uvek moguće pronaći klasičnim metodama pretraživanja jer je prostor rešenja problema najčešće prevelik da bi računar mogao pretražiti sva rešenja u nekom razumnom vremenu. Ako evoluciju promatramo kao „algoritam“ koji traži optimalno rešenje (cilj je stvoriti što napredniji organizam), možemo reći da je prirodna evolucija vrlo uspešan „algoritam“. Nakon 4,5 milijarde godina izvođenja taj algoritam i dalje pronalazi kvalitetna rješenja. Živa bića danas rešavaju velik broj problema, uključujući mnoge probleme koje današnji računari ne mogu rešiti.

Osnovni parametri koji određuju evoluciju, pa tako i genetske algoritme, su: način kodiranja svojstava jedne jedinke, način evolucije dobrote jedinke, način selekcije, ukrštanja i mutacije. Navedeni parametri su objašnjeni u narednom delu rada.

Zastupljenost hromozoma

Prvi korak u razvoju GA je kodiranje promenljivih u obliku konačnog niza koji se nazivahromozom. Tradicionalno, hromozomi su jednostavni binarni string. Ova jednostavna zastupljenost nije pogodna za kombinatorne probleme. Međutim, da bi se navedeni problem eliminisao, u ovom radu su hromozomi predstavljeni celim brojevima. U GA smo izabrali dvodimenzionalnu strukturu u predstavljanju rešenja. Ova matrica ima n redova i kolona. Elementi svake kolone pokazuju broj proizvoda. Pored toga, elementi svakog reda pokazuje broj dobavljača. Na primer, kada imamo 2 dobavljača i 5 proizvoda, hromozom matrica je 2×5 .

Inicijalizaciju populacije

Za bilo GA neophodno je da se definiše početna populacija. Najčešći metod je da se nasumično generišu rešenja za celu populaciju. Nakon toga GA iterativno poboljšava postojeća rešenja. Kreiranje inicijalne populacije je jedan od četirij trenutaka u kojima se u procesu optimizacije koristi slučajni odabir. Osim ovde, slučajni odabir se koristi i pri selekciji jedinki za sledeću generaciju, odabiru tačke prekida kod ukrštanja i za mutaciju gena jedinke. Ograničenje za obradu i fitnes evaluaciju

Fitnes vrednost je mera dobrote rešenja u odnosu na originalni cilj. Pronalaženje izvodljivog rešenja (koje će zadovoljiti sva ograničenja u modelu) je veliki problem i postoje poteškoće u pronalaženju čak i jednog izvodljivog rešenja. Stoga je korišćen navedeni metod da bi se eliminisali opisani nedostaci.

Izbor operatera

Izbor roditelja za proizvodnju narednih populacija GA igra izuzetno važnu ulogu. Cilj je da se omogući najsposobnijim jedinkama da budu izabrani

i da se češće reprodukuju. Međutim, sve jedinke u populaciji imaju šanse da budu izabrane za reprodukciju sledeće generacije. Svakoj jedinki dodeljena je određena verovatnoća. Time se postiže to da jedinke sa većom verovatnoćom imaju veću šansu da budu izabrane. Postoji nekoliko načina za određivanje i raspoređivanje verovatnoće jedinkama, na primer, rulet selekcija, turnir selekcija i proširenja tehnike skaliranja i metode rangiranja.

Genetski operatori

Postoje dve vrste operatera u genetskim algoritmima: mutacija i ukrštanje.

U procesu ukrštanja učestvuju dve jedinke koje se nazivaju roditelji. Dakle, ukrštanje je binarni operator. Ukrštanjem nastaje jedna ili dve nove jedinke koje se nazivaju deca. Najvažnija karakteristika ukrštanja je da deca nasleđuju svojstva svojih roditelja. Ako su roditelji dobri (prošli su proces selekcije), tada će najverovatnije i dete biti dobro, ako ne i bolje od svojih roditelja.

Drugi operator koji je karakterističan za genetski algoritam je mutacija ili slučajna promena jednog ili više gena.

Provera rešenja

Poslednji korak u metodologiji da provera da li je metod našao rešenje koje je dovoljno dobro da ispuni očekivanja korisnika. Provera rešenja podrazumeva proveru ispunjenja postavljenih kriterijuma.

Podešavanje parametara genetskog algoritma

Jedan od važnih postupaka prilikom sprovođenja genetskog algoritma je podešavanje parametara. U tom cilju izabrana je metoda Centralnog kompozitnog dizajna.

Diskusija

Nakon definisanja najznačajnijih GA parametara pristupa se utvrđivanju najbolje vrednosti ovih parametara koji dovode do najbolje vrednosti fitnes funkcije. Tabela 6 pokazuje optimalne rezultate.

U sledećem delu na numeričkom primeru prikazana je primenljivost predloženih parametara GA.

Zaključak

U ovom radu prikazan je model koji analizira više proizvoda i više dobavljača sa realnim ograničenjima u pogledu magacinskog prostora i budžeta. Pod ovim uslovima, problem je formulisan kao nelinearan i definisan je genetski algoritam za rešenje opisanog problema. Na kraju je dat numerički primer koji demonstrira primenu predložene metodologije.

Ključne reči: Genetski algoritam, ekonomske proizvodnja dobara, optimizacija.

Datum prijema članka: 17. 08. 2011.

Datum dostavljanja ispravki rukopisa: 18. 10. 2011.

Datum konačnog prihvatanja članka za objavljivanje: 19. 10. 2011.