Hafshejani *et al. Journal of Industrial Engineering International* 2012, **8**:27
http://www.jiei-tsb.com/content/8/1/27

Journal of Industrial Engineering International
a SpringerOpen Journal

ORIGINAL RESEARCH Open Access

Retracted: Using genetic algorithm approach to solve a multi-product EPQ model with defective items, rework, and constrained space

Kiamars Fathi Hafshejani[1*], Changiz Valmohammadi[1] and Alireza Khakpoor[2]

Retraction

Springer would like to retract the article "The Hafshejani *et al.*" published in Journal of Industial Engineering International (2012, **8**:27). The article was mistakenly published due to a workflow error although it had not been accepted by the Editorial Board of the journal. Springer accepts full responsibility for this and would like to apologize to the authors of the article as well as the Editors and readers of the journals.

Abstract

The Economic Production Quantity (EPQ) model is often used in the manufacturing sector to assist firms in determining the optimal production lot size that minimizes overall production-inventory costs. There are some assumptions in the EPQ model that restrict this model for real-world applications. Some of these assumptions are (1) infinite space of warehouse, (2) all of the produced items are perfect, and (3) only one type of goods is produced. In this paper, we develop the EPQ model by assuming that each produced lot contains some imperfect items and scraps. In addition, we have more than one kind of products along with warehouse space limitations. Under these conditions, we formulate the problem as a non-linear programming model and propose a genetic algorithm to solve it. At the end, we present a numerical example to illustrate the applications of the proposed methodology and identify the optimal value of the parameters of the genetic algorithm.

Keywords: EPQ, Multi-product, Perfect, Imperfect and scrap items, Constrained space, Genetic algorithm

Background

The Economic Production Quantity (EPQ) model can be considered as an extension to the well-known Economic Order Quantity (EOQ) model, and it is a technique to find out optimum production quantity by considering costs of procurement, inventory holding, and shortage. As the first assumptions of the EPQ model may not be valid for many real-life conditions, many researchers have developed EOQ and EPQ models. In real-life manufacturing systems, generation of defective items is inevitable. Hence, many researchers consider producing defective items in EOQ and EPQ models. For instance, Hayek and Salameh (2001) assumed that all of the defective items produced are repairable and derived an optimal operating policy for the EPQ model. The basic assumptions of this model are allowing backorders; all of the defective items are reworked and become perfect quality. They also consider rework time in their model. Rosenblatt and Lee (1986) proposed an EPQ model for a production system which contains defective production. The basic assumption in their model is that the production system produces 100% non-defective products from the starting point of production until a time point which is a random variable. At this time point, the system becomes out of control and starts to produce defective items with a percentage of production until the end of production period. Also, they assumed that the distribution of time passes is exponential until the system gets out of control. Kim and Hong (1999) extended Rosenblatt and Lee's model with the assumption that the time passes are arbitrarily distributed until the system gets out of control. Salameh and Jaber (2000) developed an EPQ model for circumstances where a fraction of the ordered lot is of imperfect quality and has a uniform distribution. Their

* Correspondence: fathi@azad.ac.ir
[1]Department of Industrial Management, Islamic Azad University, South Tehran Branch, P.O. Box: 11365/4435, Tehran, Iran
Full list of author information is available at the end of the article

model assumed that shortage is not permitted. Chiu et al. (2007) presented a procedure to determine the optimal run time for an EPQ model with scrap, rework, and stochastic machine breakdowns. In real-life manufacturing systems, generation of defective items and random breakdown of production equipment are inevitable.

Hou (2007) presented an EPQ model with imperfect production processes, in which the setup cost and process quality are functions of capital expenditure. This model illustrates the relationship among production run length, setup reduction, and process quality improvement in an imperfect production system. He showed that investment in setup reduction leads to reduction in optimal production run length and reduces lot size, whereas investment in process quality improvement leads to an increase in optimal production run length and increases lot size. At the end, he proposed that it is very important to investigate the optimal allocation of investment between both options. In several cases, producing new or transforming defective products takes place on a common facility. Consequently, it is necessary to coordinate the production and rework activities with respect to the timing of operations and with regard to the appropriate lot sizes for both processes. Buscher and Lindner (2007) presented a lot size model which addresses all of these aspects. In addition, they cited that it is very important how completed units are assigned at one stage to partial lots - called batches - for shipment to the next operation.

Liao et al. (2009) studied maintenance and production programs with the EPQ model for an imperfect process involving a deteriorating production system with increasing hazard rate. The imperfect repair restores the system to an operating state but leaves its failure until perfect preventive maintenance (PM) is performed. They introduced two types of PM, namely imperfect and perfect PM. The probability that perfect PM is performed depends on the number of imperfect maintenance operations performed since the last renewal cycle. In addition, they represent that if the PM rate is estimated based on the actual data, analysts can use the learning curves to project the PM costs in the integrated EPQ model.

One of the most important aspects of the extension of EOQ and EPQ models is to fuzzify their parameters. For instance, in the research of Lee and Yao (1998), they fuzzified demand and production quantity to solve economic production quantity per cycle.

In this paper, we develop the EPQ model by assuming that each produced lot contains some imperfect items and scraps. In addition, we have more than one kind of products along with warehouse space limitations. Under these conditions, we formulate the problem as a nonlinear programming model and propose a genetic algorithm to solve it. At the end, we present a numerical example to illustrate the applications of the proposed methodology and identify the optimal value of the parameters of the genetic algorithm.

The remainder of the paper is as follows: in 'Problem definition' and 'Problem modeling' sections (Method), the definition and the modeling of the problem are presented, respectively, followed by a genetic approach in 'The solution algorithm' section to solve the model. In the 'A numerical example' section, a numerical example to demonstrate the application of proposed algorithms is presented. Finally, in the 'Conclusions' section, the paper wraps up with the conclusion and some recommendations for future research.

Methods

Problem definition

In this paper, we assume that a production company produces multi-products that receive raw materials from a supplier. All produced items are inspected, and the time of inspection is zero. After inspection, there are three types of products. The first type is perfect products, the second is defective but repairable items, and the third is defective and not repairable products. The second and third types are called imperfect and scrap, respectively. After separation of all products, the imperfect products are reworked and changed to perfect products. We assume that all imperfect products after reworking are changed to perfect products, and we sell scraps with reduced cost. We have three types of materials near the machine: (1) raw materials, (2) perfect products, and (3) reduced-cost products, all of which are called work in process (WIP) inventory. Other conditions are as follows:

(a) There are n kinds of products.
(b) The warehouse space of the company for all products is limited.
(c) Shortage and delay are not allowed.
(d) All parameters, such as demand rate, rate of imperfect and scrap items, and setup cost, are known and deterministic.

Under these conditions, we want to determine the optimal production quantity that minimizes total costs and satisfies the constraint, too, and we prove that the revenue of the inventory system does not depend on the lot size.

Problem modeling

For modeling, we have to extend the classical EPQ model with regard to the conditions of the problem. We note that in this problem, we have limited warehouse space and three types of products: perfect, imperfect, and scrap. In this section, at first, we define the parameters in the 'Parameters and notations' section. Then, we pictorially demonstrate the situation using an inventory graph in the 'Inventory graph' section. In the 'Costs calculations' section, we derived the different costs, and finally, we present the model of the problem in the 'Problem formulation' section.

Parameters and notations

For $i = 1,\ldots, n$, we define the parameters of the model as follows:

n the number of products
Q_i order quantity for product i
P_i production rate for product i
D_i demand rate for product i
A_i setup cost for each cycle for product i
h_i rate of holding cost for product i
M_i price of one raw material for product i
S_i setup time for product i
m_i time of machining for product i
R_i rate of production cost in unit time for product i
c_i average of cost for producing any unit of product i
v_i average of value added for product i
$\bar{W}_i$ average of invest value of work in process inventory for product i
$\bar{I}_i$ average of inventory in warehouse for product i
p_{1i} percent of imperfect items for product i
p_{2i} percent of scrap items for product i
s_{1i} price of perfect products for product i
s_{2i} price of scrap items for product is
T_i cycle time for product i
TP_i sum of setup and production time in each cycle time for product i
t_i average time of production for any unit of product i
f_i required space for perfect product for product i
F total space of warehouse for all products
$\mathrm{TC}_{\mathrm{P}i}$ total procurement cost for product i
$\mathrm{TC}_{\mathrm{O}i}$ total setup cost for product i
$\mathrm{TC}_{\mathrm{I}i}$ total inspection cost for product i
$\mathrm{TC}_{\mathrm{WIP}i}$ total holding cost for work in process inventory for product i
$\mathrm{TC}_{\mathrm{H}i}$ total holding cost for perfect products in warehouse for product i
TC total annual cost for all products

Inventory graph

In order to calculate all of the inventory's costs, it is necessary to survey the work in process and the warehouse inventory. For the problem at hand, the graph of quantity of raw materials in terms of time is demonstrated in Figure 1a. In addition, the graphs of perfect and scrap of work in process inventory in terms of time are shown in Figure 1b,c, respectively. In this problem, the rate of demand is constant, and then the graph of the quantity of final products in the warehouse is similar to the EOQ model which is illustrated in Figure 1d; it is obvious that only perfect products are delivered to the warehouse.

Cost calculations

In this model, shortage and delay are not permitted. Hence, total cost of all products per year (TC) is the sum of total providence cost ($\mathrm{TC_P}$), total setup cost ($\mathrm{TC_O}$), total inspection cost ($\mathrm{TC_I}$), total holding cost for work in process inventory ($\mathrm{TC_{WIP}}$), and total holding cost for inventory in the warehouse ($\mathrm{TC_H}$) for all products. In other words, we have

$$\mathrm{TC} = \sum_{i=1}^{n}(\mathrm{TC}_{\mathrm{P}i} + \mathrm{TC}_{\mathrm{O}i} + \mathrm{TC}_{\mathrm{I}i} + \mathrm{TC}_{\mathrm{WIP}i} + \mathrm{TC}_{\mathrm{H}i}). \quad (1)$$

However, before starting to calculate the costs, it is necessary to define some of the parameters. In any cycle, setup time, production time, and reworking time are equal to S_i, m_iQ_i, and $m_i(p_{1i}Q_i)$, respectively. Then, total setup and production time (TP_i) for product i is as follows:

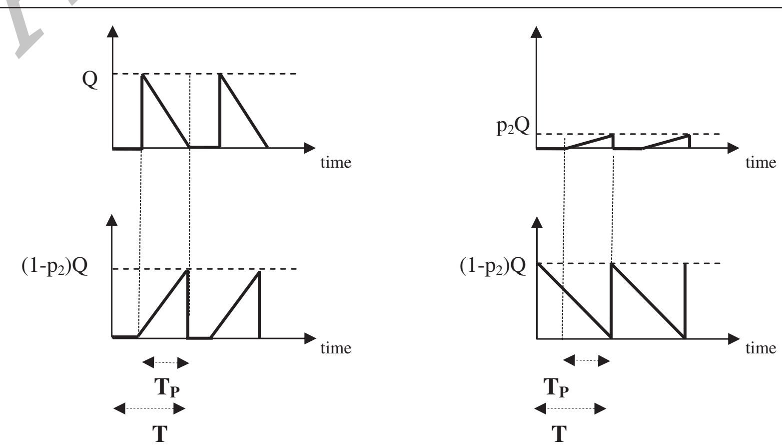

Figure 1 Inventory graphs in terms of time: (a) raw material, (b) perfect products, (c) scrap items, and (d) product in the warehouse.

$$\begin{aligned}\mathrm{TP}_i &= S_i + m_iQ_i + m_i(p_{1i}Q_i)\\ &= S_i + m_iQ_i(1+p_{1i}).\end{aligned} \tag{2}$$

Then, the average of operation time for each unit of product i is equal to Equation 3:

$$t_i = \frac{\mathrm{TP}_i}{Q_i} = \frac{S_i}{Q_i} + m_i(1+p_{1i}). \tag{3}$$

Regarding our definition of R_i, which is the rate of production cost per unit time, we can define v_i and c_i as follows:

$$v_i = R_it_i = R_i\left[\frac{S_i}{Q_i} + m_i(1+p_{1i})\right] \tag{4}$$

$$c_i = M_i + v_i = M_i + R_i\left[\frac{S_i}{Q_i} + m_i(1+p_{1i})\right]. \tag{5}$$

As in this model delay is not allowed, supply and demand are equal together. Hence, we have

$$(1-p_{2i})Q_i = D_iT_i \Rightarrow T_i = \frac{(1+p_{2i})Q_i}{D_i}. \tag{6}$$

In the 'Parameters and notations' section, we defined s_{1i} and s_{2i} as the price of perfect and scrap items, respectively. Then, average revenue in unit time is as follows:

$$\begin{aligned}\mathrm{TR}_i &= \frac{(1-p_{2i})Q_is_{1i} + p_{2i}Q_is_{2i}}{T_i}\\ &= D_is_{1i} + \frac{p_{2i}}{(1-p_{2i})}D_is_{2i}; \quad i=1,\ldots,n.\end{aligned} \tag{7}$$

It can be seen that revenue in unit time does not depend on the lot size.

Now, with regard to Equations 2 to 6, we can calculate all costs. Since the annual rate of demand for each product is known, total provision cost for product i per unit time is obtained through Equation 8:

$$\mathrm{TC_P} = \sum\nolimits_{i=1}^{n}\frac{M_iQ_i}{T_i} = \sum\nolimits_{i=1}^{n}\frac{M_iD_i}{(1-p_{2i})}; \quad i=1,\ldots,n. \tag{8}$$

The cost of setup accrues only one time for each product. We can calculate total setup cost per unit time as follows:

$$\mathrm{TC_O} = \sum\nolimits_{i=1}^{n}\frac{A_i}{T_i} = \sum\nolimits_{i=1}^{n}\frac{A_iD_i}{Q_i(1-p_{2i})}; \quad i=1,\ldots,n. \tag{9}$$

In this paper, it is assumed that all of the products are inspected and all of the imperfect products after reworking are transformed to perfect quality. Then, any product is inspected only once after its production. Total cost of inspection per unit time equals Equation 10:

$$\mathrm{TC_I} = \sum\nolimits_{i=1}^{n}\frac{I_iQ_i}{T_i} = \sum\nolimits_{i=1}^{n}\frac{I_iD_i}{(1-p_{2i})}; \quad i=1,\ldots,n. \tag{10}$$

In this stage, we want to calculate the holding cost for work in process inventory. The most famous method to present the holding cost per unit time (C_{H}) is as follows (Silver et al. 1998):

$$C_{\mathrm{H}i} = h_ic_i\bar{I}_i, \tag{11}$$

where h_i is the rate of holding cost for product i, so we have

$$C_{\mathrm{WIP}i} = h_i\bar{w}_i, \tag{12}$$

where $\bar{w}$ is the average monetary value for work in process inventory, that is, the sum of the monetary value of the average of raw materials, perfect items, and reduced-cost products. For any items, the average inventory of raw materials is the total of raw materials (it is equal to the surface under the graph of inventory) divided by the time of one cycle. The average of the monetary value of raw materials is obtained by multiplying the average inventory of raw materials by the price of any raw materials. Similarly, the average monetary value of perfect products and reduced-cost products can be calculated. Then, the average monetary value for all of the works in process inventory is as follows (Equation 13):

$$\begin{aligned}\bar{w}_i &= \frac{\frac{1}{2}Q_i\mathrm{TP}_i}{T_i}M_i + \frac{\frac{1}{2}(1-p_{2i})Q_i\mathrm{TP}_i}{T_i}c_i + \frac{\frac{1}{2}p_{2i}Q_i\mathrm{TP}_i}{T_i}c_i\\ &= \frac{1}{2}\frac{Q_i\mathrm{TP}_i}{T_i}(M_i+c_i) = \frac{D_i}{2(1-p_{2i})}\\ &\quad (S_i + m_i(1+p_{1i})Q_i)\left[2M_i + \frac{R_iS_i}{Q_i} + R_im_i(1+p_{1i})\right].\end{aligned} \tag{13}$$

With regard to Equations 12 and 13, the average holding cost for work in process inventory is as follows:

$$\mathrm{TC_{WIP}} = \sum\nolimits_{i=1}^{n}h\frac{Di}{2(1-p_{2i})}(S_i + m_i(1+p_{1i})Q_i)\left[2M_i + \frac{R_iS_i}{Q_i} + R_im_i(1+p_{1i})\right] i=1,\ldots,n. \tag{14}$$

In this step, we want to calculate the holding cost, but at first, we have to estimate the average of inventory in the warehouse. Regarding Figure 1d, it is equal to Equation 15:

$$\bar{I}_i = \frac{\frac{1}{2}Q_i(1-p_{2i})T_i}{T_i} = \frac{1}{2}Q_i(1-p_{2i}). \tag{15}$$

Then, the total holding cost is similar to Equation 16:

$$\begin{aligned} \mathrm{TC_H} &= \sum_{i=1}^{n} h_i c_i \bar{I}_i \\ &= \frac{1}{2}h_i\left[M_i + \frac{R_iS_i}{Q_i} + R_i m_i(1+p_{1i})\right] \\ &\quad (1-p_{2i})Q_i; \quad i=1,\ldots n. \end{aligned} \tag{16}$$

Now, we can calculate the total annual cost of all products using Equation 17:

$$\begin{aligned} \mathrm{TC} = \sum_{i=1}^{n} & \frac{M_iD_i}{(1-p_{2i})} + \frac{A_iD_i}{Q_i(1-p_{2i})} + \frac{I_iD_i}{(1-p_{2i})} + \frac{h_i}{2} \\ & \left[\frac{D_i}{(1-p_{2i})}(S_i + m_iQ_i(1+p_{1i}))\right. \\ & \left[2M_i + \frac{R_iS_i}{Q_i} + R_im_i(1+p_{1i})\right] + (1-p_{2i})Q_i \\ & \left.\left[M_i + \frac{R_iS_i}{Q_i} + R_im_i(1+p_{1i})\right]\right]; \quad i=1,\ldots,n. \end{aligned} \tag{17}$$

Problem formulation

As we described earlier, the goal is to determine economic production quantity in order to minimize the total annual cost given in Equation 17 and satisfy the constraint.

Hence, we can formulate the problem as follows:

$$\begin{aligned} \text{Min TC} = \sum_{i=1}^{n} & \frac{M_iD_i}{(1-p_{2i})} + \frac{A_iD_i}{Q_i(1-p_{2i})} + \frac{I_iD_i}{(1-p_{2i})} \\ & + \frac{h_i}{2}\left[\begin{array}{l} \frac{D_i}{(1-p_{2i})}(S_i + m_iQ_i(1+p_{1i})) \\ \left[2M_i + \frac{R_iS_i}{Q_i} + R_im_i(1+p_{1i})\right] \\ + (1-p_{2i})Q_i\left[M_i + \frac{R_iS_i}{Q_i} + R_im_i(1+p_{1i})\right] \end{array}\right]; \\ & i = 1,\ldots,n \\ \text{St}: \; & \sum_{i=1}^{n}(1-p_{2i})Q_if_i \le F \\ & Q_i > 0. \end{aligned} \tag{18}$$

In the next section, we present an efficient algorithm to solve this objective function.

The solution algorithm

The formulation given in Equation 18 is a non-linear programming model which is hard to solve by conventional optimization techniques. Then, we have to use heuristic search. Simulating the natural evolutionary process of human beings results in stochastic optimization techniques, of which the most applicable is genetic algorithm (Gen 1997). Solving a non-linear programming problem by traditional techniques will lead to a local optimum solution. In other words, the global optimum cannot be obtained. If meta-heuristic approach such as genetic algorithm (GA) is used, we will get the closest solution to the global optimum (Shabani et al. 2011).

The usual form of GA was described by Goldberg (1989). Generally, conventional algorithms for optimization have a point-to-point approach and maybe they fall in local optima. However, GA performs a multiple directional search by maintaining a population of solutions. Each solution is called a chromosome. The chromosomes which improved in consecutive iterations are called generations. During each generation, the chromosomes are evaluated using some measures of fitness. To create the next generation, new chromosomes, called offspring, are formed by either crossover or mutation operator. In addition, the chromosomes that have higher fitness are kept for the next generation. After several generations, the algorithm converges to the best chromosome which hopefully represents the optimum or suboptimal solution to the problem.

In the next subsections, we describe the required steps to solve the aforementioned model by GA.

Initial conditions

The required initial conditions to start solving a model with GA are as follows:

(a) Population size: It is the number of chromosomes that is kept in each generation, but its value changes in consecutive iterations to improve objective function; we denoted it by pop-size or N.
(b) Crossover rate: It is defined as the ratio of the number of offspring produced by crossover operator and denoted by P_C.
(c) Mutation rate: It controls the number of chromosomes to undergo mutation operation and denoted by P_M.

Chromosome

In the GA method, each individual in the population is called a chromosome, and any chromosome contains some genes. In this model, we present a chromosome by a matrix that has one row and n columns. Each column shows the quantity of production for each product. Figure 2 presents one chromosome for the problem at hand.

Evaluation

Each chromosome in the GA method was evaluated with some measures, and we have to assign a fitness value for

$$Q_i\ [Q_1\ \ Q_2\ \ \ldots Q_n]$$

Figure 2 Chromosome presentation.

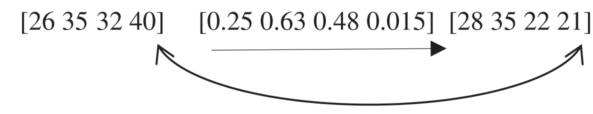

Figure 4 Graphical representation of mutation operator.

it. In this model, it signifies the value of the objective function. For a constrained optimization problem, the main issue is to control the feasibility of chromosomes. In order to control infeasible solutions, we have to employ penalty policy presented by Gen (1997). Because this problem is a minimization one, penalty is defined as a positive value. The more infeasible chromosomes, the more are the penalties. Hence, when a chromosome is feasible, its penalty is zero. In this case, the fitness function for a chromosome is the sum of its objective function and its penalty.

Initial population

To perform GA, at first, we have to define the first generation for the GA method randomly with regard to population size.

Crossover

Crossover is the main genetic operator. It operates on two chromosomes at a time and generates offspring by combining the features of both chromosomes (Gen 1997). At first, we have to select a pair of chromosomes from the generation randomly with probability P_c. One simple way to achieve crossover is to create a binary chromosome randomly. We do not change the genes with a value of 0, but we consider the genes with a value of 1. To produce offsprings, genes of parents are crossed so that they have the same positions as those with a value of 1. Figure 3 demonstrates the crossover operation for the four products.

Mutation

Mutation is a background operator which produces a random change in chromosomes, and maybe it results in a chromosome with a higher fitness value. For our model, a chromosome is randomly produced as its genes are between 0 and 1. We do not change the genes with a value more than that of P_M, but, we replace the genes with a value less than that of P_M with a new random value within the boundaries of the parameter. Figure 4 shows an example of the mutation operator for the four products, and P_M is equal to 0.05.

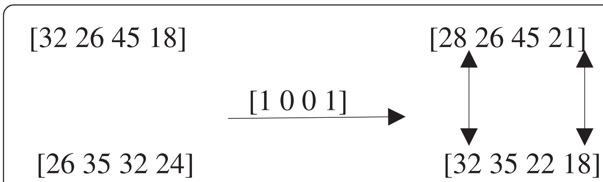

Figure 3 An example of crossover operation.

Chromosome selection

After producing the offsprings with the crossover and mutation operators and measuring their fitness value, we have to make the next generation based on the chromosomes which have the highest fitness. Then, we select N chromosomes among the parents and offsprings with the best fitness value. In this problem, we select the chromosomes that result in less cost.

Stopping criteria

Genetic algorithm is a sequence of computational steps that converge to optimal solution. We have to define some measures to stop the generations. Stopping criteria are a set of conditions that when the method satisfies them, a good solution is obtained. In this paper, we use two stopping criteria. At first, after some generations, the algorithm examines the values of fitness. If there is no improvement in the fitness function values for some consecutive generations, the algorithm stops. In another case, the algorithm continues for some generations again and checks the values of fitness another time.

Results and discussions

A numerical example

In order to demonstrate the application of the proposed genetic algorithm, we present a numerical example in this section. The values of all parameters are given in Table 1. In addition, we assume that the total space of the warehouse is 1,500.

With this data, we run the proposed genetic algorithm for 110 times, but we only change the parameters of the genetic algorithm. We assume that the rate of crossover and mutation change is in the range of 0.45 to 0.85 and 0.005 to 0.05, respectively. Also, the number of

Table 1 Data for the example

Product	A_i	M_i	D_i	S_i	m_i	p_{1i}	p_{2i}	R_i	h_i	I_i	f_i
1	12	8	32	0.009	0.01	014	0.05	12	0.16	15	13
2	13	9	25	0.001	0.04	0.25	0.09	14	0.25	12	16
3	14	5	42	0.002	0.099	0.15	0.05	12	0.42	14	10
4	15	9	38	0.004	0.03	0.28	0.11	11	0.25	12	8

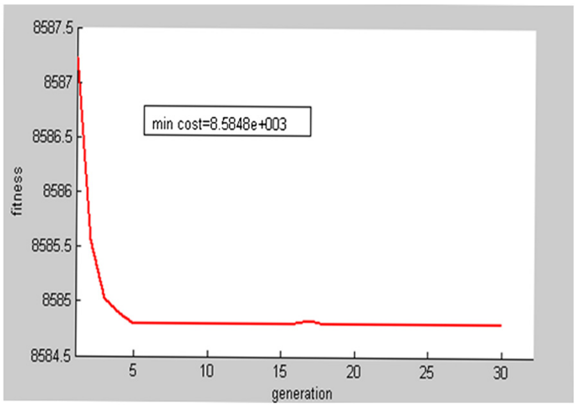

Figure 5 The graph of convergence.

chromosomes (pop-size) falls between 20 and 60. Next, we estimate the equation of regression of these results. In the next step, we optimized this equation with regard to the range of parameters. In addition, we assume that the pop-size is only an integer.

The optimized parameters are 0.85, 0.05, and 60 for cross-over rate, mutation rate, and pop-size, respectively. Now, with these parameters, the graph of fitness value in terms of the number of generations is presented in Figure 5.

Figure 3 shows that the minimum cost is 8,584 and the algorithm converges after five generations.

Conclusions

One inevitable aspect of manufacturing systems is the production of defective products. In this paper, we developed a multi-product EPQ model with defective items and reworking. In addition, the warehouse space is limited for all products. In this condition, we formulated the problem as a non-linear programming, and in order to solve it, a much easier genetic algorithm was applied. At the end, we presented a numerical example to demonstrate the application of the proposed algorithm, and in this example, we optimized the parameters of the genetic algorithm. Also, for future research, some recommendations are presented as follows:

(a) Other heuristic search techniques such as ant colony optimization or simulated annealing algorithm can be used to solve the presented model and compare their results with the proposed genetic algorithm.
(b) In the future, researchers can add some limitations such as delay and shortage to the model. Aside from warehouse space constraint, they can consider constraints about budget and so on. Using some constraints together makes the problem too hard, and using a meta-heuristic approach to solve the model is inevitable.
(c) We can extend the model in such a way that some parameters such as the rate of demand become random or fuzzy variables.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

KFH participated in the design of the study and carried out the genetic algorithm studies. CV contributed in the problem definition, participated in the sequence alignment, and drafted the manuscript. AK participated in the development of the model. All authors read and approved the final manuscript.

Authors' information

Dr. KFH is an assistant professor in the Department of Industrial Management at Islamic Azad University, South Tehran Branch. His research interests lie in the area of project management, statistics, knowledge management, operations management, and operation research. For about 15 years, he has taught undergraduate, graduate, and industry courses and carried out research in various aspects of industrial engineering and management. He has published research papers in journals such as the Research Journal of Applied Sciences Engineering and Technology, World Applied Sciences Journal, American Journal of Scientific Research, International Journal of Economic and Management Science, and Middle-East Journal of Scientific Research. Dr. CV is an assistant professor in the Department of Industrial Management at Islamic Azad University, South Tehran Branch. His research interests lie in the area of quality and productivity management, strategic management, innovation management, knowledge management, and operations management. For about 15 years, he has taught undergraduate, graduate, and industry courses and carried out research in various aspects of industrial engineering and management. He has published research papers in journals such as The TQM Journal, Innovation: Management, Policy & Practice, International Journal of Productivity and Performance Management, Business Strategy Series, Industrial and Commercial Training, etc. He is a senior member of the American Society for Quality (ASQ). AK holds a MA degree in the field of Industrial Management from Islamic Azad University, Qazvin Branch.

Author details

[1]Department of Industrial Management, Islamic Azad University, South Tehran Branch, P.O. Box: 11365/4435, Tehran, Iran. [2]Department of Management/Accounting, Islamic Azad University, Qazvin Branch, P.O. Box: 34185–1416, Qazvin, Iran.

Received: 22 May 2010 Accepted: 3 March 2012
Published: 3 December 2012

References

Buscher U, Lindner G (2007) Optimizing a production system with rework and equal sized batch shipments. Comput Oper Res 34:515–535
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
Gen M (1997) Genetic algorithm and engineering design. Wiley, New York
Goldberg D (1989) Genetic algorithm in search, optimization and machine learning. Addison-Wesley, Reading
Hayek PA, Salameh MK (2001) Production lot sizing with the reworking of imperfect quality items produced. Prod Plann Control 12:584–590
Hou KL (2007) An EPQ model with set up cost and process quality as functions of capital expenditure. Appl Math Model 31:10–17
Kim CH, Hong Y (1999) An optimal production run length in deteriorating production processes. Int J Prod Econ 58:183–189
Lee HM, Yao JS (1998) Economic production quantity for fuzzy demand quantity and fuzzy production quantity. Eur J Oper Res 109:203–211
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
Rosenblatt MJ, Lee HL (1986) Economic production cycles with imperfect production processes. IIE Trans 18:48–55

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

Shabani A, Torabipour SM, Farzipour R (2011) Container selection in the presence of partial dual-core factors. Int J Phys Distrib Logist Manag 41(10):991–1008

Silver EA, Pyke DF, Peterson R (1998) Inventory management and production planning and scheduling, 3rd edn. Wiley, New York

doi:10.1186/2251-712X-8-27
Cite this article as: Hafshejani *et al.*: **Retracted: Using genetic algorithm approach to solve a multi-product EPQ model with defective items, rework, and constrained space.** *Journal of Industrial Engineering International* 2012 **8**:27.