

## An analysis of multi-item inventory model using particle swarm optimization under discrete delivery orders and limited storage space

Harun Ozturk\*<sup>1</sup>, Fatih Ahmet Senel<sup>2</sup>

Submitted: 09/11/2018 Accepted : 06/09/2019

**Abstract:** This study explores an economic production quantity (EPQ) model designed with the assumptions of discrete delivery orders and storage capacity constraints for a multi-item production inventory system. The main purpose of this study is to determine the optimal order quantity, the optimal number of deliveries and the optimal delivery quantity. First, the developed model as part of this study is analyzed using Genetic Algorithm (GA). Numerical analysis results are compared with those of previous studies and it was found that it is possible to have better results with an increasing number of iteration. The same model is then analyzed using Particle Swarm Optimization (PSO) algorithm. A comparison of the optimization methods showed that PSO gives better results over the GA under the same number of iterations and using the same population. The effects of important model parameters such as number of iterations, population, crossover, mutation rate on the optimal solution are analyzed. The results showed that PSO performs better than the GA with respect to the total cost and the total runtime as the solution of the problem in question.

**Keywords:** inventory, production, multi-item, optimization, genetic algorithm, particle swarm optimization

### 1. Introduction

Developed by Harris in 1913 [1], Economic Order Quantity (EOQ) model is the oldest model used in inventory management and production planning. This was followed by the classical Economic Production Quantity (EPQ) model developed by Taft in 1918 [2]. This inventory model was suggested for a single product, which is manufactured in a single-stage manufacturing system. Today, these inventory control models are used in several industries due to the ease-of-use and effectiveness they offer. Moreover, the main purpose of these models is to define overlapping costs which may include inventory holding cost and order cost, and to determine the optimal lot size which minimizes the total cost. However, these models are known to have a number of weaknesses as they are built on several assumptions. In this respect, both inventory models have been studied by many researchers under real-life conditions and mathematical models have been developed. Moreover, it is further suggested that the purchase of a high-speed machine with a high production capacity for multiple products is a rather common choice of the industry when compared to the purchase of a machine specialized only in a single product, as the former is the economical option. The purpose of these models is to determine the batch sizes and production sequence of the products which minimize the total cost.

Eilon [3] developed a multi-item inventory model in order to find the solution for the basic production problem of defining the quantity of each product or the lot size to be produced and its impact on the manufacturing cycle time. Rogers [4] suggested a computational approach for the purpose of finding the manufacturing program adding up to the minimal cost for multiple products manufactured in a single manufacturing system (or machine). Then, Bomberger [5] brought a dynamic programming

solution to the problem of multi-item manufacturing planning in a single manufacturing plant or a single machine using the assumption of a single type of product being manufactured at one moment in time, taking into consideration the setup cost and a setup time associated with producing each item over an infinite planning horizon. Madigan [6] suggested an alternative approach to the planning programs developed in other studies based on multi-item single-machine or single-system inventory problem under the assumption of long-term planning. Doll and Whybark [7] used an iterative procedure in order to find the optimal cycle time in a multi-item single-machine inventory problem for a single-item environment. Silver [8] proposed a simple method for determining the common cycle time and the order quantities in a multi-item single machine inventory model in which several items are packaged in different containers from the same batch supply. It was shown that this method gave nearly the optimal results when compared to other complex solutions; and optimal solutions were obtained for the case of two-item production in a custom lot. Elmaghraby [9] offered a review of the studies in which several models are developed with additional assumptions applied in the context of single-machine (or single-facility) multi-item production planning problem, and presented solutions to this problem using analytical methods and analyzed the feasibility of this environment using the model. Gupta and Kyparisis [10] studied the production planning for single-machine setting and provided recommendations for future research.

Gallego and Moon [11] analyzed the effect of reduced setup cost and setup time on the total cost in a multi-product single machine inventory problem. Arcade [12] suggested a number of solutions to the multi-item inventory problem with the purpose of producing several items in a single-machine or single-facility setting. Khouja et al. [13] applied a Genetic Algorithm (GA) to the solution suggested by Bomberger [5] for the inventory problem and they showed in their numerical analysis that GA offers better results when compared to dynamic programming approach. Moon et al.

<sup>1</sup>Süleyman Demirel University, Isparta, Turkey  
ORCID ID: 0000-0003-0193-6663

<sup>2</sup>Süleyman Demirel University, Isparta, Turkey,  
ORCID ID: 0000-0003-1918-7277

\*Corresponding Author Email: harunozturk@sdu.edu.tr

[14] suggested a hybrid genetic algorithm for the solution of the problem of single-machine (or single facility) multi-item production planning based on time-varying lot sizes. Kim et al. [15] developed a heuristic algorithm to define the optimal lot size, common production cycle time, number of deliveries and delivery quantity resulting in the minimal total cost in a multi-item single-machine (or single production system) inventory environment. Tang and Teunter [16] suggested a multi-item single-machine inventory problem for the performance of both manufacturing and reworking activities on a single production line.

Teunter et al. [17] used a number of heuristics methods to solve the same optimization problem. Both studies were based on a common cycle time policy with the assumption of single lot production in each cycle. Taleizadeh et al. [18] developed a multi-item EPQ model based on the assumptions of common cycle time and backordering in which the production defective-rate followed either a uniform or a normal probability distribution. Nevertheless, they assumed that the defective items are taken out of the inventory as scrap products at the end of screening process. Taleizadeh et al. [19] suggested an EPQ model for a multi-item inventory system based on the criteria of partial backordering and service level. They assumed that the defective items are discarded as scrap products at the end of the production rather than reworking. Taleizadeh et al. [19] developed a production inventory model which included defective products, such products are reworked or repaired, and among such reworked or repaired products were scrap products and the model allowed for partial backordering. The main purpose of this problem is to determine the optimal cycle time, the optimal production quantity, and the maximum level of backordering which results in the minimum total cost. Zanoni et al. [20] suggested an easy-to-use algorithm to solve the same problem, having relaxed the constraint of common cycle time and a single reworking lot for each item in each cycle. Taleizadeh et al. [21] developed an EPQ model with reworking in a single-stage production inventory system. Taleizadeh et al. [22], on the other hand, developed a production model using the assumptions of common cycle time, backordering, reworking or repair and the existence of scrap products in a defective multi-item production inventory system. The authors analyzed the changes in the results of the optimal solutions where defective-rate follows either a uniform or a normal probability distribution.

Other related papers including Chang et al. [23], Pal et al. [24], Mahata and Mahata [25], Nabil et al. [26] and Pasandideh et al. [27] considered some optimization methods for solving different inventory problems. Taleizadeh et al. [28] developed a multi-item single machine production inventory model with the assumptions of restricted common cycle time and remanufacturing, backordering and machine failure. The authors analyzed two conditions where the failure in the production process emerges during the backordering elimination and after backordering is eliminated. Đorđević et al. [29] considered a deterministic multi-product EOQ inventory problem with the storage space constraints. They modelled the problem as a combinatorial optimization problem, and developed two heuristic methods to solve this problem approximately. Mokhtari [30] suggested a production inventory model for defective items in which all defective items are reworked to make them as-good-as-perfect. Ant Colony Optimization (ACO) and the GA were used to determine the optimal production and order lot sizes which minimize total cost.

In recent years, many efficient methods for determining the optimal solution to the multi-item multi constraints lot-sizing problem exist in the inventory literature. Pasandideh and Niaki

[31] explored the assumptions of warehouse space limitation, discrete delivery policy, and equal delivery quantities in a multi-item inventory problem. This problem is formulated as a nonlinear integer programming model and a GA method is suggested in order to solve this model. Numerical examples were given in order to define the optimal number of deliveries and the product quantity in each delivery of five different items to show the feasibility of the suggested algorithm and to assess its performance. The authors, having assigned 8, 0.85, and 0.25 for the parameters of population size ( $N$ ), crossover rate ( $P_c$ ) and mutation rate ( $P_m$ ), presented the solution values obtained after ten iterations. A closer look into the solution values obtained using the same algorithm with increased number of iterations showed that the total cost is around \$5800 after the 10th iteration. In other words, this study does not present any clear information on the optimal solution results obtained for the number of deliveries and the quantities delivered using the developed algorithm. It can be seen in the diagram included to the paper that increased number of iterations results in reduced cost and the resulting cost was approximately \$5500. Nabil et al. [32] modified the model which has some shortcomings in the paper of Pasandideh and Niaki [31]. They have solved the problem in a shorter period of time with the heuristic algorithm they developed because the problem has less constraints and decision variables. The numerical results have shown that better optimal values can be achieved.

This paper investigates the work of Pasandideh and Niaki [31] in a lot sizing decision problem, and provides the new solutions to the numerical example. The optimal number of deliveries and optimal delivery quantities are re-determined by using the GA for the same values of the parameters of  $N$ ,  $P_c$  and  $P_m$ . From the optimal results, it was observed that the total cost approaches to a smaller optimal value with an increasing number of iteration. The effects of important parameters of the model on the results of the optimal solution are analyzed using sensitivity analysis. Then, the problem is solved using the Particle Swarm Optimization (PSO) algorithm; optimal number of deliveries and optimal delivery quantity are obtained; and the effects of important parameters on the optimal solution results are also analyzed. Finally, the optimization methods are compared based on the results of the optimal solution, and to do this, the same values are assigned to the relevant parameters of both methods.

The rest of the paper is organized as follows. The next section describes the production inventory model developed by Pasandideh and Niaki [31]. Section 3 presents the solution method for the problem. Section 4 offers a numerical solution of the model using the PSO and GA methods, and a comparison of these methods along with managerial insight is given. Section 5 concludes the paper.

## 2. The model of Pasandideh and Niaki

The following assumptions are considered in the development of a mathematical model for the inventory problem where buyer and supplier interact:

- The production rate is known and constant.
- The demand of each item is known and constant.
- Supplier delivers the orders in multiple shipments.
- Buyer covers the costs arising from each shipment.
- Number of shipments and the delivery quantity of each shipment are defined by the buyer.
- The delivery quantity of each shipment is equal.
- Buyer has a limited warehouse space.
- Setup cost and holding cost are known.
- Backordering and delayed payments are allowed.

For  $i = 1, \dots, n$ , the following parameters are used to develop the model:

- $n$  the number of items
- $Q_i$  the order quantity for item  $i$
- $P_i$  the production rate for item  $i$
- $D_i$  the demand rate for item  $i$
- $T_i$  the cycle time for item  $i$
- $T_{pi}$  the production time in each cycle of item  $i$
- $T_{di}$  the production downtime in each cycle of item  $i$
- $t_i$   $i^{\text{th}}$  time between two consecutive shipments of the product
- $k_i$   $i^{\text{th}}$  delivery quantity of each shipment of the product
- $m_i$   $i^{\text{th}}$  number of shipments in each cycle of the product
- $U_i$   $i^{\text{th}}$  max. number of shipments in each cycle of the product
- $L_i$   $i^{\text{th}}$  min. number of shipments in each cycle of the product
- $f_i$   $i^{\text{th}}$  the space occupied by each unit of the product
- $f$  available warehouse space for all products
- $b_i$   $i^{\text{th}}$  cost of each shipment of the product
- $A_i$   $i^{\text{th}}$  setup cost associated with each cycle of the product
- $h_i$   $i^{\text{th}}$  holding cost associated with each unit of the product
- $c_i$   $i^{\text{th}}$  reserve cost associated with each unit of the product

Fig. 1 presents the inventory problem in a diagram. Order quantity,  $Q_i$ , of each product will be delivered with a number of  $k_i$  shipments with the same delivery quantity of  $m_i$ .

$$Q_i = m_i k_i; i = 1, \dots, n. \quad (1)$$

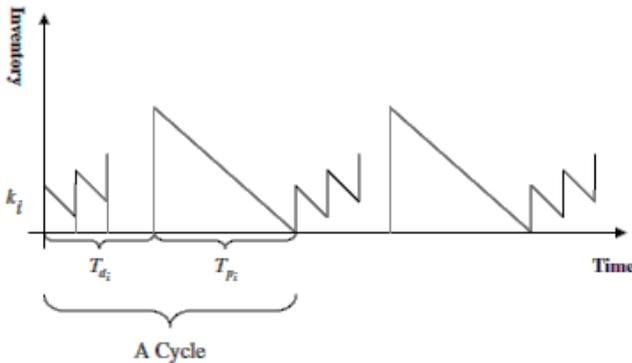


Fig. 1 The behavior of inventory level over time

In Fig. 1, cycle time,  $T_i$ , is the function of the delivery time of the orders,  $T_{di}$ , and the production time,  $T_{pi}$ .

$$T_i = T_{di} + T_{pi} = \frac{Q_i}{D_i}; i = 1, \dots, n. \quad (2)$$

Delivery time of the orders,  $T_{di}$ , involve the time between two consecutive shipments,  $t_i$ , and it is calculated as follows:

$$t_i = \frac{k_i}{P_i}; i = 1, \dots, n. \quad (3)$$

Total cost is the sum of setup cost, reserve cost, shipment cost and holding cost. Using the abovementioned assumptions, Pasandideh and Niaki [31] derived the total cost in unit time,  $TCU$ , as follows:

$$TCU = \sum_{i=1}^n \left[ \frac{A_i D_i}{Q_i} + c_i D_i + \frac{b_i D_i}{k_i} + \frac{h_i}{2} \left( Q_i - (Q_i - k_i) \frac{D_i}{P_i} \right) \right] \quad (4)$$

## 2.1. The constraint

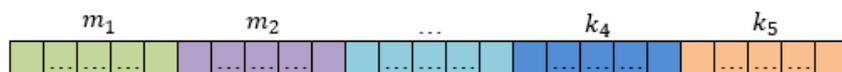


Fig. 2 Chromosome structure

The purpose of this problem is to find integer values for batch quantity, shipment capacity and the number of shipments which will result in the minimal total cost function as given in Equation (4). This problem involves two constraints.

- a) The warehouse space necessary for product storage is limited.
- b) The number of deliveries must be in the range between permitted minimum and maximum values.

Here, the problem can be formulated as follows, using the total cost function available in Equation (4) and the constraints defined above:

$$TCU = \sum_{i=1}^n \left[ \frac{A_i D_i}{Q_i} + c_i D_i + \frac{b_i D_i}{k_i} + \frac{h_i}{2} \left( Q_i - (Q_i - k_i) \frac{D_i}{P_i} \right) \right] \quad (5)$$

s.t.:

$$\sum_{i=1}^n f_i Q_i \leq f, \quad (6)$$

$$L_i \leq m_i \leq U_i; \quad (7)$$

$$Q_i = m_i k_i; i = 1, \dots, n. \quad (8)$$

$$m_i, k_i, Q_i: \text{integer.} \quad (9)$$

As it can be seen from Equation (5), the problem cannot be solved analytically. For this reason, the inventory control problem is a NP-hard problem.

## 3. Solution method

### 3.1. Genetic Algorithm

Genetic Algorithm (GA) is an optimization method which was developed by Holland [33] inspired by Darwin's theory of natural selection. Based on the natural selection process of life, the GA performs a query. In the GA, members of the living population are expressed in terms of their chromosomal structure. It can be expressed using chromosomal structures such as binary encoding, permutation encoding, value encoding, etc. Chromosomal structure, in this study, was represented using binary coding. Each member of the population, i.e. each chromosome, consists of variables which are to be optimized. This study aimed to optimize five values of  $m$  and  $k$ , each. Thus, the chromosome structure is built in a way to express five  $m$  and another five  $k$  values (Fig. 2). Definition of initial population

Initial population of the GA is created randomly. First, the user defines the necessary number of members of population. Then, chromosomes are randomly formed in a binary structure to express all the variables.

#### 3.1.1. Crossover

In the GA, crossover is the process of creating better individuals, having selected the individuals from a pool of good individuals. Crossover can be performed from one point or two points. This study used one-point crossover process. The new individuals created as a result of random one-point crossover of the two chromosome samples are shown in Fig. 3a and Fig. 3b.

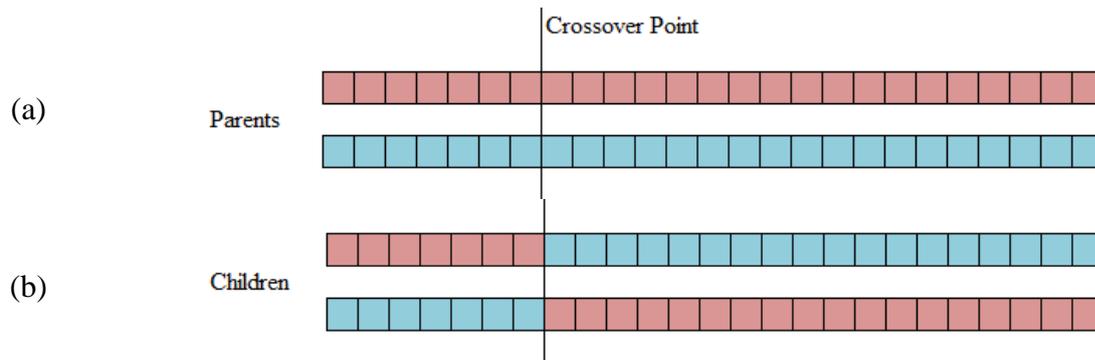


Fig. 3 Crossover sample (a) Parents before crossover (b) Children after crossover

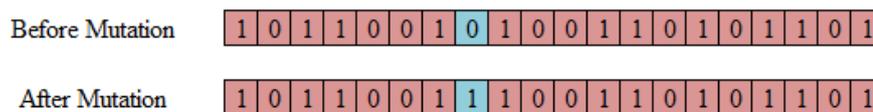


Fig. 4 Mutation of a randomly selected bit

### 3.1.2. Mutation

Mutation is used to prevent GA from converging on local minima. Mutation of the chromosomes with binary encoding is conducted with the reversal of a randomly selected bit. In other words, the value of '1' is assigned to a randomly selected bit if it was originally '0' and vice versa otherwise. Fig. 4 shows an example of the mutation process.

### 3.1.3. Selection

One of the most important steps of the GA is selection. In this step, members of the former generation are selected to be transferred to the next generation as a new generation is created. As the diversity of the population will suffer when only the good candidates are transferred, good candidates must be transferred to the new generation with a specific probability. It is necessary to keep the probability of transferring good candidates high. Nevertheless, it must be ensured that bad candidates are also transferred, but with a lower probability. The methods which are tournament selection, roulette wheel selection, random selection, etc. are used for this purpose. Tournament selection was the method used in this study. The tournament selection method involves a random selection of  $s$  candidates for a tournament against each other. Then, the best candidates from these matches are transferred to the next generation. This process continues until the number of individuals as defined by the user is achieved in the new generation.

### 3.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a method developed by Kennedy and Eberhart in 1995 based on the hunting behavior of swarms of birds and fish [34]. In the PSO algorithm, first the number of particles in the swarm is calculated and the initial population is created randomly. The best values and location information of each particle and the swarm as a whole are recorded. Each particle in the search space updates its location to the global best of the swarm. Particles calculate their location and velocity according to Equations 10 and 11.

$$\bar{x}_{n+1}^i = \bar{x}_n^i + \bar{v}_{n+1}^i \quad (10)$$

$$\bar{v}_{n+1}^i = \omega \bar{v}_n^i + c_1 r_1 (\bar{p}_n^i - \bar{x}_n^i) + c_2 r_2 (\bar{p}_n^g - \bar{x}_n^i) \quad (11)$$

Where,  $i$  is the  $i$ th particle in the swarm,  $\omega$  is the inertia weight parameter,  $n$  is the number of iterations,  $r_1$  and  $r_2$  are random

number in the range of  $[0,1]$ .  $\bar{x}$  is the location factor,  $\bar{v}$  is the velocity factor,  $\bar{p}^i$  is the best location found by the  $i$ th particle,  $\bar{p}^g$  is the best location found by the swarm. The flowchart of PSO algorithm is shown in Fig. 5.

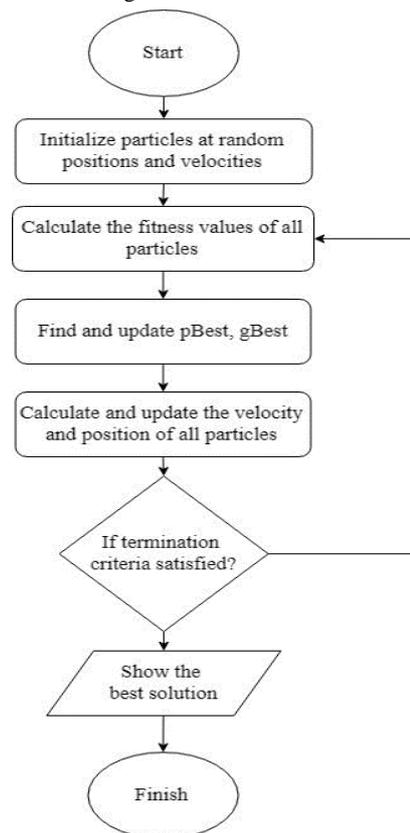


Fig. 5 Flowchart of the PSO

## 4. Numerical Example and Sensitivity Analysis

The numerical example was taken from the study proposed by Pasandideh and Niaki [31]. Table 1 shows the parameters necessary for the 5-item production inventory problem. In this example, it is assumed that  $L_i$  and  $U_i$  are the same for all of the five products where  $L_i = 5$  and  $U_i = 35$ , and  $f = 7900$ .

The multi-item inventory control problem was solved using the GA and PSO metaheuristic optimization methods. As it was the case in Pasandideh and Niaki [31], the values assigned for  $N, P_c$  and  $P_M$  parameters were 8, 0.85 and 0.25 respectively, and the solution results obtained at the 10th iteration are given in Table 2.

**Table 1.** General data for the example

Product	$D_i$	$P_i$	$c_i$	$A_i$	$b_i$	$h_i$	$f_i$
1	21	66	19	30	6	4	5
2	18	57	23	88	2	9	8
3	27	71	37	71	9	7	4
4	16	29	14	63	4	9	3
5	19	99	24	44	5	4	9

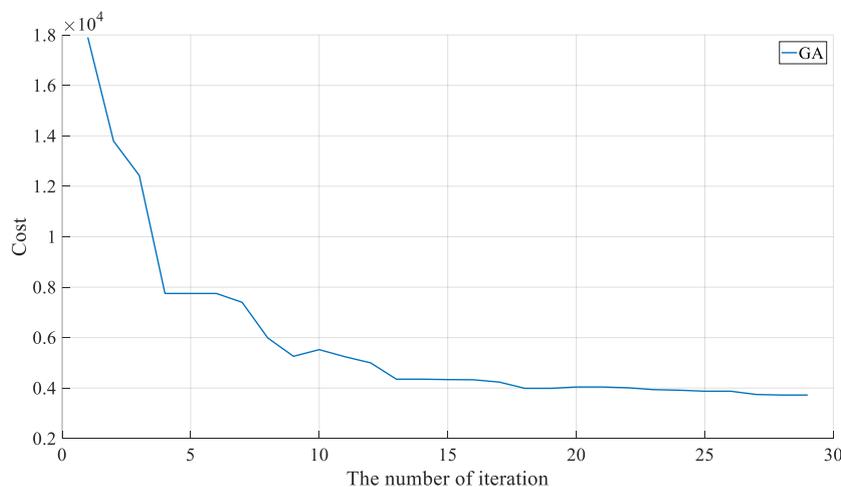
**Table 2.** The results for GA method after 10 iterations

$P_M$	$P_c$	$n$	$N$	$m_1$	$m_2$	$m_3$	$m_4$
0.25	0.85	10	8	6	28	23	12
$m_5$	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$f$	Time(ms)
6	89	9	5	6	12	4725.04680	334.78

It was found from the comparison of the values assigned to the parameters in question for the GA method against those of the literature reports that selection of higher values such as 0.25 instead of 0.05 for mutation rate makes it harder for the GA method to function. The reason behind this is the fact that mutation is used to prevent the GA from converging on local minima. When assigned a greater value, mutation rate will lead to an inconsistency in the population which in return will make it harder to reach at a global minimum. Thus, mutation rate must be selected from the range between 0.01 and 0.05 when working with GA. A closer look into the Fig. 6 included in study published by Pasandideh and Niaki [31] showed that better results are obtained with increasing number of iterations. Therefore, optimal solution result is given in Table 3 as it was obtained after 30 iterations. Moreover, the total cost function which was obtained after 150 iterations where all the other parameters were kept constant is also given in Table 3. Figs. 6-7 illustrate these changes graphically, respectively.

**Table 3.** The results for the GA method after 30 and 150 iterations

$n$	$P_M$	$P_c$	$N$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
30	0.25	0.85	8	6	6	15	9	13
	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$f$	Time(ms)	
	13	4	16	2	11	3723.11041	17.05	
150	0.25	0.85	8	5	5	20	6	5
	50	6	2	4	6	3499.14225	48.13	



**Fig. 6** Change in total cost values through 30 iterations

For the case where the number of iterations and population were the same, i.e.  $n = 30, N = 8$ , optimal delivery quantities and number of deliveries and optimal total cost were obtained using the PSO method. And results are shown in Table 4. When compared to the results obtained using the GA method, Table 4 shows that 14.8823% decrease in the total cost is achieved. Although there were no statistically significant differences, it was found that PSO method gave better results when the solution results obtained using both methods were compared. In order to obtain better results using the GA method, one will be required to invest more time to the process as the number of iterations or population is needed to be increased. This finding will be further discussed in the next section using the sensitivity analysis.

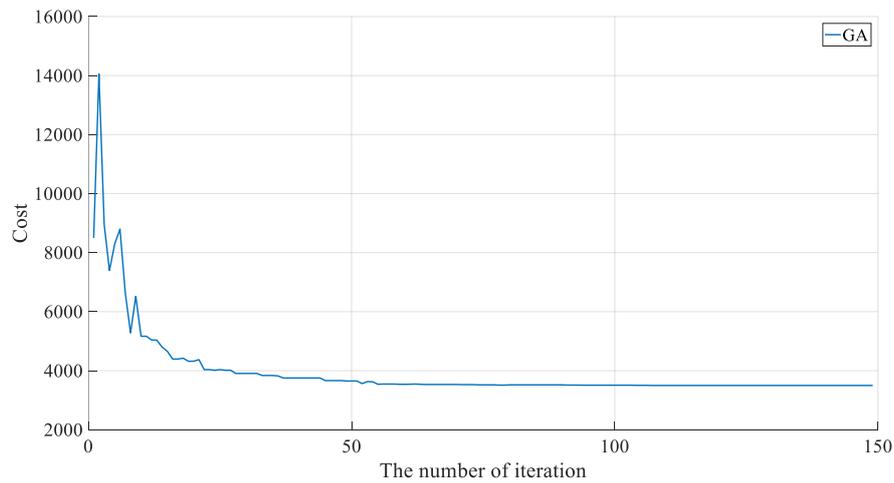
**Table 4.** The results for PSO method

$n$	$N$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
30	8	7	5	5	5	5
$k_2$	$k_3$	$k_4$	$k_5$	$f$	Time(ms)	
5	5	11	9	5	3169.02515	

The next section explores the effects of parameters of the model on the optimal solution results using sensitivity analysis.

#### 4.1. Sensitivity Analysis

Sensitivity analyses are used in this section to show that the problem in question is feasible and to present the effect of the parameters on the optimal solution results. The problem was solved using the GA and PSO optimization methods and the cost function given in Section 3. First, the problem was solved using the GA method for the conditions where the population was 30 and the number of iterations was increased to 600 and 1000 with the increase in the number of iterations and the results for the optimal solution are given in Table 5. These results were then compared based on the value of the total cost function and the time required for computation. Table 5 shows that the total cost adds up to 3129,20509 when the number of iterations was 600 and that this result was computed in 179.48 milliseconds. Moreover, the GA method was repeatedly used up to the 1000th iteration and it was found that the total cost was not altered.



**Fig. 7** Change in total cost values through 150 iterations

**Table 5.** The sensitivity analysis of the GA method with different iteration numbers

	$P_M$	$P_c$	$n$	$N$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$f$	Time (ms)
	0.25	0.85	30	8	6	6	15	9	13	13	4	16	2	11	3723.11041	17.05
GA	0.25	0.85	600	8	5	6	6	5	5	5	4	5	5	5	3129.20509	179.48
	0.25	0.85	1000	8	5	5	6	5	5	5	5	5	5	6	3128.98973	286.76

**Table 6.** The sensitivity analysis of the GA method with different iteration and population numbers

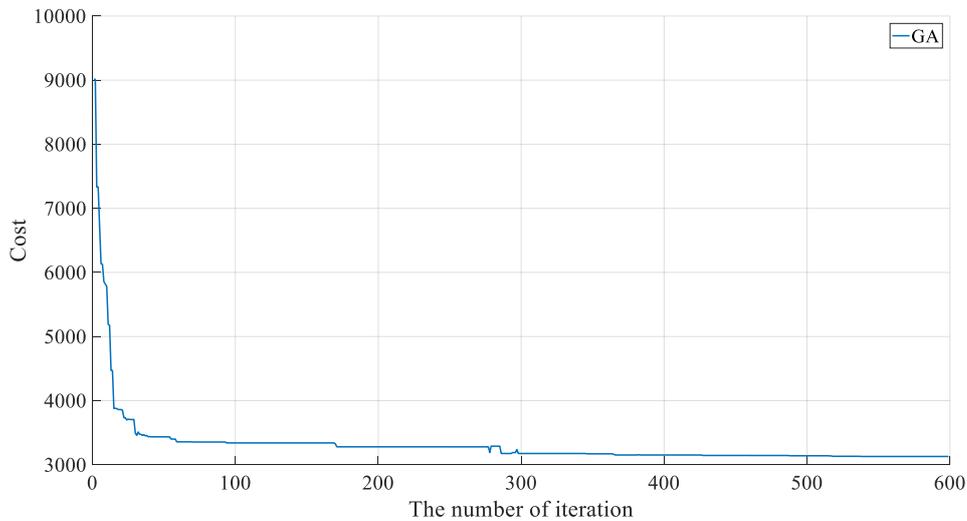
	$P_M$	$P_c$	$n$	$N$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$f$	Time (ms)
	0.25	0.85	50	10	7	5	6	5	5	4	12	5	12	11	3292.05049	10.06
	0.25	0.85	50	30	5	12	7	23	5	9	2	5	1	4	3189.88359	17.04
GA	0.25	0.85	50	50	5	6	5	5	5	7	4	9	6	6	3128.11361	28.07
	0.25	0.85	100	10	5	6	9	5	5	13	3	5	6	5	3177.88217	14.04
	0.25	0.85	100	30	5	5	6	6	5	5	6	5	4	12	3163.36340	33.09
	0.25	0.85	100	50	5	5	5	6	5	12	5	7	4	12	3171.58219	47.13

**Table 7.** The sensitivity analysis of the GA method with different parameter values

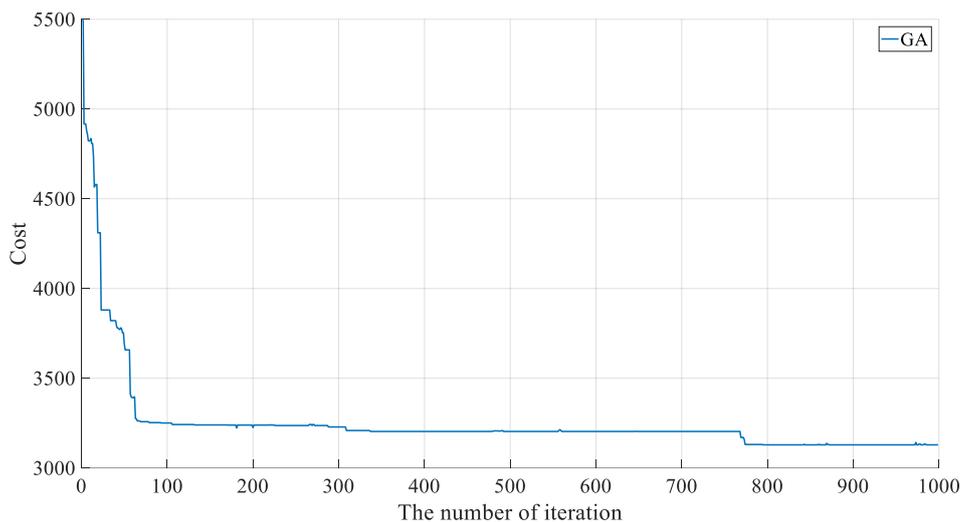
	$P_M$	$P_c$	$n$	$N$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$f$	Time (ms)
	0.01	0.75	100	50	5	12	5	5	14	5	6	7	9	2	3278.61051	51.14
	0.05	0.75	100	50	5	8	7	5	5	12	3	5	5	5	3155.28235	49.12
	0.01	0.66	100	50	11	13	5	8	5	5	2	19	3	8	3252.65524	49.13
	0.05	0.66	100	50	5	5	5	5	5	5	6	25	6	6	3282.99321	49.17
GA	0.01	0.85	100	50	5	11	20	8	11	6	2	2	3	2	3239.17239	52.17
	0.05	0.85	100	50	5	6	7	12	5	5	4	5	2	6	3142.06072	51.16
	0.01	0.66	500	100	5	20	5	6	5	6	2	8	4	6	3148.86843	433.17
	0.05	0.75	500	100	5	6	6	5	5	12	4	5	5	5	3151.98691	445.18
	0.1	0.75	500	100	5	12	6	5	5	5	2	5	5	5	3135.36298	434.17
	0.1	0.66	500	100	5	5	6	11	5	5	6	5	2	5	3145.51906	432.15

Fig. 8 shows that optimal solution was yet to be achieved when the number of iterations was 600. Accordingly, Fig. 9 shows the best result achieved with the GA methods when the number of iterations was 1000. Now, the effects of changes in the parameters of inventory problem on the results of the optimal solution will be discussed. Tables 6 and 7 show the values obtained as a result of the changes in the parameters when using the GA method. Table 6 shows that the total cost is decreased when the number of iterations is increased under the assumption of mutation and crossover rates and population were constant. However, it was found that the total cost is increased when the population was 50. Table 6 further shows that the total cost is decreased when the population is increased under the assumption of mutation and crossover rates and the number of iterations were constant. However, it was observed that the total cost is increased when the number of

iterations was 100 and when population was increased from 30 to 50. This is a direct result of the failure to assign suitable values to the parameters in the GA method. Table 7 shows the literature findings reported for the parameters used for the GA method. A closer look into the parameters associated with optimal values of total cost in Table 7, it can be concluded that a rather great mutation rate (0.1) needs to be used for the GA method. Such a parameter value does not capture the logic of the GA method. In other words, in order to be able to solve this problem using the GA method, one needs to assign illogical values to the parameters from the GA method point of view. Thus, it will be safe to say that GA method is not suitable for the solution of the inventory problem.



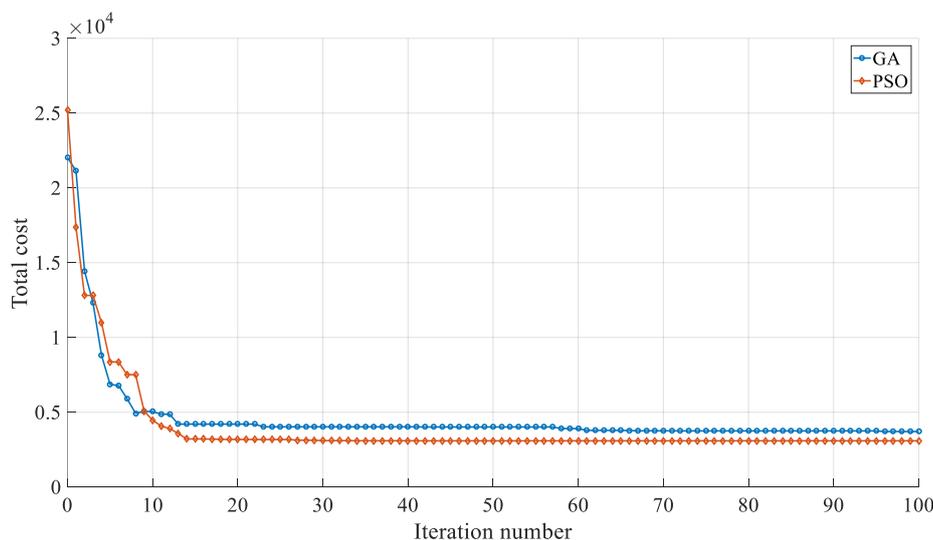
**Fig. 8** Change in total cost values through 600 iterations



**Fig. 9** Change in total cost values through 1000 iterations

**Table 8.** The sensitivity analysis of the PSO method with different parameter values

	$n$	$N$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$f$	Time (ms)
PSO	50	10	5	5	5	5	5	6	5	6	5	8	3127.27956	10.03
	50	30	5	6	5	5	5	6	4	7	5	6	3118.47704	12.03
	50	50	5	5	5	5	5	6	5	7	5	6	3118.53704	14.04
	100	10	5	5	5	5	5	6	5	7	5	6	3118.53704	7.02
	100	30	5	6	5	5	5	6	4	7	5	6	3118.47704	15.04
	100	50	5	5	5	5	5	6	5	7	5	6	3118.53704	28.07



**Fig. 10** The comparison between the GA and the PSO methods

Table 8 shows the effects of different number of iterations and population on the optimal solution for the solution of optimization problem using the PSO method. Table 8 shows that increasing the number of iterations and population has an almost insignificant effect on the total cost. A brief reading of the results showed that PSO method was able to solve the problem with the use of minimum parameter values and that the optimal result did not differ even when these values were increased. When the results are analysed, it was observed that PSO method is much more convenient in the solution of this problem when compared to the GA method. It was rather easy for the PSO method to obtain values, which cannot be obtained using the GA method, and this was shown with results. Furthermore, it can be safe to say that PSO method is much more advantageous in terms of runtime when compared to the GA method. Fig. 10 shows the total cost function values obtained from both the GA and the PSO methods as the number of iterations is increased.

## Conclusions

The classical EPQ model is developed in a way to minimize the total cost under the assumptions of single-item production, infinite warehouse area, and each order is received in a single delivery. With the advancements in today's technology, production processes are also developed which made it possible for multi-item production in addition to single-item production. With the widespread application of multi-item production systems, new problems emerged with regards to the delivery of the orders. The

number of shipments needed for the delivery of orders and the amount of products included in a single delivery affect the total cost. Thus, delivery of the order is as important as the manufacturing process and the delivery process has become an important aspect of the operations of a manufacturing plant which require careful planning. The production inventory model where each order is received in a single delivery and each shipment included the same amount of products was optimized using the GA and PSO methods under limited warehouse space. The purpose of this study was to define the optimal number of deliveries and the product quantity shipped in each delivery, i.e. the optimal order quantity, in order to minimize the total cost.

The solution of this problem was presented with a numerical example and the effects of important model parameters used in the optimization methods such as population, number of iterations, crossover and mutation rates were explored using sensitivity analyses. The results of the numerical analysis, when compared to the study which offered an approximate solution for the same problem with up to 30 iterations, showed that increasing the number of iterations allowed for an almost optimal solution of the problem. Using the sensitivity analysis, it was found that better results were obtained with the assignment of suitable values to the parameters of the GA method, that the total cost function was improved, i.e. that it was reduced.

Nevertheless, the same problem was solved using PSO and the results of optimal solution were explored using sensitivity analyses. A comparison of the results of optimal solutions showed that PSO method is able to offer better results in a shorter period of time when compared to the GA method. It was observed with

the sensitivity analyses that PSO method allows for the solution of the problem with reduced number of iterations and reduced population. From the results, it was concluded that PSO method is much more convenient for the solution of the problem explored in this study when compared to the GA method. Moreover, the same results were obtained with less computation as in Nobil et al. [32].

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## References

- [1] F. W. Harris, "How Many Parts to Make at Once," *Factory, Mag. Manag.*, vol. 10, no. 2, p. 135–136, 1913.
- [2] E. W. Taft, "The most economical production lot.," *Iron Age*, vol. 101, no. 18, pp. 1410–1412, 1918.
- [3] S. Eilon, "Scheduling for batch production," *Inst. Prod. Eng. J.*, vol. 36, no. 9, pp. 549–570, 1957.
- [4] J. Rogers, "A Computational Approach to the Economic Lot Scheduling Problem," *Manage. Sci.*, vol. 4, no. 3, pp. 264–291, 1958.
- [5] E. E. Bomberger, "A Dynamic Programming Approach to a Lot Size Scheduling Problem," *Manage. Sci.*, vol. 12, no. 11, pp. 778–784, 1966.
- [6] J. G. Madigan, "Scheduling a Multi-Product Single Machine System for an Infinite Planning Period," *Management Science*, vol. 14, INFORMS, pp. 713–719, 1968.
- [7] C. L. Doll and D. C. Whybark, "An Iterative Procedure for the Single-Machine Multi-Product Lot Scheduling Problem," *Management Science*, vol. 20, INFORMS, pp. 50–55, 1973.
- [8] E. A. Silver, "A Simple Method of Determining Order Quantities in Joint Replenishments under Deterministic Demand," *Management Science*, vol. 22, INFORMS, pp. 1351–1361, 1976.
- [9] S. E. Elmaghraby, "The Economic Lot Scheduling Problem (ELSP): Review and Extensions," *Management Science*, vol. 24, INFORMS, pp. 587–598, 1978.
- [10] S. K. Gupta and J. Kyparisis, "Single machine scheduling research," *Omega*, vol. 15, no. 3, pp. 207–227, Jan. 1987.
- [11] G. Gallego and I. Moon, "The Effect of Externalizing Setups in the Economic Lot Scheduling Problem," *Oper. Res.*, vol. 40, no. 3, pp. 614–619, Jun. 1992.
- [12] S. Arcade, "Single machine multi-product batch scheduling: Testing several solution methods," *Omega*, vol. 21, no. 6, pp. 709–711, Nov. 1993.
- [13] M. Khouja, Z. Michalewicz, and M. Wilmot, "The use of genetic algorithms to solve the economic lot size scheduling problem," *Eur. J. Oper. Res.*, vol. 110, no. 3, pp. 509–524, Nov. 1998.
- [14] I. Moon, E. A. Silver, and S. Choi, "Hybrid genetic algorithm for the economic lot-scheduling problem," *Int. J. Prod. Res.*, vol. 40, no. 4, pp. 809–824, 2002.
- [15] T. Kim, Y. Hong, and S. Y. Chang, "Joint economic procurement—production–delivery policy for multiple items in a single-manufacturer, multiple-retailer system," *Int. J. Prod. Econ.*, vol. 103, no. 1, pp. 199–208, Sep. 2006.
- [16] O. Tang and R. Teunter, "Economic Lot Scheduling Problem with Returns.," *Prod. Oper. Manag.*, vol. 15, no. 4, p. 488, 2006.
- [17] R. Teunter, O. Tang, and K. Kaparis, "Heuristics for the economic lot scheduling problem with returns," *Int. J. Prod. Econ.*, vol. 118, no. 1, pp. 323–330, Mar. 2009.
- [18] A. Taleizadeh, A. A. Najafi, and S. T. Akhavan Niaki, "Economic Production Quantity model with scrapped items and limited production capacity," *Sci. Iran.*, vol. 17, no. 1 E, pp. 58–69, 2010.
- [19] A. A. Taleizadeh, H. M. Wee, and S. J. Sadjadi, "Multi-product production quantity model with repair failure and partial backordering," *Comput. Ind. Eng.*, vol. 59, no. 1, pp. 45–54, 2010.
- [20] S. Zaroni, A. Segerstedt, O. Tang, and L. Mazzoldi, "Multi-product economic lot scheduling problem with manufacturing and remanufacturing using a basic period policy," *Comput. Ind. Eng.*, vol. 62, no. 4, pp. 1025–1033, May 2012.
- [21] A. A. Taleizadeh, L. E. Cárdenas-Barrón, J. Biabani, and R. Nikousokhan, "Multi products single machine EPQ model with immediate rework process," *Int. J. Ind. Eng. Comput.*, vol. 3, no. 2, pp. 93–102, 2012.
- [22] A. A. Taleizadeh, S. G. Jalali-Naini, H. M. Wee, and T. C. Kuo, "An imperfect multi-product production system with rework," *Sci. Iran.*, vol. 20, no. 3, pp. 811–823, 2013.
- [23] H.-C. Chang, L.-Y. Ouyang, K.-S. Wu, and C.-H. Ho, "Integrated vendor–buyer cooperative inventory models with controllable lead time and ordering cost reduction," *Eur. J. Oper. Res.*, vol. 170, no. 2, pp. 481–495, Apr. 2006.
- [24] S. Pal, M. K. Maiti, and M. Maiti, "An EPQ model with price discounted promotional demand in an imprecise planning horizon via Genetic Algorithm," *Comput. Ind. Eng.*, vol. 57, no. 1, pp. 181–187, 2009.
- [25] G. C. Mahata and P. Mahata, "Analysis of a fuzzy economic order quantity model for deteriorating items under retailer partial trade credit financing in a supply chain," *Math. Comput. Model.*, vol. 53, no. 9–10, pp. 1621–1636, 2011.
- [26] A. H. Nobil, A. H. A. Sedigh, and L. E. Cárdenas-Barrón, "A multiproduct single machine economic production quantity (EPQ) inventory model with discrete delivery order, joint production policy and budget constraints," *Ann. Oper. Res.*, pp. 1–37, 2017.
- [27] S. H. R. Pasandideh, S. T. A. Niaki, and M. H. Far, "A multiproduct EOQ model with permissible delay in payments and shortage within warehouse space constraint: a genetic algorithm approach," *Int. J. Math. Oper. Res.*, vol. 10, no. 3, p. 316, 2017.
- [28] A. A. Taleizadeh, L. E. Cárdenas-Barrón, and B. Mohammadi, "A deterministic multi product single machine EPQ model with backordering, scraped products, rework and interruption in manufacturing process," *Int. J. Prod. Econ.*, vol. 150, pp. 9–27, Apr. 2014.
- [29] L. Đorđević, S. Antić, M. Čangalović, and A. Lisec, "A metaheuristic approach to solving a multiproduct EOQ-based inventory problem with storage space constraints," *Optim. Lett.*, vol. 11, no. 6, pp. 1137–1154, 2017.
- [30] H. Mokhtari, "A joint internal production and external supplier order lot size optimization under defective manufacturing and rework," *Int. J. Adv. Manuf. Technol.*, pp. 1–20, 2017.
- [31] S. H. R. Pasandideh and S. T. A. Niaki, "A genetic algorithm approach to optimize a multi-products EPQ model with discrete delivery orders and constrained space," *Appl. Math. Comput.*, vol. 195, no. 2, pp. 506–514, 2008.
- [32] A. H. Nobil, S. T. A. Niaki, and E. Nobil, "An effective and simple algorithm to solve the discrete multi-product economic production quantity model," *Econ. Comput. Econ. Cybern. Stud. Res.*, vol. 51, no. 3, pp. 251–261, 2017.
- [33] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [34] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948.