Applied Mathematical Modelling 39 (2015) 5011–5032

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

GPSO-LS algorithm for a multi-item EPQ model with production capacity restriction

Mohammadali Pirayesh *, Saeed Poormoaied

Department of Industrial Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

ARTICLE INFO

Article history:
Received 28 May 2013
Received in revised form 5 March 2015
Accepted 25 March 2015
Available online 24 April 2015

Keywords:
Economic production quantity
Multi-items
Particle swarm optimization
Genetic algorithm

ABSTRACT

In this research a multi-item economic production quantity (EPQ) model with a single machine is investigated. It is assumed that the production capacity of the machine is limited, with no shortages allowed. The model formulated in this study has been developed such that the objective function is to minimize the total inventory cost where the optimal order and production quantities for each item are the decision variables. In this research a hybrid algorithm hereby called GPSO-LS is proposed to find a near-optimal solution. The proposed algorithm is based on genetic algorithm and particle swarm optimization. In this context, the Taguchi method is used to tune the parameters of the algorithm. Lower and upper bounds for the optimal value of the objective function have been developed in order to measure the quality of the solutions provided by GPSO-LS. Numerical results obtained show the effectiveness of the proposed GPSO-LS and the features of the presented model. A main finding of this study is that increasing the production rate and/or decreasing the demand rate of items reduces the total inventory cost. This finding supports managers in making decisions such as investment in increasing production capacity, resorting to external sources, or incurring lost sales cost.

1. Introduction

In recent years, competitive markets have forced manufacturers to reduce their operational costs. Inventory, a main contributor to the operational cost, consists of the goods held in a warehouse to satisfy customer demand. Inventory is necessary to buffer the supply and demand [1]. Inventory control determines when and how much of the goods should be replenished in order to minimize the costs associated with holding stock, running out of stock, and placing orders. Inventory control is a problem common to all enterprises.

The economic order quantity (EOQ) model is undoubtedly one of the oldest models in the inventory literature [2]. The economic production quantity (EPQ) model proposed by Hadley and Whitin [3] is an extension of the well-known EOQ model where an item is produced by a single machine. The EPQ and EOQ models are still widely accepted by many industries today for their simplicity and effectiveness [4]. In spite of their wide acceptance, some practitioners and researchers have questioned the practical applications of the EOQ and EPQ model. Woolsey [5] severely criticized the use of the EOQ model, arguing that the assumptions do not meet the real world situations; see Jaber et al. [6] for a concise review of these critiques. One of these assumptions is that the EPQ has sufficient production capacity to satisfy demand. However, the production capacity may not be enough to meet the demands of all items in a multi-item situation.

* Corresponding author at: Azadi Sq. Ferdowsi University of Mashhad, Mashhad, Iran. Tel.: +98 511 8805027; fax: +98 511 8796778.
E-mail address: pirayesh@um.ac.ir (M. Pirayesh).

http://dx.doi.org/10.1016/j.apm.2015.03.047
0307-904X/

A considerable amount of research has been carried out to address the multi-item EPQ model. Byrne [7] analyzed a multi-item EPQ model using a search simulation approach. In his model, shortages may occur and unfilled demand is backordered. Hwang et al. [8] considered the advantages of setup time reduction and quality improvements in multi-product capacitated EPQ models. When the setup time is reduced, more time can be devoted to production, especially when the production time is restricted. It is clear that capital investments are needed for the setup reduction and quality improvements. So, they developed a model to determine when and how much to invest to maximize the system's profit. Jaber and Bonney [9] surveyed the setup reduction work from a learning point of view. Fransoo et al. [10] proposed a hierarchical approach for production capacity coordination in a multi-item production system. In this approach, they allocated capacity to individual products for maximizing the expected profit. Demand that is not met using the existing inventory is lost. Kreng and Wu [11] investigated the problems of operational flexibility, optimal production rate and production runs in multi-item EPQ models with production rates as decision variables. Choi and Noble [12] considered a multi-item EPQ model with respect to material handling equipment selection and requirements, unit load size, and material flow path selection. These constitute issues that have been ignored in the traditional EPQ model. In their model, there is sufficient production capacity to meet the total demand. Das et al. [13] formulated a multi-item inventory problem with demand-dependent unit cost and infinite replenishment in both crisp and fuzzy environments under the constraints of the total storage area, total average shortage cost and total average inventory investment cost. In their paper, inventory costs are dependent on the respective quantities, where limited shortages are allowed and fully backlogged. Maity and Maiti [14] considered a multi-item production-inventory model with fuzzy constraints. In this study, the space required per item, the available storage space and investment capital are assumed to be imprecise. In this model, shortages are allowed and the rate of production is assumed to be a controllable function of time. Sharma [15] studied a multi-item EPQ model in a flexible production environment that it is feasible to interchange the production rate of two items. In this study, the production rates could be increased or decreased with no shortages allowed. Hou [16] considered an EPQ model with imperfect production process in which both setup cost and process quality are functions of capital expenditure. A mathematical model was derived to investigate the effects of an imperfect production process on the optimal production cycle time. In their model, the production capacity is assumed to be large enough such that shortages do not occur. Islam and Roy [17] formulated a multi-item EPQ model considering the flexibility and reliability of a production process for a demand-dependent unit production cost with fuzzy parameters, storage space constraints, and no shortages. Pasandideh et al. [18] considered a multi-item EPQ problem with limited warehouse-space where shortages are backlogged. In a later paper, Pasandideh et al. [19] developed a multi-item EPQ model where defective items are reworked with no shortages. Taleizadeh et al. [20–23] studied a multi-item EPQ model with random defective items and failure to repair. In their paper, a single machine with limited production capacity and shortages has been considered. Mandal et al. [24] formulated a multi-item production inventory control problem for defective items with fuzzy time horizon where the production rate is a time-dependent controllable variable. They assumed sufficient production capacity to avoid shortages. Taleizadeh et al. [25] considered two multi-item EPQ models with and without rework where shortages are allowed and backordered. For each model, the cycle length, the backordered and production quantities of each product are optimized. Maity [26] considered multi-item problem in a production-inventory system under imprecise space constraints. The demand is dependent on time and known. In this model, the production rate is limited, but it satisfies demand. Bjork [27] investigated a multi-item EPQ with fuzzy cycle time given enough production capacity to satisfy demand for all items. Lee and Yang [28] proposed an extended version of the EPQ model that includes a positive re-setup point and a fixed lot size to deal with random demand. Due to the random demand, shortages may occur during the production period and/or at the end of the replenishment cycle.

From the literature review of multi-item EPQ models, it may be concluded that some researchers have assumed that there is a sufficient production capacity to meet demand for all items. However, this may be an unrealistic assumption in real-life multi-item production system. Thus, in this paper, a limited production capacity is considered. That is, the production rate is less than the demand rate making shortages unavoidable. A policy of incurring shortages is neither operationally nor economically appropriate because of the eventual loss of potential customers in the long run [29]. On the other hand, it is very common for management to resort to external sources when demand exceeds the available production capacity [30]. Hence, a policy is proposed in this study whereby any demand that is not supplied by production is met by an outside supplier in order to prevent shortages.

The paper is organized as follows. Following the introduction in Section 1, a mathematical programming problem is formulated in Section 2. The procedure to solve this problem is introduced in Section 3. In Section 4, the Taguchi method is applied to tune the parameters of the algorithm. Lower and upper bounds for the optimal value of the objective function are derived in Section 5. The numerical results are presented in Section 6. Finally, conclusions are drawn and suggestions for future research are discussed in Section 7.

2. Modeling the problem

In this paper, multi-item EPQ model in a deterministic environment where a certain number of items are produced by a single machine is considered and dealt with. The classic EPQ model assumes that the production rate is more than the demand rate. There are some practical situations where the production rate may get less than that of the demand rate. For example, when an item reaches the growth stage in its production life, its demand increases and may exceed its production rate [31]. As another example consider the case when the production process is imperfect and some defective items needs to be reworked. In this case the production capacity may be limited because the production and the rework processes use the same resources. In this situation, the inefficiency of the previous research where the shortage is permitted to be dealt with limited production capacity is mitigated.

Thus in this paper, the manufacturer is allowed to procure the supplementary demands from the outside in order to deal with this situation and prevent shortages. The objective is to minimize the total inventory cost including holding cost, ordering cost, setup cost, purchasing cost, and production cost of the item where the order and production quantities are decision variables. The model is formulated on the basis of the following assumptions:

1. Demand for each item is continuous and it has a constant rate.
2. During a production run, production of each item is continuous and it has a constant rate.
3. Shortages are not allowed.
4. All items are produced by a single machine.
5. The machine can produce one item at a time.
6. All the production capacity must be used.
7. The machine setup time is zero.

For items $k = 1, \ldots, n$, the following notations are defined:

n: number of items.
Q_{1k}: order quantity of item k (decision variable).
Q_{2k}: production quantity of item k (decision variable).
R_{1k}: net inventory at the beginning of production in a period for item k for which the production rate is more than the demand rate (decision variable).
R_{2k}: net inventory at the beginning of production in a period for item k for which the production rate is less than the demand rate (decision variable).
P_k: production rate of item k (units/unit time).
D_k: demand rate of item k (units / unit time).
h_k: holding cost rate of item k ($/unit/ unit time).
A_{1k}: ordering cost of item k ($/order).
A_{2k}: setup cost for production of item k ($/setup).
C_{1k}: price of ordering per unit of item k ($/unit).
C_{2k}: price of production per unit of item k ($/unit).
T_k: duration of period of item k (unit time).
t_{pk}: duration of production in each period of item k (unit time).
$NI_k(t)$: net inventory at time t for item k (units).
TMC_k: total material cost of item k per unit time ($/unit time).
THC_k: total holding cost of item k per unit time ($/unit time).
TOC_k: total ordering cost of item k per unit time ($/unit time).
TIC_k: total inventory cost of item k per unit time ($/unit time.
$STIC$: total inventory cost of all items per unit time ($/unit time).

In order to fully explain the modeling methodology, the proposed model is first applied to the following two conditions for a single item, and then the model is extended to multi items.

Condition 1: the production rate is more than demand rate, but the existence of just a single machine results in some restrictions in production capacity.
Condition 2: the production rate is less than the demand rate.

2.1. Single item (with a production rate that is more than the demand rate)

Consider a single machine producing item k and the production rate is assumed to be more than the demand rate. Suppose that restrictions in production lead to the procurement of a certain portion of the demand from an outside supplier. The objective here is to determine the optimum quantities of production and the order such that the total inventory cost including holding cost, ordering cost, setup cost, purchasing cost, and production cost is minimized.

Fig. 1 shows the net inventory changes during a given period. At the beginning of the period the order quantity (Q_{1k}) is received. This batch is consumed for the duration of t_{1k} with the demand rate (D_k) until the net inventory decreases to R_{1k}. Then the machine starts to produce for the duration of t_{pk} during which production and consumption occur simultaneously. Since the production rate is more than the demand rate, the net inventory increases with the rate of $P_k - D_k$ in the length of t_{pk}. At the end of t_{pk} the production stops and the goods in the inventory are consumed with the demand rate (D_k) until the net inventory decreases to zero. The other periods will continue as well.

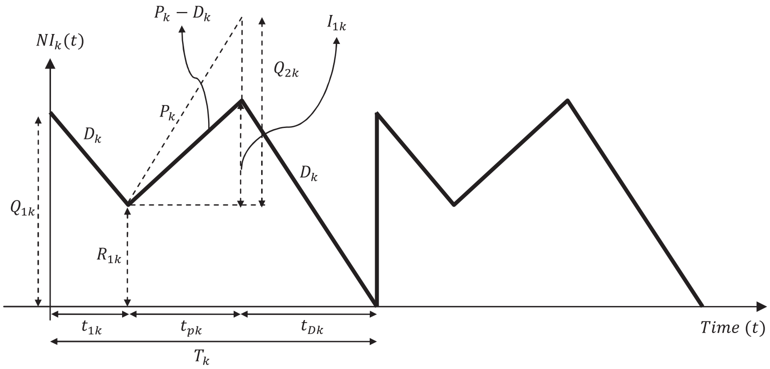

Fig. 1. Net inventory diagram when $P_k > D_k$.

From Fig. 1, we can conclude that:

$$t_{1k} = \frac{Q_{1k} - R_{1k}}{D_k}, \tag{1}$$

$$t_{pk} = \frac{Q_{2k}}{P_k}, \tag{2}$$

$$t_{Dk} = \frac{I_{1k} + R_{1k}}{D_k}, \tag{3}$$

$$I_{1k} = Q_{2k}\left(1 - \frac{D_k}{P_k}\right), \tag{4}$$

$$T_k = \frac{Q_{1k} + Q_{2k}}{D_k}. \tag{5}$$

The total inventory cost (TIC_k) is:

$$TIC_k = TMC_k + TOC_k + THC_k, \tag{6}$$

where

$$TMC_k = (C_{1k}Q_{1k} + C_{2k}Q_{2k}) \times \frac{1}{T_k}, \tag{7}$$

$$TOC_k = (A_{1k} + A_{2k}) \times \frac{1}{T_k}, \tag{8}$$

$$THC_k = h_k \times \frac{1}{T_k}\left[\frac{(Q_{1k} + R_{1k})(t_{1k})}{2} + \frac{(2R_{1k} + I_{1k})(t_{pk})}{2} + \frac{(I_{1k} + R_{1k})(t_{Dk})}{2}\right]. \tag{9}$$

After substituting t_{1k}, t_{Pk}, t_{Dk}, I_{1k} and T_k from Eqs. (1)-(5) in Eqs. (7)-(9), the total inventory cost is formulated as follows:

$$TIC_k = (C_{1k}Q_{1k} + C_{2k}Q_{2k}) \times \frac{D_k}{Q_{1k} + Q_{2k}} + (A_{1k} + A_{2k}) \times \frac{D_k}{Q_{1k} + Q_{2k}} + \frac{h_k D_k}{Q_{1k} + Q_{2k}}\left[\frac{(Q_{1k} + R_{1k})(Q_{1k} - R_{1k})}{2D_k} + \frac{\left(2R_{1k} + Q_{2k}\left(1 - \frac{D_k}{P_k}\right)\right) \times Q_{2k}}{2P_k} + \frac{\left(Q_{2k}\left(1 - \frac{D_k}{P_k}\right) + R_{1k}\right)^2}{2D_k}\right]. \tag{10}$$

TIC_k may be rewritten as follows:

$$TIC_k = \frac{D_k(C_{1k}Q_{1k} + C_{2k}Q_{2k}) + D_k(A_{1k} + A_{2k}) + \frac{h_k}{2}\left(Q_{1k}^2 + Q_{2k}^2\left(1 - \frac{D_k}{P_k}\right) + 2R_{1k}Q_{2k}\right)}{Q_{1k} + Q_{2k}}, \tag{11}$$

$S.t.:$

$$R_{1k} \leqslant Q_{1k}, \tag{12}$$

$$Q_{1k}, Q_{2k}, R_{1k} \geqslant 0, \tag{13}$$

where Q_{1k}, Q_{2k} and R_{1k} are the decision variables.

2.2. Single item (with a production rate that is less than the demand rate)

In this case, the production rate is assumed to be less than the demand rate. Thus replenishment from an outside source is necessary in order to prevent any shortages.

Fig. 2 shows the net inventory changes during a period for an item such as k. At the beginning of the period the order quantity (Q_{1k}) is received. This batch is consumed for the duration of t_{1k} with the demand rate (D_k) until the net inventory decreases to R_{2k}. Then the machine starts to produce for the duration of t_{2k} until the net inventory decreases to zero. Production and consumption of goods occur simultaneously during t_{2k}. Since the production rate is less than the demand rate, the net inventory in the length of t_{2k} decreases at the rate of $D_k - P_k$. Thus, the net inventory decreases to zero and production stops at the end of the period. The other periods will proceed in the same manner.

From Fig. 2, it can be seen that:

$$t_{1k} = \frac{Q_{1k} - R_{2k}}{D_k}, \tag{14}$$

$$t_{2k} = \frac{R_{2k}}{D_k - P_k}, \tag{15}$$

$$T_k = \frac{Q_{1k} + Q_{2k}}{D_k}, \tag{16}$$

$$Q_{2k} = \frac{P_k}{D_k - P_k} \times R_{2k}, \tag{17}$$

And, the total inventory cost (TIC_k) will be described as shown in Eq. (6), in which

$$TMC_k = (C_{1k}Q_{1k} + C_{2k}Q_{2k}) \times \frac{1}{T_k}, \tag{18}$$

$$TOC_k = (A_{1k} + A_{2k}) \times \frac{1}{T_k}, \tag{19}$$

$$THC_k = h_k \times \frac{1}{T_k}\left[\frac{(Q_{1k} + R_{2k})t_{1k} + R_{2k}t_{2k}}{2}\right]. \tag{20}$$

By substituting t_{1k}, t_{2k} and T_k from Eqs. (14)-(16) in Eqs. (18)-(20), the total inventory cost can be formulated as follows:

$$TIC_k = \frac{1}{T_k}\left[(C_{1k}Q_{1k} + C_{2k}Q_{2k}) + (A_{1k} + A_{2k}) + \frac{h_k}{2}\left(\frac{\frac{(Q_{1k}^2 - R_{2k}^2)}{D_k} + \frac{R_{2k}^2}{D_k - P_k}}{2}\right)\right]. \tag{21}$$

Then, TIC_k may be summarized as follows:

$$TIC_k = \frac{D_k}{Q_{1k} + Q_{2k}}\left[(C_{1k}Q_{1k} + C_{2k}Q_{2k}) + (A_{1k} + A_{2k}) + \frac{h_k}{2}\left(\frac{Q_{1k}^2(D_k - P_k) + R_{2k}^2 P_k}{D_k(D_k - P_k)}\right)\right]. \tag{22}$$

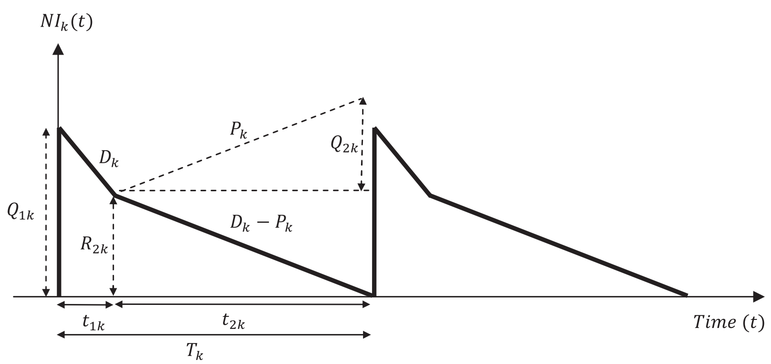

Fig. 2. Net inventory diagram when $P_k < D_k$.

$S.t.:$

$$Q_{2k} = \frac{P_k}{D_k - P_k} \times R_{2k}, \tag{23}$$

$$R_{2k} \leqslant Q_{1k}, \tag{24}$$

$$Q_{1k}, Q_{2k}, R_{2k} \geqslant 0, \tag{25}$$

where Q_{1k}, Q_{2k} and R_{2k} are the decision variables.

2.3. Multi items

In this section, the models of Sections 2.1 and 2.2 are extended to the case of multi items. We have n items whose demands must be met without any shortages. Some items have demand rates less than the production rates (and are shown by index i) and others have demand rates that exceed their production rate (as shown with index j). Fig. 3 shows the net inventory changes for 4 items where items 1 and 3 have a demand rate that is less than the production rate ($D_i < P_i, i = 1, 3$) and items 2 and 4 have a demand rate that is more than the production rate ($D_j > P_j, j = 2, 4$).

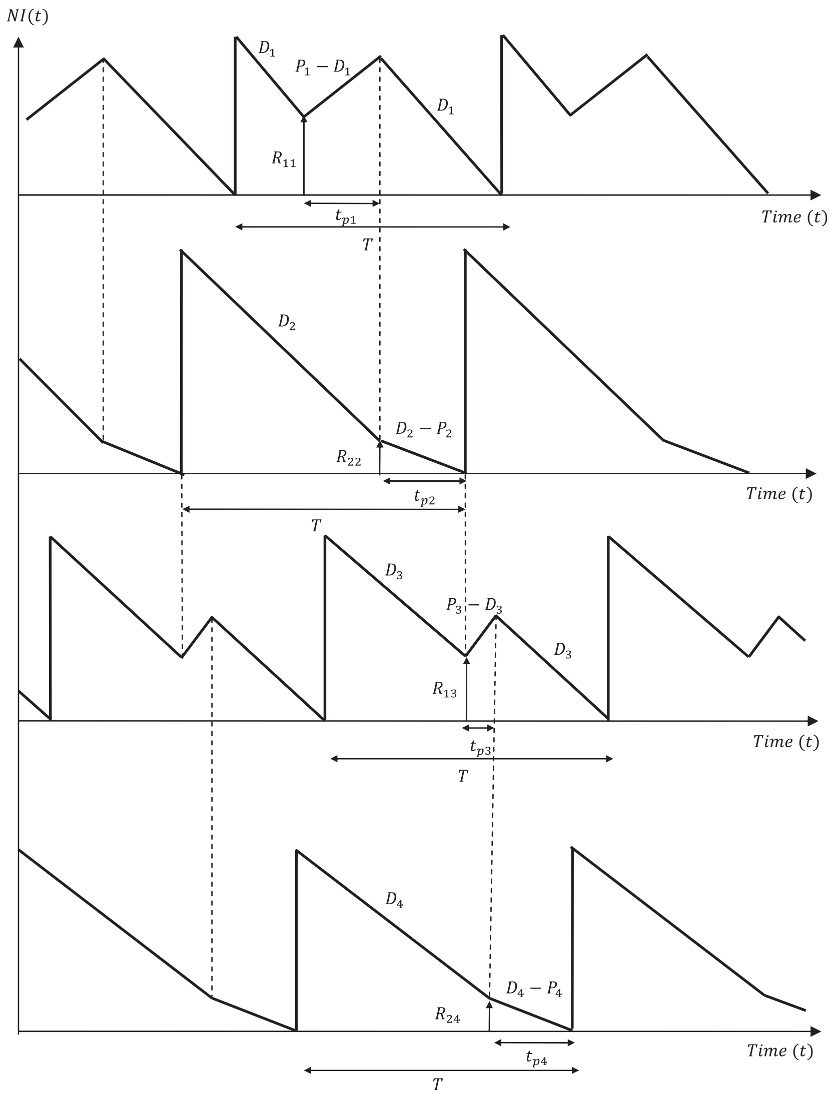

Fig. 3. Net inventory diagram for four items.

The model proposed in this paper must be used in order to prevent shortages. The objective function here is the sum of total inventory costs of all items (*STIC*) and it can be formulated as follows:

$$STIC = \sum_{\forall i} \frac{D_i(C_{1i}Q_{1i} + C_{2i}Q_{2i}) + D_i(A_{1i} + A_{2i}) + \frac{h_i}{2}\left(Q_{1i}^2 + Q_{2i}^2\left(1 - \frac{D_i}{P_i}\right) + 2R_{1i}Q_{2i}\right)}{Q_{1i} + Q_{2i}} + \sum_{\forall j} \frac{D_j}{Q_{1j} + Q_{2j}}\left[(C_{1j}Q_{1j} + C_{2j}Q_{2j}) + (A_{1j} + A_{2j}) + \frac{h_j}{2}\left(\frac{Q_{1j}^2(D_j - P_j) + R_{2j}^2 P_j}{D_j(D_j - P_j)}\right)\right]. \tag{26}$$

The length of period must be assumed to be the same for all items, i.e. $T_1 = T_2 = \ldots = T_n = T$ in order to ensure the validity of assumption 5 (the machine can produce one item at a time). This is a well-known assumption in the literature on the EPQ model [22,23,32].

Equation (27) satisfies assumptions 5 and 6 (all the production capacity must be used).

$$\sum_{k=1}^{n} t_{pk} = T, \tag{27}$$

where t_{pk} is the duration of production in a period of item k which is equal to $\frac{Q_{2k}}{P_k}$.

If the order (production) quantity of an item is zero, then its ordering (setup) cost must be omitted from the objective function. This can be applied by using binary variables.

Finally, the mathematical programming of the problem may be written as follows:

$$Min\,STIC = \sum_{\forall i} \frac{D_i(C_{1i}Q_{1i} + C_{2i}Q_{2i}) + D_i(A_{1i}y_{1i} + A_{2i}y_{2i}) + \frac{h_i}{2}\left(Q_{1i}^2 + Q_{2i}^2\left(1 - \frac{D_i}{P_i}\right) + 2R_{1i}Q_{2i}\right)}{Q_{1i} + Q_{2i}} + \sum_{\forall j} \frac{D_j}{Q_{1j} + Q_{2j}}\left[(C_{1j}Q_{1j} + C_{2j}Q_{2j}) + (A_{1j}y_{1j} + A_{2j}y_{2j}) + \frac{h_j}{2}\left(\frac{Q_{1j}^2(D_j - P_j) + R_{2j}^2 P_j}{D_j(D_j - P_j)}\right)\right]. \tag{28}$$

$S.t.:$

$$\sum_{k=1}^{n} \frac{Q_{2k}}{P_k} = T, \tag{29}$$

$$T_k = T; \quad k = 1, \ldots, n, \tag{30}$$

$$T_k = \frac{Q_{1k} + Q_{2k}}{D_k}; \quad k = 1, \ldots, n, \tag{31}$$

$$R_{1i} \leqslant Q_{1i}; \quad \forall i, \tag{32}$$

$$Q_{2j} = \frac{P_j}{D_j - P_j} \times R_{2j}; \quad \forall j, \tag{33}$$

$$R_{2i} \leqslant Q_{1j}; \quad \forall j, \tag{34}$$

$$Q_{1k} \leqslant My_{1k}; \quad k = 1, \ldots, n, \tag{35}$$

$$Q_{2k} \leqslant My_{2k}; \quad k = 1, \ldots, n, \tag{36}$$

$$Q_{1k}, Q_{2k}, R_{1k}, R_{2k} \geqslant 0; \quad k = 1, \ldots, n, \tag{37}$$

$$T \geq 0, \tag{38}$$

$$y_{1k}, y_{2k}\{0, 1\}; \quad k = 1, \ldots, n. \tag{39}$$

M is a big positive number.

3. The approach to solution

The model provided above is a mixed nonlinear programming problem. To find a closed form of the expression for the optimum solution, one must focus on the Karush–Kuhn–Tucker (KKT) conditions. However, it is impossible to analytically solve the equations derived from KKT conditions. As a part of this study, an attempt was made to generate a heuristic algorithm to find the exact solution for this problem. However, these efforts only resulted in finding a lower and an upper bound for the optimal value of the objective function. Consequently, a hybrid meta-heuristic algorithm was used to solve the proposed model.

In this paper, particle swarm optimization (PSO) is employed along with genetic algorithm (GA) to explore the search-space, and it is combined with local search to update the particles. This is called GPSO-Based local search (GPSO-LS) and it is developed in order to find a near-optimum solution.

Moreover, GA and PSO models are proposed to justify the efficiency of the GPSO-LS. The details of the algorithms are given in the following subsections.

3.1. Particle swarm optimization

The particle swarm optimization algorithm is an evolutionary computation technique developed by Kennedy and Eberhart [33]. The underlying motivation for the development of PSO algorithm was the social behavior of animals such as flocking of birds, schooling of fish that lead to swarm theory. In PSO, a set of randomly generated solutions are reproduced in the search-space towards the optimal solution over a number of iterations based on a large amount of information about the search-space that is assimilated and shared by all members of the swarm. In fact, solutions are particles having position and velocity. Each particle's movement is influenced by its local best-known position. It is also guided towards the best-known positions in the search-space that are updated as better positions are found by other particles. This is expected to move the swarm toward the best solution. The basic stages of a PSO algorithm consist of initialization and updating position and velocity of the particles. The structure of the developed PSO is presented next.

3.1.1. Initializing particles' positions and exploration velocities

PSO is initialized with a group of random particles and then searches for optima by updating generations. Each element of particle i is assigned a random number between $\vec{b}_{lo}$ and $\vec{b}_{up}$ which are the upper and lower bounds of the search-space. In this model, the lower bound of each variable is zero and a large positive number obtained with respect to the dimensions of the problem is assigned as the upper bound of each variable. The position of particle i is represented as shown in Fig. 4. Column k includes the decision variables of item k; $k = 1, \ldots, n$. The first and the second rows are the order and the production quantities of the items, respectively. The third row is the net inventory at the beginning of production in the period (R_{1i} or R_{2i} which depends on demand rate and production rate).

At first, the positions, $\vec{x}_i$, and the exploration velocities, $\vec{v}_i$, of the initial swarm of particle i can be randomly generated as follows:

$$\vec{x}_i = \vec{x}_{min} + Rand(\vec{x}_{max} - \vec{x}_{min}),$$

$$\vec{v}_i = \vec{v}_{min} + Rand(\vec{v}_{max} - \vec{v}_{min}).$$

Q_{2i} is randomly generated and then $Q_{1i} = T \times D_i - Q_{2i}$ is computed based on Eq. (31). Also R_{1i} is randomly generated such that $R_{1i} < Q_{1i}$. Finally, $R_{2j} = \frac{D_j - P_j}{P_j} \times Q_{2j}$ is computed based on Eq. (33). Therefore, particle positions and velocity updates in PSO are only applied to Q_{2i}. Since the decision variables Q_{1i} or Q_{2i} may be zero for the optimal solution, it was decided to generate ten percent of the initial population with $Q_{1i} = 0$ or $Q_{2i} = 0$.

3.1.2. Updating exploration velocity and position

Let $\vec{g}$ be the best particle among all particles and $\vec{p}_i$ be the best fitness value for particle i. Then, the exploration velocity is updated as follows:

$$\vec{v}_i \leftarrow w\vec{v}_i + \varphi_p r_p(\vec{p}_i - \vec{x}_i) + \varphi_g r_g(\vec{g} - \vec{x}_i).$$

Moreover, the position of particle i is updated as:

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i,$$

where r_p and r_g are uniformly distributed random numbers between 0 and 1. The coefficients φ_p and φ_g are the given acceleration constants towards $\vec{p}_i$ and $\vec{g}$, respectively; and w is the inertia weight. Note that the inertia weight controls how much of the previous exploration velocity should be retained from the previous step. A larger inertia weight facilitates a global search, while a smaller inertia weight facilitates a local search. A balance can be obtained between global and local

Q_{1i}	$\cdots$	Q_{1k}	$\cdots$	Q_{1n}
Q_{2i}	$\cdots$	Q_{2k}	$\cdots$	Q_{2n}
R_{1i} or R_{2i}	$\cdots$	R_{1k} or R_{2k}	$\cdots$	R_{1n} or R_{2n}

Fig. 4. A particle and chromosome representation.

exploration to speed up search results using a dynamically adjustable inertia weight formulation. Introducing a linearly decreasing inertia weight into the original PSO significantly improves its performance through the parameter study of inertia weight. The linear distribution of the inertia weight is expressed as follows:

$$w = w_{max} - \frac{w_{max} - w_{min}}{N_1} \times k,$$

where, w_{max} and w_{min} are the initial and final values of the weighting coefficient. N_1 and k are the maximum iteration number and iteration counter, respectively. Usually, w_{max} and w_{min} are set to be 0.9 and 0.4, respectively [34].

3.1.3. Termination criterion

Termination criterion checks whether or not the method has found a solution that is good enough to meet the user's expectations. A stopping criterion is defined based on different conditions such as achieving a predetermined solution, stopping at a certain or a predetermined (CPU) time, or the algorithm is repeated until a maximum number of iterations is reached. This last criterion is used in this research.

The pseudo code of the developed PSO algorithm for this problem is shown in Fig. 5.

3.2. Genetic algorithm

Genetic algorithm (GA) is a stochastic global search which operates on a population of potential solutions applying the principle of survival of the fittest to converge to a near optimal solution. GA is a procedure to search the population in parallel and its search direction is influenced by the objective function. The algorithm is independent of the complexity of the considered performance index and all that is needed is to specify the objective function and finite bounds of variables. For more details regarding genetic algorithm, the reader is referred to [35].

3.2.1. The chromosomes

In the GA method, a chromosome is represented by a matrix as shown in Fig. 4. It should be noted that in chromosome representation, the products with $D < P$ are placed first and then the products with $D > P$ are placed.

3.2.2. Evaluation

To compare the chromosomes, a fitness function is needed to be assigned to each of them. In this study, the objective function is applied as the fitness function.

3.2.3. Initial population

The initial population for GA is similar to the one used for PSO. Also it is noteworthy that GA operations including crossover and mutation are only applied to Q_{2i}.

3.2.4. Crossover method

A uniform crossover is employed in this algorithm. In this type of crossover, a random mask string whose length is equal to the length of the chromosome is generated first. The bits of this string determine the parent whose corresponding bit will supply the offspring. This process is illustrated in Fig. 6 (that only shows a part of the strings). The offspring 1 is generated by taking the genes from parent 1 if the corresponding mask bit is 1 and the genes from parent 2 if the corresponding mask bit is 0. The offspring 2 is created using the inverse of the mask string.

3.2.5. Mutation method

In this study, the reverse mutation method is used. In this method, two genes are randomly selected and then the genes among them are reversed. This process is illustrated in Fig. 7 (that only shows a part of the strings).

3.2.6. Chromosomes selection

Several selection methods, such as the roulette wheel, tournament, ranking, and elitist are discussed by Michalewicz [36]. In this research, the ranking method is employed to guide the search process towards more promising regions in the search-space. Furthermore, a detailed explanation of the operation of ranking selection can be found in [37].

3.2.7. Termination criterion

The stopping criterion of GA is similar to the one used for PSO (to reaching a specific number of generations).

Fig. 8 depicts the pseudo code of the developed GA for this study.

3.3. Local search algorithm (LS)

Local search algorithm moves from one solution to the next solution in the search space by applying local changes to improve the fitness function. In this study, local change for a given solution is constructed by selecting one bit (gene) of a given solution randomly and changing the selected bit at random. The change in a given solution is done until the improvement is made. Fig. 9 depicts the pseudo code of the local search algorithm developed in this study.

- **Initialization**
  Pop_size_1: Population size,
  φ_p: Behavioral parameter,
  φ_g: Behavioral parameter,
  Max_G_1: Maximum number of iterations.
- Initialize each particle $\vec{x}_i \in R^n$ with a random position in the search-space:

$$\vec{x}_i \sim U(\overrightarrow{b_{lo}}, \overrightarrow{b_{up}})$$

  Where $\overrightarrow{b_{lo}}$ and $\overrightarrow{b_{up}}$ are the lower and upper boundaries of the search-space.

- Set each particle's best known position to its initial position:

$$\vec{p}_i \leftarrow \vec{x}_i$$

- Initialize each particle's velocity $\vec{v}_i \in R^n$ to random values:

$$\vec{v}_i \sim U(-\vec{d}, \vec{d})$$

  Where $\vec{d} = |\overrightarrow{b_{lo}} - \overrightarrow{b_{up}}|$.

- Initialize the swarm's best known position $\vec{g}$ to the $\vec{x}_i$ for which $f(\vec{x}_i)$ is the lowest.
- Until a termination criterion is met, repeat the following:
  - For each particle $\vec{x}_i$ in the swarm do the following:
    Pick two random numbers: $r_p, r_g \sim U(0,1)$
    Update the particle's velocity $\vec{v}_i$ as follows:

$$\vec{v}_i \leftarrow w\vec{v}_i + \varphi_p r_p(\vec{p}_i - \vec{x}_i) + \varphi_g r_g(\vec{g} - \vec{x}_i)$$

    Bound the velocity, that is, for all dimensions i, update v_i:

$$v_i \leftarrow Bound(v_i, -d_i, d_i)$$

    Where $Bound(x, l, u) = \begin{cases} l, & x < l \\ u, & x > u \\ x, & else \end{cases}$

    Move the particle to its new position by adding its velocity:

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i$$

    Bound the position, that is, for all dimensions i update x_i:

$$\vec{x}_i \leftarrow Bound(x_i, b_{lo_i}, b_{up_i})$$

    If $(f(\vec{x}_i) < f(\vec{p}_i))$ then update the best known position for particle i:

$$\vec{p}_i \leftarrow \vec{x}_i$$

    If $(f(\vec{x}_i) < f(\vec{g}))$ then update the swarm's best known position among all:

$$\vec{g} \leftarrow \vec{x}_i$$

- Now $\vec{g}$ holds the best found position in the search-space.

Fig. 5. Pseudo code of PSO.

Parent 1	1760	0	2356	1280	0
Parent 2	0	1250	0	1950	2340

Mask	0	0	1	0	1

Offspring 1	1760	0	0	1280	2340
Offspring 2	0	1250	2356	1950	0

Fig. 6. An example of the crossover operation.

Parent 1	1670	345	2356	1790	0

After mutation:

Parent 1	1670	0	1790	2356	345

Fig. 7. An example of the mutation operation.

- **Initialization**
  Input
  Pop_size_2: Population size,
  P_e: Rate of elitism,
  P_m: Rate of mutation,
  Max_G_2: Maximum number of iterations.
  Generate Pop_size_2 feasible solutions randomly and save them in the population Pop.

- **Loop until the terminal condition**

  for $i = 1$ to Max_G_2 do

  - **Elitism based selection**
    number of elitism $ne = P_e.Pop\_size_2$;
    select the best ne solutions in Pop and save them in Pop_1;

  - **Crossover**
    number of crossover $nc = (Pop\_size_2 - ne)/2$;
    for $j = 1$ to nc do
    randomly select two solutions $\vec{x}_A$and $\vec{x}_B$ from Pop;
    generate $\vec{x}_C$ and $\vec{x}_D$ by one-point crossover to $\vec{x}_A$ and $\vec{x}_B$;
    save $\vec{x}_C$ and $\vec{x}_D$ to Pop_2;
    end for

  - **Mutation**
    for $j = 1$ to nc do
    select a solution $\vec{x}_j$ from Pop_2;
    mutate each bit of $\vec{x}_j$ under rate P_m and generate a new solution $\vec{x}_j'$;
    if $\vec{x}_j'$ is unfeasible
    update $\vec{x}_j'$ with a feasible solution by repairing $\vec{x}_j'$;
    end if
    update $\vec{x}_j$ with $\vec{x}_j'$ in Pop_2;

    end for

  - **Updating**
    update $Pop = Pop_1 + Pop_2$;

  end for

- **Returning the best solution**
  return the best solution $\vec{x}$ in Pop;

Fig. 8. Pseudo code of GA.

3.4. Hybrid of GA and PSO based on local search (GPSO-LS)

In this subsection, a hybrid intelligent algorithm combined GA with PSO based local search will be used. The schematic representation of the proposed algorithm is depicted in Fig. 10. Both algorithms work with the same initial population. This approach uses $2N$ individuals, randomly generated as explained in Section 3.1.1. These individuals can be concerned as chromosomes in the case of GA or as particles in the case of PSO. The $2N$ individuals are sorted by fitness values, and then GA is applied to the top N individuals to create N individuals. Make a copy of the new N individuals for the next generation. Next, the N individuals created from GA and the remaining N individuals are considered as particles for the PSO algorithm. Having applied the PSO algorithm, the top N individuals created by PSO, are sent to the next generation with the N individuals created by GA. After that, the local search algorithm (LS) is applied to improve the fitness function. The number of function evaluations is the termination criterion for this algorithm.

for $i = 1$ to Pop_size_3
do
Generate a neighborhood solution of $\vec{x}_j$ and make new solution $\vec{x}_j'$:
Select bit i in solution j randomly and do one of the following perturbation randomly:

$$Q_{2i} \leftarrow Q_{2i} + rand(x_{max} - Q_{2i})$$
$$Q_{2i} \leftarrow Q_{2i} - rand(Q_{2i} - x_{min})$$

Where x_{min} and x_{max} are lower and upper bound of the search-space.
while $(f(\vec{x}_j') < f(\vec{x}_j))$
end for

Fig. 9. Pseudo code of LS.

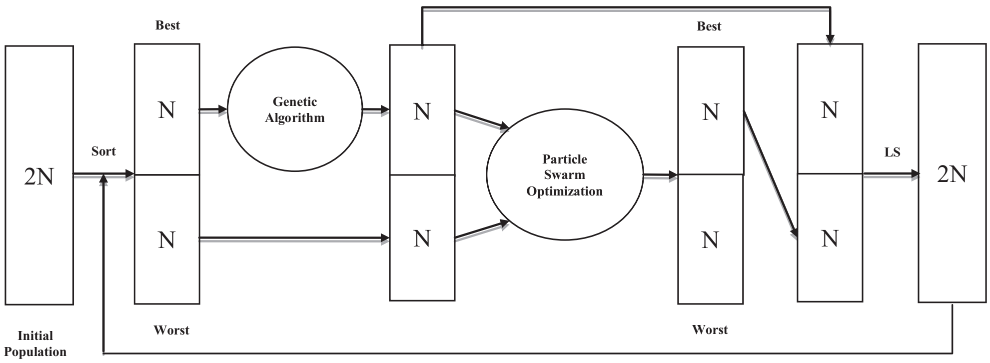

Fig. 10. Schematic representation of GPSO-LS.

4. Taguchi parameter design

The effectiveness of evolutionary algorithms greatly depends on the correct choice of parameters. In this section, we investigate the behavior of parameters φ_p, φ_g, and Pop_size_1 on the effectiveness of the proposed PSO and parameters P_e, P_m, and Pop_size_2 on GA. A full factorial design that tests all possible combinations of factors, is the method widely used in the most reported research. However, when the number of the factors increases, it does not seem to be effective. There exist several experimental design techniques such as trial and-error procedure, Response Surface Methodology (RSM), and regression analysis among which Taguchi method [38] has been successfully applied to tune parameters. Taguchi method uses an orthogonal array to organize the experimental results. In the mid-1980s, the Japanese quality consultant, Genichi Taguchi, popularized a cost-efficient approach known as robust parameter design. He assumed that there are two types of factors operating on a process: control factors and noise factors. Taguchi divided the factors into two basic clusters: controllable and noise factors (uncontrollable). Due to unpractical and often impossible omission of noise factors, Taguchi attempted to minimize the impact of noise and found the best level of influential controllable factors on the basis of robustness. Moreover, Taguchi specified the relative importance of each factor with respect to its main impact on the performance. A transformation of petition data to another value which is the measure of variation was developed by Taguchi. The transformation is the signal-to-noise (S/N) ratio. Here, the term 'signal' denotes the desirable value (response variable) and 'noise' denotes the undesirable value (standard deviation). Therefore, the S/N ratio indicates the amount of variation present in the response variable. The aim is to maximize the signal-to-noise ratio. In the Taguchi method, the S/N ratio of a minimization objective is as follows [39]:

$$(S/N)_i = -10log_{10}[y_i^2], \tag{40}$$

where $(S/N)_i$ is S/N ratio and y_i is objective value of trial i.

In the following, the PSO parameter tuning is described in detail. Three levels as shown in Table 1 are considered for each parameter. The selected orthogonal array should be able to accommodate for the factor level combinations in the experiment. Hence, L9 is an appropriate array.

4.1. Parameter tuning

To tune the parameters (φ_p, φ_g, and N_1) an experiment based on the L9 orthogonal array is designed. Nine different combinations of factors shown in Table 2 are considered. We refer to a combination of factors as a trial. In each trial, eight instances which are instances No. 7–14 of Table 6, in Section 6, are solved.

Table 1
Factors and their levels.

Factors	φ_p	φ_g	Pop_size_1
Level 1	1.5	1.5	300
Level 2	1.75	1.75	500
Level 3	2.05	2.05	700

Table 2
Combination of factors in each trial.

Trial	Level of factors		
	φ_p	φ_g	Pop_size_1
1	1	1	1
2	1	2	2
3	1	3	3
4	2	1	2
5	2	2	3
6	2	3	1
7	3	1	3
8	3	2	1
9	3	3	2

Due to the stochastic nature of PSO, five replications are performed for each instance to achieve more reliable results. Totally 9 × 8 × 5 = 360 experiences are performed by the PSO algorithm. The objective values of instances could not be used directly because the scales of instances are different. Thus, the relative percentage deviation (RPD) defined in Eq. (43) is used for each instance.

$$RPD_{ijk} = \frac{Ob_{ijk} - MOb_{ij}}{MOb_{ij}} \times 100, \quad i = 1, \ldots, 9;\ j = 1, \ldots, 8;\ k = 1, \ldots, 5, \tag{41}$$

where, RPD_{ijk} and Ob_{ijk} are RPD and objective value of trial i, instance j, replication k, respectively, and $MOb_{ij} = min_{\forall k} Ob_{ijk}$.

After converting the objective values to RPDs, the mean RPD is calculated for each trial. These mean RPDs are then transformed to S/N ratios using Eq. (40) in which y_i is:

$$y_i = \frac{\sum_{j=1}^{8} \left(\frac{\sum_{k=1}^{5} RPD_{ijk}}{5} \right)}{8}. \tag{42}$$

Table 3 contains the numerical results of S/N ratios.

Table 4 includes mean S/N ratio for each level of factors. According to the results shown in Table 4, the maximum S/N ratio is on level 3 for each factor. Therefore, the best values of parameters of PSO are $\varphi_p = 2.05$, $\varphi_g = 2.05$ and $Pop\_size_1 = 700$.

To validate the results of the Taguchi method, some methodologies such as regression analysis by which we can confirm that the parameters setting have been modified correctly were used.

Using Taguchi procedure similar to that of PSO, it may be concluded that the best values of parameters for GA are $Pop\_size_2 = 500$, $P_e = 0.03$, $P_m = 0.2$ and for GPSO-LS are $Pop\_size_3 = 1000$, $\varphi_p = 2.05$, $\varphi_g = 2.05$, $P_e = 0.02$, $P_m = 0.2$.

The number of function evaluations is set to 300, and it is considered to be the stopping criterion for all the three algorithms in order to judge their performance on similar grounds.

Table 3
S/N ratio of each trail.

Trail	S/N ratio
1	−19.20
2	−18.73
3	−17.61
4	−18.41
5	−16.33
6	−17.25
7	−15.31
8	−15.45
9	−14.28

Table 4
Mean S/N ratio for each level of factors.

Factors	φ_p	φ_g	Pop_size_1
Level 1	−18.51	−17.64	−17.30
Level 2	−17.33	−16.83	−17.14
Level 3	−15.01	−16.38	−16.41

5. Lower and upper bounds

To gauge the quality of solutions provided by the GPSO-LS, lower and upper bounds for the optimal value of the objective function are developed.

5.1. Lower bound

To obtain a lower bound a valid relaxation must be solved. Our approach is to relax the production capacity restriction by making a modification on the production rate of each item. We refer to this model as the *modified model*. Consider the following notations:

MP_k: modified production rate of item k.
I: set of items which have demand rate less than the production rate.
J: set of items which have demand rate more than the production rate.

It is clear that the demand of at least $D_j - P_j$ must be procured from the outside for items that belong to J. Thus, we consider that a demand of P_j is satisfied through production for such items in the *modified model*.

To relax the production capacity restriction, we must have $\sum_{\forall i \in I} \frac{D_i}{MP_i} + \sum_{\forall j \in J} \frac{D_j}{MP_j} \leqslant 1$.

The following procedure provides the modified production rates:

Step 1: Set $H = I$ and $B = \Phi$ (null set).
Step 2: If $\sum_{\forall i \in H} \frac{D_i}{P_i} < 1$, then set $\varepsilon = 1 - \sum_{\forall i \in H} \frac{D_i}{P_i}$ and go step 4. Otherwise go step 3.
Step 3: $\frac{D_b}{P_b} = min_{\forall i \in H} \left\{ \frac{D_i}{P_i} \right\}$, insert item b to B and set $H = H - \{b\}$ and go step 2.
Step 4: Set $MP_i = P_i$ for item i; and $MP_g = \alpha.P_g$ for item $g; gG$.

$G = BUJ$ (union of sets B and J)
α is:

$$\alpha = \frac{\sum_{\forall b \in B} \frac{D_b}{P_b} + |J|}{\varepsilon}, \tag{43}$$

where $|J|$ is cardinal number of J.

Proposition1. The optimum solution of the *modified model* provides a valid lower bound for the optimal value of the objective function.

Proof. Any solution that is feasible for the original model is a feasible solution for the *modified model* since the two models are the same with the only difference being that the overall production capacity is increased in the *modified model*. Thus, the feasible solution set of the original model is a subset of the feasible solution set of the *modified model*.

The production capacity restriction is relaxed in the *modified model* and all the demands for the items can be met by production (except items J for which the demand equal to $D_j - P_j$ must be procured from the outside). For all items (such as k) we have $C_{1k} > C_{2k}$ and $A_{1k} > A_{2k}$. These cost parameters result in total inventory cost for production to be less than the cost of ordering from the outside. Consequently, the optimum solution of the *modified model* provides a lower bound for the optimal value of the objective function.

The optimum solution of the *modified model* is:

$$\forall i \in I; \quad R_{1i} = 0, \; Q_{1i} = 0, \; Q_{2i} = D_i T, \tag{44}$$

$$\forall j \in J; \quad R_{2j} = 0, \; Q_{1j} = (D_j - P_j)T, \; Q_{2j} = P_j T, \tag{45}$$

in which:

$$T = \sqrt{\frac{2\left(\sum_{\forall h\in H}A_{2h} + \sum_{\forall b\in B}A_{2b} + \sum_{\forall j\in J}A_{2j}\right)}{\sum_{\forall h\in H}h_hD_h\left(1-\frac{D_h}{P_h}\right) + \sum_{\forall b\in B}h_bD_b\left(1-\frac{D_b}{MP_b}\right) + \sum_{\forall j\in J}h_jP_j\left(1-\frac{P_j}{MP_j}\right)}}. \tag{46}$$

And, the minimum of total inventory cost for the *modified model* (TC_{mod}) is

$$TC_{mod} = \sqrt{2\left(\sum_{\forall h\in H}A_{2h} + \sum_{\forall b\in B}A_{2b} + \sum_{\forall j\in J}A_{2j}\right)\left(\sum_{\forall h\in H}h_hD_h\left(1-\frac{D_h}{P_h}\right) + \sum_{\forall b\in B}h_bD_b\left(1-\frac{D_b}{MP_b}\right) + \sum_{\forall j\in J}h_jP_j\left(1-\frac{P_j}{MP_j}\right)\right)} + \sum_{\forall j\in J}\sqrt{2(D_j-P_j)A_{1j}h_j} + \sum_{\forall i\in I}C_{2i}D_i + \sum_{\forall j\in J}C_{1j}P_j + \sum_{\forall j\in J}C_{2j}(D_j-P_j). \tag{47}$$

□

5.2. Upper bound

An upper bound is derived through finding a feasible solution. The approach here is to select some items such that all their demands can be totally met via production. The other items are totally satisfied from an outside supplier.

The following procedure provides a feasible solution.

Step 1: For each item, calculate TPC_i (TPC_j) and TOC_i (TOC_j) from Eqs. (49)-(52).

$$\forall i \in I;\quad TPC_i = \sqrt{2D_iA_{2i}h_i\left(1-\frac{D_i}{P_i}\right)} + C_{2i}D_i, \tag{48}$$

$$TOC_i = \sqrt{2D_iA_{1i}h_i} + C_{1i}D_i, \tag{49}$$

$$\forall j \in J;\quad TPC_j = \sqrt{2(D_j-P_j)A_{1j}h_j} + C_{2j}P_j + C_{1j}(D_j-P_j), \tag{50}$$

$$TOC_j = \sqrt{2D_jA_{1j}h_j} + C_{1j}D_j. \tag{51}$$

$TPC_i(TPC_j)$ is the minimum of total production cost when all the demand for item i (j) is totally satisfied via production. TOC_i (TOC_j) is the minimum of total ordering cost when item i (j) is totally procured from the outside.

Step 2: Set $F = \Phi$ and $K = I \cup J$ (*all items*).
Step 3: $\frac{TPC_f}{TOC_f} = max_{\forall k\in K}\left\{\frac{TPC_k}{TOC_k}\right\}$.
Step 4: If $\sum_{\forall f\in F}\frac{D_f}{P_f} \leqslant 1$ then insert item f to F, set $K = K - \{f\}$ and go to step 3.
Step 5: The feasible solution is:

$$\forall f \in F;\quad R_{1f} = 0,\quad Q_{1f} = 0,\quad Q_{2f} = D_fT,\quad T = \sqrt{\frac{2\sum_{\forall f\in F}A_{2f}}{\sum_{\forall f\in F}h_fD_f\left(1-\frac{D_f}{P_f}\right)}}, \tag{52}$$

$$\forall i \in I-F;\quad R_{1i} = 0,\quad Q_{1i} = \sqrt{\frac{2D_iA_{1i}}{h_i}},\quad Q_{2i} = 0, \tag{53}$$

$$\forall j \in J-F;\quad R_{2j} = 0,\quad Q_{1j} = \sqrt{\frac{2D_jA_{1j}}{h_j}},\quad Q_{2i} = 0. \tag{54}$$

For item f_L (f_L is the last item identified in *step3*):

$$R_{1f_L} = 0 \quad \text{or} \quad R_{2f_L} = 0,\quad Q_{1f_L} = D_{f_L}T - Q_{2f_L},\quad Q_{2f_L} = P_{f_L}\left(T - \sum_{\forall f\in F}\frac{Q_{2f}}{P_f}\right). \tag{55}$$

6. Numerical results

At first, some items as shown in Table 5 are constructed. In this Table, the parameters are randomly generated from: D_k = U [2000,6000], P_k = U [3000,7000], C_{1k} = U [30,40], C_{2k} = U [20,30], A_{1k} = U [20000, 40000], A_{2k} = U [6000, 12000], h_k = U [200,500]; $k = 1, \ldots, 15$. 14 numerical instances included 2 to 14 items are randomly constructed. Table 6 shows these instances.

The proposed algorithms are coded in Visual C++ and run on an IBM-compatible PC with a Pentium 2.33 GHz processor under Windows 7. Table 7 contains the results obtained by PSO, GA, and GPSO-LS in terms of the fitness function and CPU time. In order to make a comparison between the algorithms, all of the three algorithms are run five independent times, and the best solution among the five runs is recorded.

According to Table 7, one can see that GPSO-LS has a better performance than GA and PSO in both fitness and CPU time.

To provide a more accurate comparison of the performance of the algorithm, one-sided, paired-samples t-tests are carried out between all the three algorithms. To compare the proposed algorithms statistically, two hypotheses are organized as follows. Hypothesis 1 is defined to compare GA with PSO and hypothesis 2 is defined to compare PSO with GPSO-LS. μ_{GA}, μ_{PSO} and $\mu_{GPSO\text{-}LS}$ represent mean fitness functions of the algorithms.

$$\text{Hypothesis (1)} \qquad H_0 : \mu_{GA} \geqslant \mu_{PSO} \quad H_1 : \mu_{GA} < \mu_{PSO}, \tag{56}$$

$$\text{Hypothesis (2)} \qquad H_0 : \mu_{PSO} \geqslant \mu_{GPSO\text{-}LS} \quad H_1 : \mu_{PSO} < \mu_{GPSO\text{-}LS}. \tag{57}$$

The results are summarized in Table 8, where the P-value is given as the minimum level of significance to accept the hypothesis. To compare the proposed algorithms in terms of CPU time, we can carry out paired-samples t-tests similarly. The results for this comparison are summarized in Table 9.

From the results reported in Tables 8 and 9, it can be seen that for significant level of $\alpha = 0.05$, two hypotheses can be accepted for both the objective function and CPU time. This confirms that PSO is significantly better than GA and also that GPSO-LS is significantly better than PSO. Therefore, GPSO-LS is a more efficient algorithm in terms of objective function and CPU time for the proposed model.

Table 5
Data of items.

Item	D_k	P_k	A_{1k}	A_{2k}	C_{1k}	C_{2k}	h_k
1	3,679	4,983	29,180	7,684	36	26	214
2	3,826	4,568	30,239	6,151	33	23	363
3	3,972	5,129	22,834	8,073	37	24	315
4	3,985	4,847	32,979	10,073	38	21	393
5	4,927	4,998	22,388	11,781	32	24	494
6	3,984	5,136	33,187	8,143	38	24	205
7	4,636	4,935	22,388	9,781	31	24	294
8	4,446	3,984	22,425	11,941	40	23	289
9	4,690	3,970	36,691	11,065	37	26	430
10	3,135	3,046	32,318	6,760	33	20	453
11	5,782	4,690	30,356	7,865	37	21	352
12	4,168	3,829	31,700	6,838	38	27	408
13	5,292	4,381	21,832	10,814	39	21	391
14	5,872	4,297	21,545	6,910	32	26	461
15	4,819	4541	31,649	9,573	40	29	460

Table 6
Data of instances.

Instance No.	Number of items	Items
1	2	2, 9
2	3	6, 9, 10
3	4	4, 9, 10, 13
4	5	3, 5, 7, 10, 11
5	6	2, 4, 6, 9, 11, 14
6	7	3, 7, 8, 9, 10, 11, 12
7	8	1, 3, 6, 9, 11, 12, 13, 14
8	9	1, 2, 3, 5, 8, 9, 11, 12, 14
9	10	1, 2, 4, 6, 7, 8, 9, 11, 12, 15
10	11	1, 2, 3, 5, 6, 7, 8, 9, 11, 12, 14
11	12	1, 2, 3, 4, 5, 7, 8, 9, 10, 12, 13, 14
12	13	1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13, 14, 15
13	14	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15
14	15	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Table 7
Fitness function and CPU time of the instances.

Instance No.	GA		PSO		GPSO-LS	
	Fitness	CPU (s)	Fitness	CPU (s)	Fitness	CPU (s)
1	514,646	54.1	514,646	49.8	514,646	43.6
2	883,704	58.3	883,704	53.4	883,704	46.8
3	1,432,084	59.4	1,432,084	54.1	1,432,084	48.1
4	1,539,851	67.8	1,519,851	62.9	1,518,840	56
5	2,199,214	81.3	2,087,441	76	2,078,549	68.8
6	2,943,324	77.4	2,645,290	71.4	2,636,248	65.9
7	3,225,038	87.9	3,015,038	83.6	2,915,038	78.4
8	3,484,945	97.8	3,284,945	92.5	3,084,945	88.3
9	3,868,744	101.1	3,468,744	97	3,210,744	90.1
10	4,090,749	116.5	3,740,749	110.8	3,600,749	103.1
11	4,493,636	123.9	4,103,636	117.8	3,803,636	110.6
12	4,849,036	138.2	4,449,036	131.2	4,349,036	124.4
13	5,474,987	130.9	4,874,987	122.5	4,704,987	114.3
14	5,742,579	149.4	5,442,579	141.7	5,242,579	133.2

Table 8
One-sided paired-samples t-tests with respect to fitness function.

Hypothesis	*P*-value
Hypothesis 1	0.002
Hypothesis 2	0.018

Table 9
One-sided paired-samples t-tests with respect to CPU time.

Hypothesis	*P*-value
Hypothesis 1	0.018
Hypothesis 2	0.013

Consider instance No. 3 which consists of 4 items including items 4, 9, 10 and 13. Fig. 11 shows the convergence path of the fitness function for this instance illustrating the efficiency of the GPSO-LS algorithm in comparison to PSO and GA.

Table 10 contains the best solution (minimum total inventory cost), standard deviation for 20 times running of the GPSO-LS, lower bound, percentage gap between the best solution and the lower bound (Gap_L), upper bound and percentage gap between the best solution and the upper bound (Gap_U) for each instance. From the results shown in Table 10, one can see that:

- For instances 1–9, the standard deviation is zero. It denotes that the GPSO-LA is more stable for small size problems.
- For each instance, the best solution stands between the lower and upper bound. It confirms that the results of the GPSO-LS are trustworthy.

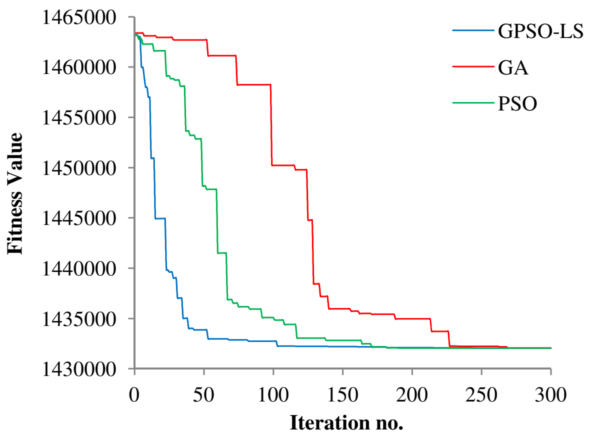

Fig. 11. The convergence path of fitness function for instance No. 3.

Table 10
Results of GPSO-LS.

Instance No.	Best solution	Standard dev.	Lower bound	Gap_L (%)	Upper bound	Gap_U (%)
1	514,646	0	368,264	40	769,485	33
2	883,704	0	709,385	25	1,029,486	14
3	1,432,084	0	1,039,385	38	2,014,948	29
4	1,518,840	0	1,110,385	37	2,110,384	28
5	2,078,549	0	1,828,364	14	2,998,270	31
6	2,636,248	0	2,268,075	16	3,140,385	16
7	2,915,038	0	2,673,947	9	3,918,938	26
8	3,084,945	0	3,018,460	2	4,014,929	23
9	3,210,744	0	3,044,805	5	4,114,927	22
10	3,600,749	12.1	3,001,457	20	4,362,947	17
11	3,803,636	18.0	3,100,691	23	5,093,834	25
12	4,349,036	11.7	3,122,957	39	5,749,274	24
13	4,704,987	21.5	3,410,940	38	5,947,837	21
14	5,242,579	24.3	4,052,945	29	6,513,947	20

6.1. Sensitivity analysis

The demand rate and the production rate of each item are the factors that affect the production capacity restriction. This restriction may be reduced through an increase in the production rate or (and) a decrease in the demand rate of each item. The effect of changes in the production rate and the demand rate on the minimum total inventory cost (*MTIC*) are investigated in this section. Sensitivity analysis is performed for instance No. 3.

In instance No. 3, we have $\frac{D_4}{P_4} + \frac{D_9}{P_9} + \frac{D_{10}}{P_{10}} + \frac{D_{13}}{P_{13}} = 4.24 > 1$. Hence, the production capacity is limited and a single machine cannot produce all four items. Then the proposed model is applied and the results are reported in Table 11 that includes the values of the decision variables for the best solution. In order to satisfy the demand of item 4, 1621 units are bought in each period from an outside supplier. This lot is received at beginning of the period, when the amount of inventory reaches a low of 195; and the machine starts to produce 570 units. The other quantities are interpreted as well. For this solution the total inventory cost is 1,432,082 \$/unit time (see Table 7). Table 12 contains the demand of each item which is met via production and ordering.

The results shown in Table 12 support the manager when he/she is faced with a decision to meet the demand of each item given that he/she does not intend to use the ordering strategy. In this situation there are just the demands shown in row 2 of Table 12 that must be met by using a single machine. This issue is a classic EPQ model which has a minimum total inventory cost of 311,101 \$/unit time. Although the total inventory cost is reduced, a part of the total demand of each item is lost. From an economics point of view, this decision is justified when the cost of lost sales is lower than the total inventory cost savings. In this example, if $2984\alpha_4 + 3511\alpha_9 + 2517\alpha_{10} + 4037\alpha_{13} < 1120983$, then meeting the demands by just using production is recommended (α_4, α_9, α_{10} and α_{13} denote lost sales costs of each unit of items 4, 9, 10 and 13, respectively).

6.1.1. Sensitivity analysis of the production rate

Let us assume that we increase the production rate of each item by 10–50%. Table 13 shows the percentage decrease in *MTIC* when the production rates of one or a combination of items is increased. As expected, more increase in the production rate causes more decrease in *MTIC*. These results assist the manager in making an efficient decision. For example, if the manager intends to increase the production rate of two items by 20% the data reported in Table 13 indicates that the best combination of items is items 9 and 10 in which *MTIC* is decreased by 8% or if he/she wants to decrease *MTIC* by 10% then a good decision is to increase the production rates of items 9, 10 and 13 by 20%.

In a practical situation, the possible way to increase the production rate of a machine is replacement of the machine by a new one. There may exist a newer machine built with a higher technology that can be substituted for the current machine used in this example. This new machine can increase the production rate of each item by 10%. The economic data needed for a replacement analysis is included in Table 14. Using the new machine makes a savings of 157,529 \$/ unit time. The payback period of the initial investment is 2 units of time. The payback period can provide an initial screening and it is not advised as the primary criterion to select an alternative machine [40]. Hence, a replacement analysis should be performed based on

Table 11
Decision variable values of the best solution for instance No. 3.

Item	4	9	10	13
Q_1	1621	1931	1384	2220
Q_2	570	648	340	690
R_1 or R_2	195	117	10	143

Table 12
The demand met via production and ordering for instance No. 3.

Item	4	9	10	13
Demand met via ordering (unit/unit time)	2948	3511	2517	4037
Demand met via production (unit/unit time)	1037	1179	618	1255
Total demand (unit/unit time)	3985	4690	3135	5292

Table 13
Sensitivity analysis of the production rate.

	Percentage increase in production rate				
Items	10%	20%	30%	40%	50%
4	−3.2	−6.4	−9.6	−12.8	−16
9	−2.7	−5.8	−8.9	−12	−15.1
10	−2.5	−5.5	−8.5	−11.5	−14.5
13	−3	−6.2	−9.4	−12.6	−15.8
4, 9	−5.1	−7.6	−10.1	−14.6	−16.1
4, 10	−5.6	−7.7	−10.2	−13.9	−15.8
4, 13	−4.8	−7.4	−11.4	−14.7	−17.2
9, 10	−5.4	−8	−12.1	−14.9	−17
9, 13	−4.7	−7.1	−9.6	−13.5	−16.4
10, 13	−5.1	−7.5	−11.8	−14.7	−17.1
4, 9, 10	−7.2	−9.5	−13.8	−17.8	−20.1
4, 9, 13	−8	−10.1	−14.3	−18.5	−22.4
4, 10, 13	−8.2	−10.4	−15.3	−19	−23.2
9, 10, 13	−7.9	−10	−14.7	−18.5	−20.7
4, 9, 10, 13	−11	−14.9	−19.2	−21	−25.3

Table 14
Economic Data of each machine.

	Current machine	New machine
Investment ($)	90,000 (market value)	300,000 (initial investment)
Operational cost ($/time unit)	40,000	30,000
Planning horizon (time unit)	5	5
Salvage value ($)	30,000	100,000
MTIC ($/time unit)	1,432,084	1,274,555

engineering economics for a more thorough investigation of the selection of the new machine. This analysis confirms that the replacement is justifiable with a minimum attractive rate of return (MARR) of 10%.

Suppose that the manager would like to relax the production capacity restriction through an equal increase in the production rates of a combination of items (Let E as set of this combination of items). For this purpose the increased production rate of item e, P'_e, will be $P'_e = \beta P_e$ which,

$$\beta = \frac{\sum_{\forall e \in E} \frac{D_e}{P_e}}{1 - \sum_{\forall k \in K-E} \frac{D_k}{P_k}}. \tag{58}$$

and K is set of all items.

Table 15 includes the value of β for instance No. 3. For example, the production capacity restriction could be relaxed if the production rate of all items is increased by 4.2 times that of the original values. A negative value of β is derived for cells that contain N/A. This implies that it is impossible to relax the production capacity restriction by using this combination.

6.1.2. Sensitivity analysis of the demand rate

Table 16 shows the effects of demand rate changes on *MTIC*. The demand rate of each item is changed from −50% to 50%. According to the results shown in Table 16, one can see that the same decreasing or increasing changes in the demand rates do not provide the same changes in *MTIC*. For example, −10% and 10% changes in the demand rate of item 4 result in −2.8% and 5.1% changes in MTIC, respectively. The manager may decide to decrease the demand rate of item 4 by 10% provided that the total cost reduction of 2.8% compensates for the 10% lost sales of item 4. The total cost reduction of 2.8% is equal to 40,098 $/unit time. According to the demand rate of item 4 which is 3985 unit/unit time, one can find that the breakeven of lost sales cost is 100.6 $/unit. That is, 10% decrease in the demand rate of item 4 is justified when the lost sales cost of each unit of item 4 is less than 100.6 $. On the other hand, a 10% increase in the demand rate of item 4 results in a 5.1%

Table 15
β for instance No. 3.

Items	β
4	N/A
9	N/A
10	N/A
13	N/A
4, 9	N/A
4, 10	N/A
4, 13	N/A
9, 10	N/A
9, 13	N/A
10, 13	N/A
4, 9, 10	N/A
4, 9, 13	N/A
4, 10, 13	N/A
9, 10, 13	19.2
4, 9, 10, 13	4.2

Table 16
Sensitivity analysis of the demand rate.

Percentage decrease in demand rate					
Items	−10%	−20%	−30%	−40%	−50%
4	−2.8	−3.5	−4.8	−6.3	−8.6
9	−3.1	−4.1	−5.2	−7.1	−9
10	−2.5	−3.2	−4.4	−6.1	−7.9
13	−3	−4.3	−5.1	−6.9	−8.8
4, 9	−4.6	−6.4	−7.3	−9.1	−11.2
4, 10	−5.2	−7.5	−8.1	−9.6	−12.1
4, 13	−5.6	−7.7	−8.4	−9.8	−12.4
9, 10	−5.7	−7.9	−8.9	−10	−12.8
9, 13	−6.2	−8.5	−9.3	−11.1	−13
10, 13	−6	−8.1	−8.9	−11	−12.9
4, 9, 10	−8.9	−11.8	−14.6	−17.5	−20
4, 9, 13	−8.3	−10.9	−13.4	−16.8	−19.6
4, 10, 13	−8	−10.1	−12.8	−15.7	−19
9, 10, 13	−8.3	−10	−12.1	−15	−18.9
4, 9, 10, 13	−12.1	−18.6	−21.3	−23.6	−27
Percentage increase in demand rate					
	10%	20%	30%	40%	50%
4	5.1	6.7	7.8	9.3	10.3
9	3.9	5.1	6.3	7.8	9.3
10	4.4	5.7	7.1	8.5	9.7
13	5	6.3	7.1	9	9.9
4, 9	7.6	8.8	10.4	13.5	15.7
4, 10	7.1	8.2	10.1	12.9	15
4, 13	7	8.1	9.8	12.1	14.6
9, 10	6.8	7.9	10	12.7	14.2
9, 13	7.5	8.6	10.1	12.9	15.1
10, 13	6.6	7.6	9.3	11.9	14.1
4, 9, 10	9.8	10.7	13.8	15.8	17.9
4, 9, 13	9.3	10.1	13.2	15.2	17.1
4, 10, 13	10	10.8	14	16	18.4
9, 10, 13	9.1	10	13.1	15	17.1
4, 9, 10, 13	13	15.3	17.8	24.9	28.1

increase in the total cost which is equal to 73,036 \$/unit time. It is justified when the profit of each unit of item 4 is more than 183.3 \$. One of the managerial implication of this finding is that the manager may increase the sales price of each unit of an item to cover the surplus cost when its demand rate is increased.

Suppose the manager intends to relax the production capacity restriction through an equal decrease in the demand rate of a combination of items (Let E as set of this combination of items). For this purpose the decreased demand rate of item e, D'_e will be $D'_e = \gamma D_e$ in which,

$$\gamma = \frac{1 - \sum_{\forall k \in K-E} \frac{D_k}{P_k}}{\sum_{\forall e \in E} \frac{D_e}{P_e}}. \tag{59}$$

Table 17
$100(1-\gamma)\%$ for instance No. 3.

Items	$100(1-\gamma)\%$
4	N/A
9	N/A
10	N/A
13	N/A
4, 9	N/A
4, 10	N/A
4, 13	N/A
9, 10	N/A
9, 13	N/A
10, 13	N/A
4, 9, 10	N/A
4, 9, 13	N/A
4, 10, 13	N/A
9, 10, 13	94.8%
4, 9, 10, 13	76.2%

$100(1-\gamma)\%$ is the percentage decrease in demand rate. Table 17 includes these values for the instance No. 3. A negative value of γ is derived for cells that contain N/A. This means that it is impossible to relax the production capacity restriction by this combination.

7. Conclusions

In this paper, a multi-item EPQ model with production capacity restriction is formulated. The optimum quantities for production and ordering are determined such that the total inventory cost is minimized and the optimal order and production quantities for each item are the decision variables. To solve the proposed model, GPSO-LS algorithm is utilized. To justify and validate the results obtained by the GPSO-LS algorithm, a genetic algorithm (GA) plus particle swarm optimization (PSO) were presented. In order to compare the performance of the three algorithms in terms of the fitness function and the required CPU time, they were first tuned using the Taguchi approach. The numerical comparison between the solutions obtained from the GPSO-LS algorithm and the lower (upper) bound confirmed that GPSO-LS is an effective algorithm.

One of the most important applications of the proposed model is in outsourcing strategy where the manager decides which and how much of each item must be outsourced. The procurement from an external supplier usually imposes more costs as compared with the costs of production inside the company. Therefore, the manager usually seeks ways to deal with production capacity restriction. He/she can reduce this restriction through an increase in the production rate. By sensitivity analysis, the percentage increase in the production rate of one or a combination of items can be determined to decrease the total cost by a certain percentage. A comparison between the total cost reduction and the cost of increase in the production rate supports the manager when he/she has to decide whether or not to increase the production rate of item(s). A decrease in the demand rate may be another way to mitigate production capacity restriction. However, it leads to the lost sales. The manager could make an efficient decision for the decrease in the demand rate by comparing the opportunity cost of the lost sales with the savings on the total cost gained from the decrease in the demand rate. At last, this research may be extended further by considering the proposed model in uncertain environments.

Acknowledgment

This work was supported by Research Grant No. 2/20826 from the Vice Chancellor of Research, Ferdowsi University of Mashhad, Mashhad, Iran. The authors would like to thank the respectable reviewers for their valuable comments and suggestions that improved the contents of this paper.

References

[1] S.F. Love, Inventory Control, McGraw-Hill, New York, 1979.
[2] A. Andriolo, D. Battini, R.W. Grubbström, A. Persona, F. Sgarbossa, A century of evolution from Harris's basic lot size model: survey and research agenda, Int. J. Prod. Econ. 155 (2014) 16–38.
[3] G. Hadley, T.M. Whitin, Analysis of Inventory Systems, Prentice Hall Inc., New Jersey, 1963.
[4] A.M.M. Jamal, B.R. Sarker, S. Mondal, Optimal manufacturing batch size with rework process at a single-stage production system, Comput. Ind. Eng. 47 (2004) 77–89.
[5] G. Woolsey, A requiem for the EOQ: an editorial, Hosp. Mater. Manage. Q. 12 (1990) 82–90.
[6] M.Y. Jaber, R.Y. Nuwayhid, M.A. Rosen, Price-driven economic order systems from a thermodynamic point of view, Int. J. Prod. Res. 42 (2004) 5167–5184.
[7] M.D. Byrne, Multi-item production lot sizing using a search simulation approach, Eng. Costs Prod. Econ. 19 (1990) 307–311.
[8] H. Hwang, D.B. Kim, Y.D. Kim, Multi-product economic lot size models with investment costs for setup reduction and quality improvement, Int. J. Prod. Res. 31 (1993) 691–703.

[9] M.Y. Jaber, M. Bonney, The economic manufacture/order quantity (EMQ/EOQ) and the learning curve: past, present, and future, Int. J. Prod. Econ. 59 (1999) 93–102.
[10] J.C. Fransoo, V. Sridharan, J.W.M. Bertrand, A hierarchical approach for capacity coordination in multiple products single-machine production systems with stationary stochastic demands, Eur. J. Oper. Res. 86 (1995) 57–72.
[11] B.W. Kreng, S.Y. Wu, Operational flexibility and optimal total production cost in multiple-item economic production quantity models, Int. J. Syst. Sci. 31 (2000) 255–261.
[12] S. Choi, J.S. Noble, An integrated material flow system approach for determining the economic production quantity (EPQ), Int. J. Prod. Res. 38 (2000) 3485–3511.
[13] K. Das, T.K. Roy, M. Maiti, Multi-item inventory model with quantity-dependent inventory costs and demand-dependent unit cost under imprecise objective and restrictions: a geometric programming approach, Prod. Plan. Control 11 (2000) 781–788.
[14] K. Maity, M. Maiti, Possibility and necessity constraints and their defuzzification – a multi-item production-inventory scenario via optimal control theory, Eur. J. Oper. Res. 177 (2007) 882–896.
[15] S. Sharma, A fresh approach to performance evaluation in a multi-item production scenario, Eur. J. Oper. Res. 178 (2007) 627–630.
[16] K.L. Hou, An EPQ model with setup cost and process quality as functions of capital expenditure, Appl. Math. Model. 31 (2007) 10–17.
[17] S. Islam, T.K. Roy, Fuzzy multi-item economic production quantity model under space constraint: a geometric programming approach, Appl. Math. Comput. 184 (2007) 326–335.
[18] S.H.R. Pasandideh, S.T.A. Niaki, J. Aryan Yeganeh, A parameter-tuned genetic algorithm for multi-product economic production quantity model with space constraint, discrete delivery orders and shortages, Adv. Eng. Software 41 (2010) 306–314.
[19] S.H.R. Pasandideh, S.T.A. Niaki, S.S. Mirhosseyni, A parameter-tuned genetic algorithm to solve multi-product economic production quantity model with defective items, rework, and constrained space, Int. J. Adv. Manuf. Technol. 49 (2010) 827–837.
[20] A. Taleizadeh, S.T.A. Niaki, A.A. Najafi, Multiproduct single-machine production system with stochastic scrapped production rate, partial backordering and service level constraint, J. Comput. Appl. Math. 233 (2010) 1834–1849.
[21] A. Taleizadeh, A.A. Najafi, S.T.A. Niaki, Economic production quantity model with scrapped items and limited production capacity, Scienta Iranica Trans. E Ind. Eng. 17 (2010) 58–69.
[22] A. Taleizadeh, H.M. Wee, S.J. Sadjadi, Multi-product production quantity model with repair failure and partial backordering, Comput. Ind. Eng. 59 (2010) 45–54.
[23] A. Taleizadeh, H.M. Wee, S.G. Jalali-Naini, Economic production quantity model with repair failure and limited capacity, Appl. Math. Model. 37 (2013) 2765–2774.
[24] S. Mandal, K. Maity, S. Mondal, M. Maiti, Optimal production inventory policy for defective items with fuzzy time period, Appl. Math. Model. 34 (2010) 810–822.
[25] A. Taleizadeh, S.J. Sadjadi, S.T.A. Niaki, Multiproduct EPQ model with single machine, backordering and immediate rework process, Eur. J. Ind. Eng. 5 (2011) 388–411.
[26] A.K. Maity, One machine multiple-product problem with production inventory system under fuzzy inequality constraint, Appl. Soft Comput. 11 (2011) 1549–1555.
[27] K.M. Bjork, A multi-item fuzzy economic production quantity problem with a finite production rate, Int. J. Prod. Econ. 135 (2012) 702–707.
[28] S.D. Lee, C.M. Yang, An economic production quantity model with a positive resetup point under random demand, Appl. Math. Model. 37 (2013) 3340–3354.
[29] M.A. Hariga, Economic production-ordering quantity models with limited production capacity, Prod. Plan. Control 9 (1998) 671–674.
[30] J. Heizer, B. Render, Production and Operation Management, fourth ed., Prentice Hall, New Jersey, 1995.
[31] C.H. Glock, M.Y. Jaber, S. Zolfaghari, Production planning for a ramp-up process with learning in production and growth in demand, Int. J. Prod. Res. 50 (2012) 5707–5718.
[32] S. Axsater, Inventory Control, second ed., Springer, New York, 2006.
[33] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp. 1942–1945.
[34] C.Y. Dye, T.P. Hsieh, A particle swarm optimization for solving joint pricing and lot-sizing problem with fluctuating demand and unit purchasing cost, Comput. Math. Appl. 60 (2010) 1895–1907.
[35] M. Gen, R. Cheng, Genetic Algorithms and Engineering Design, Wiley, New York, 1997.
[36] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, third ed., Springer, 1996.
[37] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[38] G. Taguchi, Introduction to quality engineering, White Plains: Asian Productivity Organization/UNIPUB, 1986.
[39] M. Rostamian-Delavar, M. Hajiaghaei-Keshteli, S. Molla-Alizadeh-Zavardehi, Genetic algorithms for coordinated scheduling of production and air transportation, Expert Syst. Appl. 37 (2010) 8255–8266.
[40] L. Blank, A. Tarquin, Engineering Economy, sixth ed., McGraw Hill, 2005.