View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Directory of Open Access Journals

International Journal of Industrial Engineering Computations 2 (2011) 213–224

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Multi products single machine economic production quantity model with multiple batch size

Ata Allah Taleizadeh[a], **Gede Agus Widyadana**[b], **Hui Ming Wee**[b*] and **Jahangir Biabani**[c]

[a]*Department of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran*
[b]*Department of Industrial and System Engineering, Chung Yuan Christian University, Chung Li 32023, Taiwan*
[c]*Department of Economic, Payame Nour University, Tehran Center, Iran*

A R T I C L E I N F O

Article history:
Received 10 November 2010
Received in revised form
4 January 2011
Accepted 6 January 2011
Available online
6 January 2011

Keywords:
Inventory control
EPQ
Multi-product multi-constraint
Multi deliveries
Meta heuristic
Extended cutting plane

A B S T R A C T

In this paper, a multi products single machine economic production quantity model with discrete delivery is developed. A unique cycle length is considered for all produced items with an assumption that all products are manufactured on a single machine with a limited capacity. The proposed model considers different items such as production, setup, holding, and transportation costs. The resulted model is formulated as a mixed integer nonlinear programming model. Harmony search algorithm, extended cutting plane and particle swarm optimization methods are used to solve the proposed model. Two numerical examples are used to analyze and to evaluate the performance of the proposed model.

© 2011 Growing Science Ltd. All rights reserved

1. Introduction

Economic production quantity plays an important role on managing the inventory. Taft (1918) introduced the popular economic production quantity (EPQ) model. Salameh and Jaber (2000) introduced an EPQ model with imperfect quality items, and the work was extended by Goyal and Cardenas-Barron (2002) who introduced an efficient solution procedure. Teng and Chang (2005) developed an EPQ model for deteriorating items with displayed stocks and price dependent demand. Huang (2005) developed an EPQ model with service level constraint and random defective rate. Teng et al. (2005) studied an EPQ model with time-varying demand and cost. Freimer et al. (2006) studied the effects of imperfect yield on an EPQ model with time-varying proportion defective items and additional repair cost. An analytical method to solve an EPQ model with varying lead times and backorder was proposed by Lai et al. (2006). Leung (2007) developed an EPQ model with flexible and reliable production systems. Liao et al. (2007) developed a production inventory model for deteriorating items with finite production rate and postponed payment. Islam and Roy (2007) considered an EPQ model with cost-dependent demand, space constraint and fuzzy parameters. Darwish (2008) developed EPQ models to consider different setup costs with backordering. Ouyang

* Corresponding author. Tel: +886-3-2654409; Fax:+886-3-2654499
E-mail: weehm@cycu.edu.tw (H. M. Wee)

© 2011 Growing Science Ltd. All rights reserved.
doi: 10.5267/j.ijiec.2011.01.002

and Rau (2008) developed an EPQ model with linear and exponentially decreasing unit production costs. Teng and Chang (2009) derived an optimal cycle length for an EPQ model under two levels of trade credit policy. Pal et al. (2009) developed two EPQ models with price discount promotional demand in fuzzy and crisp environment. Hu and Lio (2010) developed an EPQ model with shortage, postponed payment and finite replenishment rate.

In recent years, there have been different research works on vendor-buyer inventory models with multiple deliveries. Goyal and Nebebe (2000) developed a vendor-buyer inventory model where the buyer receives a batch quantity in *n* shipments. Chung and Wee (2007) developed an integrated supplier-buyer inventory model for deteriorating items with multiple deliveries. Su et al. (2007) developed an integrated supplier-retailer inventory model with two-level trade credit strategy. In their model, the retailer determines the optimal order quantity and the supplier determines the optimal number of shipment per production run. Haksever (2008) developed a mixed-integer programming model for multi products problem where the buyer orders from supplier who offers incremental quantity discounts. Pasandideh and Niaki (2008) introduced an EPQ model with discrete deliveries and space constraint.

In this research, we extend the model originally presented by Pasandideh and Niaki (2008) by considering multi products single machine system with capacity and space constraints. Our study is organized as follows. In the first section, a comprehensive literature review and background of the model are presented. Section 2 demonstrates the model development and the section 3 presents the solution method. Section 4 shows two examples to illustrate the model; concluding remarks are derived and future research topics are suggested in section 5.

2. Model development

In our model, we assume that production and demand rates of each product are known and constant. Manufacture sends orders to the customer and bears the transportation cost for each delivery to the customer. The customer determines the capacity of each delivery and the quantity of each shipment. Shortage is not permitted and the production costs consist of production, setup, holding, and transportation costs. Since all products are manufactured by a single machine with a limited capacity, a unique cycle length for all items is considered, i.e, $T_1 = T_2 = \cdots = T_n = T$ (Taleizadeh et al. 2010a, 2010b, 2010c). The purpose of this paper is to determine the optimal replenishment period, the delivery quantity and the number of delivery to minimize the total production inventory cost with space and capacity constraints.

To model the problem for $i = 1,...,n$, we use the following parameters.

n : number of products,

q_i : order quantity for i^{th} product,

r_i^p : production rate of i^{th} product,

r_i^d : demand rate of i^{th} product,

T : cycle length for all products,

t_i^p : production time in each cycle of i^{th} product,

t_i^d : down time in each cycle of i^{th} product,

ts_i :machine setup time to produce the i^{th} product,

t_i : time between two sequential shipments of each pallet for i^{th} product,

v_i : quantity order for i^{th} product,

n_i^o : number of shipments in each cycle of i^{th} product,

c_i^t : transportation cost of a shipment for i^{th} product,

A_i : set-up cost of each cycle for i^{th} product,

c_i^h : holding cost per unit of i^{th} product,

c_i^p : production cost per unit of i^{th} product,

CH : total holding costs per year,

CT : total transportation costs per year,

CP : total providence costs per year,

CA : total set-up costs per year,

TC : total costs per year.

Fig. 1 shows the inventory level of the EPQ model with discrete delivery order. In this research, manufacture delivers order of product *i*, to the customer in n_i^o times shipments with v_i units in each delivery. Finally, the model will be extended to multi products problem. From Fig. 1, during t_i^p, a pallet with capacity of k_i is delivered to the company with n_i^o jumps. During t_i^p and t_i^d, the delivered products are produced at a constant rate (Pasandideh & Niaki 2008):

$$q_i = n_i^o v_i \,. \tag{1}$$

We develop a single-product model for i^{th} product. The production cycle length is the summation of the production uptimes and the production downtimes, and we have,

$$T = t_i^p + t_i^d \,. \tag{2}$$

Also we have,

$$q_i = r_i^d T \,. \tag{3}$$

Using Eq. (1) to Eq. (3), the total replenishment time can be modeled as follows,

$$T = t_i^p + t_i^d = \frac{n_i^o v_i}{r_i^d} \,. \tag{4}$$

Since the maximum inventory level is $n_i^o v_i - (n_i^o - 1) v_i \dfrac{r_i^d}{r_i^p}$, we have $t_i^d = \dfrac{n_i^o v_i}{r_i^d} - (n_i^o - 1)\dfrac{v_i}{r_i^p}$. The production up time is as follows,

$$t_i^p = (n_i^o - 1)\frac{v_i}{r_i^p} \ . \tag{5}$$

2.1. Objective function

The total cost function consists of the sum of the production, the setup, the holding and the transportation costs as follows,

$$TC = CA + CP + CT + CH \ . \tag{6}$$

The setup cost (A_i) occurs N times per year. Therefore, the annual setup cost is as follows,

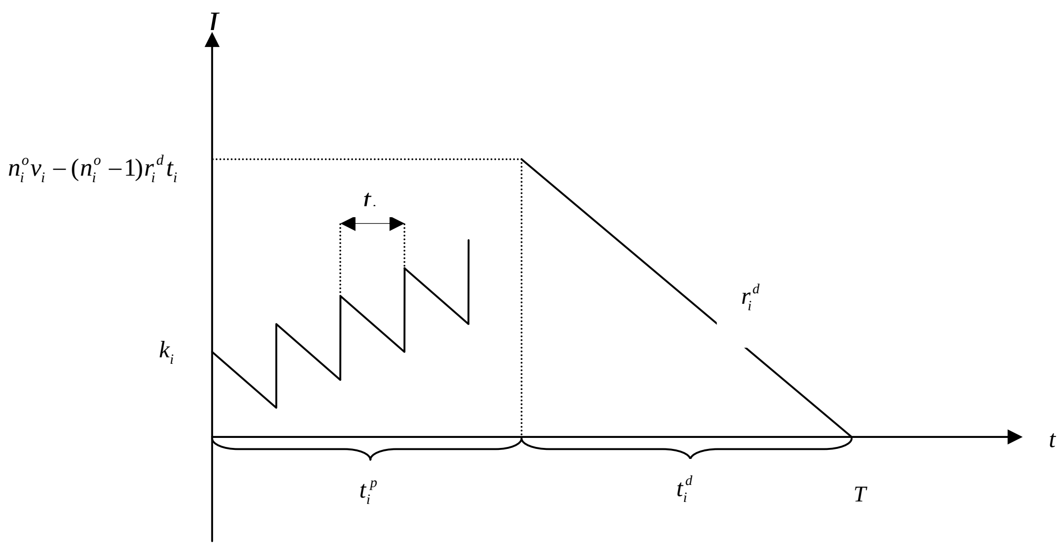

Fig. 1. The inventory level

$$CA = \sum_{i=1}^{n} NA_i \ . \tag{7}$$

For joint policy, we have $N = \frac{1}{T}$, thus

$$CA = \frac{\sum_{i=1}^{n} A_i}{T} . \tag{8}$$

The production cost per unit and the production quantity per period of the i^{th} product are c_i^p and q_i, respectively. Hence, the production cost of i^{th} product per period is $c_i^p q_i$ and the annual production quantity is $Nc_i^p q_i$. Finally, the joint production cost is as follows,

$$CP = \frac{\sum_{i=1}^{n} c_i^p n_i^o k_i}{T} = \frac{\sum_{i=1}^{n} c_i^p r_i^d T}{T} = \sum_{i=1}^{n} c_i^p r_i^d \ . \tag{9}$$

Transportation cost depends on the number of shipments and it is equal to $c_i^t n_i^o$ for each cycle and the annual transportation cost is $Nc_i^t n_i^o$. Finally the CT in joint policy can be modeled as follows,

$$CT = \frac{\sum_{i=1}^{n} c_i^t n_i^o}{T} . \tag{10}$$

According to Fig. 1, each cycle has two sections (t_i^p and t_i^d) and t_i^d is built up by a collection of trapezes. The number of trapezes for product i is $m_i - 1$. If az_i^j represents the area of trapeze j of product i, the areas of trapeze 1 and 2 are as follows,

$$az_i^1 = (\frac{v_i + (v_i - r_i^d t_i)}{2})t_i = (\frac{2v_i - r_i^d t_i}{2})t_i \ , \tag{11}$$

and

$$az_i^2 = \left(\frac{(v_i - r_i^d t_i + v_i) + (2v_i - 2r_i^d t_i)}{2}\right)t_i = (\frac{4v_i - 3r_i^d t_i}{2})t_i \ , \tag{12}$$

respectively. We have,

$$az_i^j = (\frac{2jv_i - (2j-1)r_i^d t_i}{2})t_i \quad ; \quad j = 1,\ldots,m_i - 1 \tag{13}$$

Finally, the area of all trapezes on the left of each cycle for i^{th} product can be formulated as follows,

$$\begin{aligned} az_i^j &= \sum_{j=1}^{m_i-1}\frac{2jv_i t_i - 2jr_i^d t_i^2 + r_i^d t_i^2}{2} = \sum_{j=1}^{m_i-1}\frac{2jv_i t_i}{2} - \sum_{j=1}^{m_i-1}\frac{2jr_i^d t_i^2}{2} + \sum_{j=1}^{m_i-1}\frac{r_i^d t_i^2}{2} \\ &= v_i t_i \sum_{j=1}^{m_i-1} j - r_i^d t_i^2 \sum_{j=1}^{m_i-1} j + \frac{r_i^d t_i^2}{2}\sum_{j=1}^{m_i-1} 1 = v_i t_i \frac{n_i^o\left(n_i^o - 1\right)}{2} - r_i^d t_i^2 \frac{n_i^o\left(n_i^o - 1\right)}{2} + \frac{r_i^d t_i^2}{2}\left(n_i^o - 1\right) \\ &= v_i t_i \frac{n_i^o\left(n_i^o - 1\right)}{2} - r_i^d t_i^2 \frac{n_i^{o2}}{2} + r_i^d t_i^2 \frac{n_i^o}{2} + \frac{r_i^d t_i^2}{2}\left(n_i^o - 1\right) \end{aligned} \tag{14}$$

For $k_i = r_i^p t_i$ one has,

$$az_i^j = \left(\frac{r_i^p - r_i^d}{2r_i^{p2}}\right)n_i^{o2}v_i^2 + \left(\frac{r_i^d - r_i^p}{r_i^{p2}}\right)n_i^o v_i^2 - \frac{r_i^d}{2r_i^{p2}}v_i^2 \tag{15}$$

The area of a triangle on the right side of each cycle of product i, (at_i) can be modeled as follows:

$$\begin{aligned} at_i &= \frac{1}{2}(n_i^o v_i - (n_i^o - 1)r_i^d t_i)(\frac{n_i^o v_i}{r_i^d} - (n_i^o - 1)t_i) \\ &= \frac{1}{2}\left[\frac{\left(n_i^o v_i\right)^2}{r_i^d} - 2n_i^o v_i (n_i^o - 1)t_i + (n_i^o - 1)^2 r_i^d t_i^2\right] = \frac{1}{2}\left[\frac{\left(n_i^o v_i\right)^2}{r_i^d} - 2(n_i^o - 1)n_i^o \frac{v_i^2}{r_i^p} + (n_i^o - 1)^2 r_i^d \frac{v_i^2}{r_i^{p2}}\right] \\ &= \left(\frac{1}{2r_i^d} - \frac{1}{r_i^p} + \frac{r_i^d}{2r_i^{p2}}\right)\left(n_i^o v_i\right)^2 + \left(1 - \frac{r_i^d}{r_i^{p2}}\right)n_i^o v_i^2 + \frac{r_i^d}{2r_i^{p2}}v_i^2 \end{aligned} \tag{16}$$

The total areas of each cycle of product i, (s_i) is as follows,

$$\begin{aligned} s_i = az_i + at_i &= \left(\frac{r_i^p - r_i^d}{2r_i^{p2}}\right)n_i^{o2}v_i^2 + \left(\frac{r_i^d - r_i^p}{r_i^{p2}}\right)n_i^o v_i^2 - \frac{r_i^d}{2r_i^{p2}}v_i^2 \\ &\quad + \left(\frac{1}{2r_i^d} - \frac{1}{r_i^p} + \frac{r_i^d}{2r_i^{p2}}\right)\left(n_i^o v_i\right)^2 + \left(1 - \frac{r_i^d}{r_i^{p2}}\right)n_i^o v_i^2 + \frac{r_i^d}{2r_i^{p2}}v_i^2 \\ &= \left(\frac{r_i^p - r_i^d}{2r_i^p r_i^d}\right)\left(n_i^o v_i\right)^2 + \left(\frac{r_i^{p2} - r_i^p}{r_i^{p2}}\right)n_i^o v_i^2 \end{aligned} \tag{17}$$

Using $k_i = r_i^p t_i$ and Eq. (16), and assuming N periods per year yields the total annual holding cost as $Nc_i^h\left(\left(\frac{r_i^p - r_i^d}{2r_i^p r_i^d}\right)\left(n_i^o v_i\right)^2 + \left(\frac{r_i^{p2} - r_i^p}{r_i^{p2}}\right)n_i^o v_i^2\right)$. Finally, the holding cost for joint production system is:

$$CH = \sum_{i=1}^{n} c_i^h \left(\frac{r_i^p - r_i^d}{2r_i^p r_i^d}\right)\frac{n_i^{o2} v_i^2}{T} + \sum_{i=1}^{n} c_i^h \left(\frac{r_i^{p2} - r_i^p}{r_i^{p2}}\right)\frac{n_i^o v_i^2}{T}. \tag{17}$$

Based on Eq. (4), Eq. (8), Eq. (9), Eq. (10), and Eq. (17), and implementing some simplifications, the total annual cost of production system can be modeled as follows,

$$TC = \frac{\sum_{i=1}^{n} A_i}{T} + \sum_{i=1}^{n} c_i^t \frac{n_i^o}{T} + \sum_{i=1}^{n} c_i^h \left(\frac{r_i^p - r_i^d}{2r_i^p r_i^d}\right)\frac{n_i^{o2} v_i^2}{T} + \sum_{i=1}^{n} c_i^h \left(\frac{r_i^{p2} - r_i^p}{r_i^{p2}}\right)\frac{n_i^o v_i^2}{T} + \sum_{i=1}^{n} c_i^p r_i^d \tag{18}$$

2.2. Constraints

For joint production systems, the total production and setup times must be smaller than the cycle length. In our model, $\sum_{i=1}^{n}(t_i^p + ts_i)$ must be smaller or equal to T . Therefore, the capacity limitation can be modeled as follows,

$$\sum_{i=1}^{n}(t_i^p + ts_i) \le T \tag{19}$$

Based on Eq. (4), Eq. (5) and Eq. (19), one has:

$$\sum_{i=1}^{n}(n_i^o - 1)\frac{v_i}{r_i^p} + \sum_{i=1}^{n} ts_i \le T. \tag{20}$$

The number of shipments must be smaller than the upper bound and, at least, one shipment needs to be performed. One has:

$$1 \le n_i^0 \le U_i; \text{Integer}; i = 1, \cdots, n \tag{21}$$

Finally, the complete model can be derived as follows,

$$\min: \mathrm{TC} = T^{-1}\sum_{i=1}^{n} A_i + \sum_{i=1}^{n} c_i^t \frac{n_i^0}{T} + \sum_{i=1}^{n}\left(\frac{r_i^p - r_i^d}{2r_i^p r_i^d}\right)\frac{n_i^{0^2} v_i^2}{T} + \sum_{i=1}^{n}\left(\frac{r_i^{p^2} - r_i^p}{r_i^{p^2}}\right)\frac{n_i^0 v_i^2}{T} + \sum_{i=1}^{n} c_i^p r_i^d$$

subject to

$$\sum_{i=1}^{n}(n_i^0 - 1)\frac{v_i}{r_i^p} + ts_i \le T, \tag{22}$$

$$1 \le n_i^0 \le U_i; \text{Integer}; i = 1, \cdots, n$$

$$T, v_i \ge 0\ ;\ i = 1, 2, \ldots, n$$

3. Solution method

The final model in Eq. (22) is a mixed integer nonlinear programming (MINLP) problem. Westerlund and Pettersson (1995) extended cutting plane method to solve MINLP and we use their method to solve the proposed model of this paper. In addition, in order to evaluate the performance of the proposed solution method, we use two meta-heuristic algorithms described in section 3.1 and 3.2.

3.1. Particle swarm optimization

Kennedy and Eberhart (1995) proposed particle swarm optimization (PSO) in the mid 1990s. PSO is inspired by flocks of birds (Kennedy & Eberhart, 2001). The proposed PSO algorithm consists of three main steps; at first, the positions of particle are generated. Secondly, exploration velocity is updated, and finally each position is updated. These parts are described in the following section. In an optimization problem, each particle refers to a point in the solution space that changes its position from one move (iteration) to another, based on exploration velocity updates. The type of particles is associated with the number of variables involved in a problem (Taleizadeh et al., 2010d). In this research, there are three decision variables (T, v_i, and n_i^o) for each product.

The swarm size is denoted by N. The positions and exploration velocities are given in a vector format where the superscript and subscript denote i^{th} particle in the population at k^{th} iteration (generation). "Rand" is a uniformly distributed random variable that can take any value between 0 and 1. This initialization process allows the swarm particles to be generated randomly across the design space. In order to initial the particles, we use Eq. (23) and Eq. (24), in which Δt is a constant time increment. $X_{\min}$ and $X_{\max}$ are the upper and the lower bounds on the design variables' values. X_k^i are the positions and V_k^i are the exploration velocities (Taleizadeh et al., 2010d).

$$X_0^i = X_{\min} + Rand\left(X_{\max} - X_{\min}\right) \tag{23}$$

$$V_0^i = \frac{X_{\min} + \text{Rand}(X_{\max} - X_{\min})}{\Delta t} = \frac{\text{Position}}{\text{time}} \tag{24}$$

In order to update the exploration velocity, the formula in Eq. (25) is used where *Rand*, represents a random variable distributed, uniformly. The updated velocity depends on three weight factors, namely, inertia factor, w, self confidence factor, C_1, and swarm confidence factor, C_2. The updated velocity can be modeled as follows,

$$\underbrace{V_{k+1}^i}_{\substack{\text{Velocity of Particle} \\ i \text{ at time } k+1}} = \overbrace{w}^{[0.4,1.4]} \underbrace{V_k^i}_{\substack{\text{Current} \\ \text{Motion}}} + \underbrace{\overbrace{C_1}^{[1.5,2]} Rand \frac{(P^i - X_k^i)}{\Delta t}}_{\text{Particle Memory Influence}} + \underbrace{\overbrace{C_2}^{[2,2.5]} Rand \frac{(P_k^g - X_k^i)}{\Delta t}}_{\text{Swarm Influence}} \tag{25}$$

In this research, we use $C_1 = C_2 = 2$ and $N = 100$. In order to update the positions, we used Eq. (26) which is a function over the iteration number.

$$X_{K+1}^i = X_K^i + V_{K+1}^i \Delta t \tag{26}$$

3.2 Harmony search algorithm

The harmony search (HS) algorithm is inspired from the act of musician groups (Geem et al., 2001). This algorithm seeks the optimum solution by generating random vector solutions in a harmony memory (HM) which are improved with some pitch adjusting and updating methods. In summary fantastic harmony is considered as global optimum, aesthetic standard is determined by the objective function, and pitches of instruments are desired values of the variables. The proposed HS algorithm from Taleizadeh et al. (2008) consists of three main steps; 1) parameter and harmony memory initialization, 2) new harmony generation, 3) harmony memory updates. The constant parameters of the HS algorithm include harmony memory size (HMS), harmony memory considering rate ($HMCR$), pitch adjusting rate (PAR), number of decision variables (N), and the maximum number of improvisations (NI). The HM is initialized with randomly generated solutions in a specific range limited by upper and lower bounds determined by the problem.

New harmony improvisation is based on three rules: (i) random selection (ii) HM consideration, and (iii) pitch adjustment. In random selection rule, the new value of each decision variable is randomly chosen within the allowable range of the vector solution. In HM algorithm, the random is chosen from HM with probability $HMCR$ and the random selection is performed with probability ($1-HMCR$) (Taleizadeh et al., 2008). In pitch adjustment, every component obtained by the memory consideration is examined to determine whether it should be pitch adjusted or not. The value of the decision variable is changed by Eq. (27) with probability of PAR , and it is kept without any changes with probability $1-PAR$. In Eq. (23) the BW stands for band width and denotes the amount of change for pitch adjustment. Also, $rand$ is a uniform random number between 0 and 1. For each component of the vector, the selection for increasing or decreasing are carried out with the same probability (Taleizadeh et al., 2008).

$$\mathbf{X}^{'} = \mathbf{X}^{'} \pm (rand)(BW); \quad rand \sim U\left[0,1\right] \tag{27}$$

The constraint handling part of the algorithm is performed before the HM update. The constraint handling part checks whether these constraints are satisfied or not. If they are satisfied, then the HM is updated. In this stage, if the new fitness value is better than the worst case in the HM, the worst harmony vector is replaced by the new solution vector. The last step in a HS method is to check if the algorithm has found a solution that is good enough to meet user's expectations. In this research, we use $HMS=10,\ HMCR=0.95,\ PAR=0.7,\ NI=1000$.

4. Numerical Examples

We consider two multi-products EPQ problems with discrete deliveries and capacity constraint with fifteen products. In the examples, the demand rate, the production rate, and the setup time of each product are assumed to be constant for each cycle. There are no scraped and defective items during the process.

Table 1
General data for the example 1

Product	r_i^d	r_i^p	ts_i	A_i	c_i^t	c_i^h	c_i^p	U_i
1	300	5000	0.0010	500	5	2	34	10
2	350	5500	0.0015	600	7	4	32	10
3	400	6000	0.0020	700	9	6	30	10
4	450	6500	0.0025	800	11	8	28	10
5	500	7000	0.0030	900	13	10	26	10
6	550	7500	0.0035	1000	15	12	24	10
7	600	8000	0.0040	1100	17	14	22	10
8	650	8500	0.0045	1200	19	16	20	10
9	700	9000	0.0050	1300	21	18	18	10
10	750	9500	0.0055	1400	23	20	16	10
11	800	10000	0.0060	1500	25	22	14	10
12	850	10500	0.0065	1600	27	24	12	10
13	900	11000	0.0070	1700	29	26	10	10
14	950	11500	0.0075	1800	31	28	8	10
15	1000	12000	0.0080	1900	33	30	6	10

Table 2
General data for the example 2

Product	r_i^d	r_i^p	ts_i	A_i	c_i^t	c_i^h	c_i^p	U_i
1	500	5000	0.0010	500	200	34	480	10
2	550	5500	0.0015	600	200	32	460	10
3	600	6000	0.0020	700	200	30	440	10
4	650	6500	0.0025	800	200	28	420	10
5	700	7000	0.0030	900	200	26	400	10
6	750	7500	0.0035	1000	150	24	380	10
7	800	8000	0.0040	1100	150	22	360	10
8	850	8500	0.0045	1200	150	20	340	10
9	900	9000	0.0050	1300	150	18	320	10
10	950	9500	0.0055	1400	150	16	300	10
11	1000	10000	0.0060	1500	100	14	280	10
12	1050	10500	0.0065	1600	100	12	260	10
13	1100	11000	0.0070	1700	100	10	240	10
14	1150	11500	0.0075	1800	100	8	220	10
15	1200	12000	0.0080	1900	100	6	200	10

The general data of the examples are given in Tables 1 and 2. The minimum shipment is assumed to be 1 and the maximum shipment is equal to 10. Table 3 shows the best results for the first example using the extended cutting plane method, PSO and HS algorithms, respectively. Table 4 shows the best results for the second example.

Table 3
Best results for the example 1 by extended cutting plane, PSO and HS

Method	Cutting Plane		PSO		HS	
Product	n_i^o	v_i	n_i^o	v_i	n_i^o	v_i
1	10	20.693	10	21.3	9	22.1
2	10	17.776	9	19.7	10	21.6
3	10	15.580	10	17.4	8	20.8
4	10	13.867	9	16.3	9	19.6
5	10	12.493	10	14.3	10	17.9
6	10	11.367	10	12.7	9	16.4
7	10	10.427	9	11.4	10	13.2
8	10	9.631	10	10.5	9	12.6
9	10	8.948	9	9.6	9	11.5
10	10	8.355	8	8.9	10	10.8
11	10	7.836	10	8.1	7	9.1
12	10	7.377	10	7.6	9	8.2
13	10	6.969	10	6.8	9	7.4
14	10	6.604	10	6.5	10	6.4
15	10	6.276	8	6.1	10	6.1
	T=3.308	TC=179,607	T=3.018	TC=181,640	T=2.921	TC=184,210

According to Table 3 and Table 4, the cutting plane method obtains lower total cost compared with other methods. Furthermore, in terms of the CPU time, the computation time of the extended cutting plane method is less than the other two methods. The CPU time of the cutting plane method for the first example is 4 seconds and for the second example is 5 seconds. For the first example, the average CPU times are 24 and 31 seconds for PSO and HS, respectively. In the second example, the average CPU times are 25 and 30 seconds. Each method is performed 20 times, while the corresponding standard deviations of the CPU time are 1.414 and 2 seconds for PSO and HS, respectively. For the second example, the numbers of the runs are 20 and the CPU standard deviations are 1.414 and 1.732 seconds.

Table 4
Best results for the example 2 by extended cutting plane, PSO and HS

Method	Cutting Plane		PSO		HS	
Product	n_i^o	v_i	n_i^o	v_i	n_i^o	v_i
1	1	56.069	2	60.1	1	64.8
2	1	50.975	1	55.8	1	60.1
3	1	46.730	2	49.0	1	56.6
4	1	43.137	1	45.4	1	52.3
5	1	40.058	1	41.6	1	50.1
6	1	37.388	1	38.6	1	47.2
7	1	35.053	1	36.2	1	44.7
8	1	32.992	1	34.1	1	41.9
9	1	31.159	2	32.0	1	39.6
10	1	29.520	2	29.9	1	37.7
11	1	28.045	3	28.6	1	35.2
12	1	26.710	1	25.3	1	32.9
13	1	25.496	1	23.1	1	30.2
14	1	24.388	1	22.3	1	29.1
15	1	23.372	1	21.9	1	28.2
	T=0.772	TC=4,111,100	T=0.685	TC=4,131,700	T=0.651	TC=4,168,500

5. Conclusion and future research

This paper has presented an EPQ model with multiple discrete deliveries, capacity and space constraints. The primary purpose of this research is to determine the optimal period length, the optimal number of shipments and the optimal order quantities. In order to solve the problem, we applied the extended cutting plane method, the particle swarm optimization and harmony search algorithms. Two numerical examples with fifteen products are used to illustrate the proposed model. Through the numerical examples, we have demonstrated that the extended cutting plane method performs better in terms of the objective function and the computation time. The examples also show that high holding cost and production cost result in less number of shipments in each cycle.

This research can be extended to consider shortage or multi-products and multi-constraints problems in an uncertain environment.

Acknowledgment

The authors would like to thank the anonymous referees for their comments on the earlier version of this work.

References

Chung, C. J., & Wee, H. M. (2007). Optimal replenishment policy for an integrated suppler-buyer deteriorating inventory model considering multiple JIT delivery and other cost functions. *Asia-Pacific Journal of Operational Research,* 24, 125 – 145.

Darwish, M. A. (2008). EPQ models with varying setup cost. *International Journal of Production Economics* 113, 297–306.

Freimer, M, Thomas, D. & Tyworth, J. (2006). The value of setup cost reduction and process improvement for the economic production quantity model with defects. *European Journal of Operational Research* 173, 241–251.

Geem Z.W., Kim J.H., Loganathan G.V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Simulation*, 76, 60–68.

Goyal, S. K. & Nebebe, F. (2000). Determination of economic production-shipment policy for a single-vendor-single-buyer system. *European Journal of Operational Research*, 121, 175-178.

Goyal, S. K. & Cardenas-Barron, L. E. (2002). Note on: Economic production quantity model for items with imperfect quality – a practical approach. *International Journal of Production Economics* 77, 85–87.

Haksever, C. & Moussourakis, J. (2008). Determining order quantities in multi-product inventory systems subject to multiple constraints and incremental discounts. *European Journal of Operational Research*, 184, 930-945.

Huang, Y. F. (2006). The effetc of service level constraint on EPQ model with random deffective rate. *Mathematical Problems in Engineering*, DOI 10.1155/MPE/2006/79028, 1–5.

Hu, F. & Liu, D. (2010). Optimal replenishment policy for the EPQ model with permissible delay in payments and allowable shortages. *Applied Mathematical Modeling*, 34(10), 3108-3117.

Islam, S. & Roy, T. K. (2007). Fuzzy multi-item economic production quantity model under space constraint: A geometric programming approach. *Applied Mathematics and Computation* 184, 326–335.

Kennedy J, Eberhart R (1995). Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, 1942-1945.

Kennedy J, Eberhart R (2001). *Swarm intelligence.* Academic Press, San Diego, CA.

Lai, C. S., Huang, Y. F. & Huang, H. F. (2006). The EPQ model with shortage and variable leadtime. *Journal of Applied Science* 6, 755-756.

Leung, K. M. F (2007). A generalized geometric-programming solution to an economic production quantity model with flexibility and reliability considerations. *European Journal of Operational Research*, 176, 240–251.

Liao, J. L. (2007). On an EPQ model for deteriorating items under permissible delay in payments. *Applied Mathematical Modeling*, 31, 393–403.

Ouyang, B. C., & Rau, H. (2008). An economic production lot size for continuous decrease in unit production cost. *Asia-Pacific Journal of Operational Research* 25, 673 – 688.

Pal, S., Maiti, M. K. & Maiti, M. (2009). An EPQ model with price discounted promotional demand in an imprecise planning horizon via Genetic Algorithm. *Computers & Industrial Engineering* 57, 181–187.

Salameh, M. & Jaber, M. Y. (2000). Economic production quantity model for items with imperfect quality. *International Journal of Production Economics*, 64, 59-64.

Taft, E. W. (1918). The most economical production lot. *The Iron Age*, 101, 1410–1412.

Taleizadeh, A. A., Najafi, A. A. & Niaki, S. T. A. (2010a). Economic production quantity model with scraped items and limited production capacity. *Scientia Iranica*, 17, 58-69.

Taleizadeh, A. A., Niaki, S. T. A., & Najafi, A. A. (2010b). Multi-product single-machine production system with stochastic scrapped production rate, partial backordering and service level constraint. *Journal of Computational and Applied Mathematics*, 233, 1834-1849.

Taleizadeh, A. A., Wee, H. M. & Sadjadi, S. J. (2010c). Multi-product production quantity model with repair failure and partial backordering. *Computers and Industrial Engineering*, 59, 45-54.

Taleizadeh, A.A., Niaki, S.T.A., Shafii, N., Ghavamizadeh-Meibodi, R., Jabbarzadeh A. (2010d). A particle swarm optimization approach for constraint joint single buyer-single vendor inventory problem with changeable lead time and (r,Q) policy in supply chain. International Journal of Advanced Manufacturing Technology, DOI 10.1007/s00170-010-2689-0.

Taleizadeh A.A., Moghadasi, H., Niaki, S.T.A., Eftekhari, A, (2008). An economic order quantity under joint replenishment policy to supply expensive imported raw materials with payment in advance. *Journal of Applied Sciences,* 8, 4263-4273.

Teng, J. T. & Chang, C. T. (2009). Optimal manufacturer's replenishment policies in the EPQ model under two levels of trade credit policy. *European Journal of Operational Research* 195, 358–363.

Teng, J. T., Ouyang, L. Y. & Chang, C. T. (2005). Deterministic economic production quantity models with time-varying demand and cost. *Applied Mathematical Modeling* 29, 987–1003.

Teng, J. T. & Chang, C. T. (2005). Economic production quantity models for deteriorating items with price- and stock-dependent demand. *Computers & Operations Research* 32, 297–308.

Su, C. H., Ouyang, L.Y., Ho C. H., & Chang C.T. (2007). Retailer's inventory policy and supplier's delivery policy under two-level trade credit strategy. *Asia-Pacific Journal of Operational Research* 24, 613 – 630.

Westerlund, T., and Pettersson, F., 1995. An extended cutting plane method for solving convex MINLP problems, *Computers chem. Engng.* 19, 131 – 136.