Applied Mathematics and Computation 230 (2014) 359–370

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A constrained multi-products EPQ inventory model with discrete delivery order and lot size

Leopoldo Eduardo Cárdenas-Barrón [a], Gerardo Treviño-Garza [b], Gede Agus Widyadana [c], Hui-Ming Wee [d,*]

[a] Department of Industrial and Systems Engineering, School of Engineering, Tecnológico de Monterrey, Ave. E.Garza Sada 2501 Sur, C.P. 64 849 Monterrey, Nuevo León, Mexico
[b] Department of Marketing and International Business, School of Business, Tecnológico de Monterrey, Ave. E.Garza Sada 2501 Sur, C.P. 64 849 Monterrey, Nuevo León, Mexico
[c] Department of Industrial Engineering, Petra Christian University, Surabaya, Indonesia
[d] Department of Industrial and Systems Engineering, Chung Yuan Christian University, Chung Li 320, Taiwan

ARTICLE INFO

Keywords:
EPQ
JIT
Lagrange approach
Branch and bound
Heuristic algorithms

ABSTRACT

This paper proposes an alternative heuristic algorithm for a multi-product Economic Production Quantity (EPQ) vendor–buyer integrated model with Just in Time (JIT) philosophy and a budget constraint. This type of problem is usually solved by a Mixed Integer Non Linear Problem (MINLP), which is complex and computationally expensive. The proposed heuristic algorithm involves less computation and therefore it is less expensive than the previously published algorithms. Furthermore, the heuristic algorithm is simpler as it derives the integer values for all discrete variables in a straightforward manner. Through empirical experimentation, it is demonstrated that the heuristic algorithm provides solutions closer to the lower bound in a very short time.

1. Introduction

Inventory management is one of the most important drivers for an effective enterprise. The first inventory model (i.e., EOQ) was proposed in February 1913 [1]. The two main decisions made via the EOQ model (i.e., how many products to order and when to place the order) are still widely studied. This is due to the fact that in any company there are different types of products and several constraints at the same time. For example, a company could manage orders for thousands of different products and have constraints such as space, budget, transportation capacity, number of orders, and production capacity.

In most instances, buyers or vendors optimize their ordering decisions independently. Several recent research studies have shown that the integrated vendor–buyer inventory model has better performance than the non-integrated inventory models.

This paper revisits the integrated single vendor single buyer inventory model with multi products and budget constraint proposed by Widyadana and Wee [2]. This problem is relevant for modern companies that manage thousands of products with different constraints.

Widyadana and Wee [2] extended the models of Pasandideh and Niaki [3] and Pasandideh et al. [4]. Cárdenas-Barrón et al. [5] improve and solve the inventory problem in Pasandideh et al. [4]. In inventory model proposed by Widyadana and Wee

* Corresponding author.
E-mail address: weehm@cycu.edu.tw (H.-M. Wee).

0096-3003/$ - see front matter
http://dx.doi.org/10.1016/j.amc.2013.12.077

[2], the vendor delivers products in small lots with discrete delivery orders. This inventory model is managed in a JIT environment where the buyer decides on the number products to order and deliver in a shipment.

The problem is a constrained mixed integer nonlinear problem (MINLP) and it has nonlinearities in the objective function and constraints. This type of MINLP is hard to solve using an exact method. Moreover, this MINLP is also challenging due to its high computational complexity. Therefore, Widyadana and Wee [2] developed a Lagrange method with a branch and bound procedure to solve the problem.

The MINLP has been used extensively to solve problems in engineering, finance, management science, and operations research. Many research studies have been conducted to solve MINLPs, particularly for handling highly nonlinear, combinatorial and large scale problems. Some researchers used a geometric algorithm to solve MINLPs (for instance see Islam and Roy [6,7]) and some used meta-heuristics methods such as genetic algorithm (for instance, see Pourakbar et al. [8] and Pasandideh et al. [9]). It was found that the solution procedure of Widyadana and Wee [2] was computationally expensive due to the fact that it was comprised of two parts: the Lagrange method as well as the branch and bound method. The Lagrange method contains seven steps. The branch and bound method contains five steps. Their algorithm evaluates the total cost function $n(n+1)+2$ times. Therefore, the complexity order of their solution procedure is $O(n^2)$. Also, it is worth mentioning that this MINLP is complex and computationally intensive even when a mathematical solver (such as LINGO) is used.

Widyadana and Wee [2] state that the computational time of their algorithm can be very high for large instances and that a more efficient and faster heuristic algorithm should be developed. In this direction, a simple heuristic algorithm to determine the integer value for each discrete variable is developed in this paper. Furthermore, the proposed algorithm discriminates among situations when there are one or two solutions for each discrete variable. It is important to note that the proposed algorithm derives the integer values for each discrete variable in a straightforward manner. Furthermore, the proposed heuristic algorithm evaluates the total cost immediately.

The rest of the paper is organized as follows. Section 2 presents two formulations of the MINLP problem for the integrated single vendor single buyer inventory model with multi-products and a budget constraint. Two heuristic algorithms are developed in order to solve each MINLP problem. Section 3 tests the effectiveness of the algorithm through numerical experimentation. Finally, Section 4 provides the conclusions and opportunities for future research.

2. Heuristic algorithms

For simplicity, we use the same assumptions and notation used in Widyadana and Wee [2]. The assumptions are (1) single vendor and single buyer are considered, (2) set-up and transportation times are insignificant and can be ignored, (3) demand rate is known and constant, (4) shortages are not allowed, (5) time horizon is infinite, (6) all costs are known and constant and (7) buyer pays transportation cost.

The notation is as follows:

l	number of products
m_i	shipment quantity of product i (decision variable)
K_i	number of shipments placed during period T of product i (decision variable)
P_i	vendor production rate of product i, units/year
D_i	the buyer's demand rate of product i, units/year
A_i	buyer ordering cost of product i
Av_i	vendor setup production cost of product i
b_i	transportation cost of product i
h_i	buyer holding cost per unit of product i
hv_i	vendor holding cost per unit of product i
c_i	product unit cost of product i
Bg	buyer budget

Based on the assumptions and notations described above, Widyadana and Wee [2] propose the following mixed integer nonlinear problem (MINLP) for the single vendor single buyer inventory model with multi products and budget constraint. Detail model development is shown in the Appendix.

$$Min\ Z = \sum_{i=1}^{l}\left\{\frac{D_i(A_i + Av_i)}{m_iK_i} + \frac{b_iD_i}{m_i} + \frac{m_iK_i}{2}\left[\frac{h_i + hv_i}{K_i} + hv_i\left(1 - \frac{D_i}{P_i}\right)\right]\right\} \tag{1}$$

s.t.

$$\sum_{i=1}^{l} c_i m_i K_i \leqslant Bg \tag{2}$$

$$m_i > 0; \quad i = 1, 2, 3, \ldots, l \tag{3}$$

$$K_i > 0; \quad \text{Integer}, i = 1, 2, 3, \ldots, l \tag{4}$$

In the formulation, objective function (1) is to minimize the integrated total inventory cost. Constraint (2) is the budget restriction. Constraint (3) states that all m_i's are continuous variables, and constraint (4) defines that all K_i's are discrete variables. We call this problem MINLP1. This problem contains $2l$ variables (l variables are discrete and the rest are continuous).

The problem where all m_i and all K_i are discrete variables is called MINLP2. In this case, constraint (3) must be $m_i \geqslant 1; \quad \text{Integer}, i = 1, 2, 3, \ldots, l$. In this problem, there are $2l$ discrete variables. It is worth mentioning that this problem was not considered in Widyadana and Wee [2].

The preceding optimization problems are hard to solve. We note that the difficulty in both problems lies in (1) the number of discrete variables, and (2) the nonlinearity of the objective function in the MINLP formulation.

In Sections 2.1 and 2.2, we discuss how to solve MINLP1 and MINLP2 in the general form. As it will be shown in Section 3, it is possible to get solutions closer to the lower bound within a very short time for large scale problems.

2.1. Solving the MINLP1 where all m_i are continuous variables and all K_i are discrete variables

The function Z (Eq. (1)) can be written as

$$Z = \sum_{i=1}^{l} \{X_i + Y_i\}$$

where

$$X_i = \frac{m_i(h_i + hv_i)}{2} + \frac{b_i D_i}{m_i}$$

$$Y_i = \frac{m_i K_i hv_i\left(1 - \frac{D_i}{P_i}\right)}{2} + \frac{D_i(A_i + Av_i)}{m_i K_i}$$

Note that both X_i and Y_i have the same mathematical form, which is $s_1 w + s_2/w$. Cárdenas-Barrón [10] showed via the algebraic method of complete squares that a function of type $s_1 w + s_2/w$ is always minimized for $w = \sqrt{s_2/s_1}$, which results in the minimum of $f(w) = 2\sqrt{s_1 s_2}$. Thus,

$$m_i = \sqrt{\frac{2b_i D_i}{h_i + hv_i}} \tag{5}$$

García-Laguna et al. [11] showed that the discrete solution to the following minimization problem:

$Min\ s_1 w + s_2/w; \quad$ where both s_1 and s_2 are positive

$w \geqslant 1 \quad$ and integer

is as follows:

$$w = \left\lceil -0.5 + \sqrt{0.25 + \frac{s_2}{s_1}} \right\rceil \quad \text{or} \quad w = \left\lfloor 0.5 + \sqrt{0.25 + \frac{s_2}{s_1}} \right\rfloor \tag{6}$$

where $\lceil r \rceil$ is the smallest integer greater than or equal to r, and $\lfloor r \rfloor$ is the largest integer less than or equal to r. In addition, it is clear that $\lceil r \rceil = \lfloor r + 1 \rfloor$ if and only if r is not an integer value. For this case, the problem has a unique solution for w which is $w^* = w$ (given by any of the two mathematical expressions in (6)). Otherwise, the problem has two solutions for w, i.e., $w^* = w$, and $w^* = w + 1$. This easy to apply procedure is similar to García-Laguna et al. [11], Cárdenas-Barrón et al. [12,13], and Teng et al. [14].

Based on Eq. (6), the solution for each discrete variable (K_i) is as follows:

$$K_i = \left\lceil -0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{m_i^2 hv_i\left(1 - \frac{D_i}{P_i}\right)}} \right\rceil \quad \text{or} \quad K_i = \left\lfloor 0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{m_i^2 hv_i\left(1 - \frac{D_i}{P_i}\right)}} \right\rfloor \tag{7}$$

Given the discrete value of each K_i, the continuous values for each m_i optimizing the following function can be determined

$$Min\ Z = \sum_{i=1}^{l} \left\{ \frac{D_i(A_i + Av_i)}{m_i K_i} + \frac{b_i D_i}{m_i} + \frac{m_i K_i}{2}\left[\frac{h_i + hv_i}{K_i} + hv_i\left(1 - \frac{D_i}{P_i}\right)\right] \right\}$$

Now, Z can be expressed as

$$Z = \sum_{i=1}^{l} X_i$$

where

$$X_i = m_i\left\{\frac{K_i h v_i(1-\frac{D_i}{P_i}) + h_i + h v_i}{2}\right\} + \frac{1}{m_i}\left\{\frac{D_i(A_i + A v_i + K_i b_i)}{K_i}\right\}$$

Applying the previous results from Cárdenas-Barrón [10], m_i are given by the following equation:

$$m_i = \sqrt{\frac{2D_i(A_i + A v_i + K_i b_i)}{K_i\left[K_i h v_i\left(1-\frac{D_i}{P_i}\right) + h_i + h v_i\right]}} \quad (8)$$

Based on the previous analysis, we describe the heuristic algorithm for solving the MINLP1 in detail as follows:

2.1.1. Heuristic algorithm for MINLP1

Step 0. Iterative procedure ← **false**
Step 1. Determine the initial continuous value for each m_i variable with Eq. (5) and the discrete value for each K_i variable with Eq. (7) ignoring the budget constraint. Given the discrete value of each K_i then calculate the final continuous value for each m_i variable with Eq. (8). If the solution satisfies the budget constraint, then go to **Step 6;** otherwise go to **Step 2**.
Step 2. Solve the optimization problem subject to the budget constraint. Determine the discrete value for each K_i variable with Eq. (9) where λ is calculated from Eq. (10).
Step 3. Given the discrete value of each K_i, then determine the continuous value for m_i from Eq. (11) where λ is obtained by solving Eq. (12).
Step 4. If **Iterative procedure** is false and if the solution satisfies the budget constraint, then go to **Step 6**; otherwise, **Iterative procedure** ← **true** and go to **Step 5**.
Step 5. If the solution does not satisfy the budget constraint then set budget constraint to $Bg_{new} \leftarrow \frac{Bg}{2}$.
Else
  if $|Bg_{new} - Bg| < \varepsilon$
    go to **Step 6.**
  Else
    Set $Bg_{new} = (Bg + Bg_{new})/2$
Set $Bg = Bg_{new}$ and go to **Step 1**.
Step 6. Determine the total cost with Eq. (1) and report the solution.

In Step 2, we optimize the following function adding a Lagrange multiplier λ:

$$Min\ Z = \sum_{i=1}^{l}\left\{\frac{D_i(A_i + A v_i)}{m_i K_i} + \frac{b_i D_i}{m_i} + \frac{m_i K_i}{2}\left[\frac{h_i + h v_i}{K_i} + h v_i\left(1-\frac{D_i}{P_i}\right)\right]\right\} + \lambda\left(\sum_{i=1}^{l} c_i m_i K_i - Bg\right)$$

Now, Z can be expressed as

$$Z = \sum_{i=1}^{l}\{X_i + Y_i\} + E$$

$$X_i = \frac{m_i(h_i + h v_i)}{2} + \frac{b_i D_i}{m_i}$$

Where $Y_i = \frac{m_i K_i\left[h v_i\left(1-\frac{D_i}{P_i}\right) + 2\lambda c_i\right]}{2} + \frac{D_i(A_i + A v_i)}{m_i K_i}$

$$E = -\lambda Bg$$

Applying the previous results, where m_i is also given by Eq. (5) and K_i can be determined by

$$K_i = \left\lceil -0.5 + \sqrt{0.25 + \frac{2D_i(A_i + A v_i)}{m_i^2\left[h v_i\left(1-\frac{D_i}{P_i}\right) + 2\lambda c_i\right]}}\right\rceil \quad \text{or} \quad K_i = \left\lfloor 0.5 + \sqrt{0.25 + \frac{2D_i(A_i + A v_i)}{m_i^2\left[h v_i\left(1-\frac{D_i}{P_i}\right) + 2\lambda c_i\right]}}\right\rfloor \quad (9)$$

where m_i is given by Eq. (5) and the λ value can be determined by solving the following equation:

$$\sum_{i=1}^{l} c_i \sqrt{\frac{2b_iD_i}{h_i + hv_i}} \left\{ \left\lceil -0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{\left(\frac{2b_iD_i}{h_i+hv_i}\right)\left[hv_i\left(1-\frac{D_i}{P_i}\right) + 2\lambda c_i\right]}} \right\rceil \right\} - Bg = 0 \tag{10}$$

In Step 3, given the discrete values of each K_i from Step 2, we optimize the following function using a Lagrange multiplier λ:

$$Min\ Z = \sum_{i=1}^{l} \left\{ \frac{D_i(A_i + Av_i)}{m_iK_i} + \frac{b_iD_i}{m_i} + \frac{m_iK_i}{2}\left[\frac{h_i + hv_i}{K_i} + hv_i\left(1 - \frac{D_i}{P_i}\right)\right] \right\} + \lambda\left(\sum_{i=1}^{l} c_im_iK_i - Bg\right)$$

Now, Z can be expressed as

$$Z = \sum_{i=1}^{l} X_i + E$$

where

$$X_i = m_i\left\{\frac{K_ihv_i\left(1-\frac{D_i}{P_i}\right) + h_i + hv_i + 2\lambda c_iK_i}{2}\right\} + \frac{1}{m_i}\left\{\frac{D_i(A_i + Av_i + K_ib_i)}{K_i}\right\}$$

$$E = -\lambda Bg$$

From Cárdenas-Barrón [10], m_i is given by the following equation:

$$m_i = \sqrt{\frac{2D_i(A_i + Av_i + K_ib_i)}{K_i\left[K_ihv_i\left(1 - \frac{D_i}{P_i}\right) + h_i + hv_i + 2\lambda c_iK_i\right]}} \tag{11}$$

where K_i is given by Eq. (9) and the λ value can be determined by solving the following equation:

$$\sum_{i=1}^{l} c_i \sqrt{\frac{2D_i(A_i + Av_i + K_ib_i)}{K_i[K_ihv_i(1 - \frac{D_i}{P_i}) + h_i + hv_i + 2\lambda c_iK_i]}} K_i - Bg = 0 \tag{12}$$

2.2. Solving the MINLP2 where m_i and K_i are discrete variables

MINLP2 can be solved as in MINLP1. Without loss of generality, we provide only the final results.

The discrete values for m_i are given by

$$m_i = \left\lceil -0.5 + \sqrt{0.25 + \frac{2b_iD_i}{h_i + hv_i}} \right\rceil \quad \text{or} \quad \left\lfloor 0.5 + \sqrt{0.25 + \frac{2b_iD_i}{h_i + hv_i}} \right\rfloor \tag{13}$$

The discrete values for K_i are given by

$$K_i = \left\lceil -0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{m_i^2hv_i\left(1 - \frac{D_i}{P_i}\right)}} \right\rceil \quad \text{or} \quad K_i = \left\lfloor 0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{m_i^2hv_i\left(1 - \frac{D_i}{P_i}\right)}} \right\rfloor \tag{14}$$

Given the discrete value of each K_i then the discrete value for each m_i can be determined by

$$m_i = \left\lceil -0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i + K_ib_i)}{K_i\left[K_ihv_i\left(1 - \frac{D_i}{P_i}\right) + h_i + hv_i\right]}} \right\rceil \quad \text{or} \quad \left\lfloor 0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i + K_ib_i)}{K_i\left[K_ihv_i\left(1 - \frac{D_i}{P_i}\right) + h_i + hv_i\right]}} \right\rfloor \tag{15}$$

When the budget constraint is active, then the discrete values for m_i are given by Eq. (13) and the discrete values for K_i are given by

$$K_i = \left\lceil -0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{m_i^2\left[hv_i\left(1 - \frac{D_i}{P_i}\right) + 2\lambda c_i\right]}} \right\rceil \quad \text{or} \quad K_i = \left\lfloor 0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{m_i^2\left[hv_i\left(1 - \frac{D_i}{P_i}\right) + 2\lambda c_i\right]}} \right\rfloor \tag{16}$$

where the λ value can be determined by solving the following equation:

$$\sum_{i=1}^{l} c_i \left[-0.5 + \sqrt{0.25 + \frac{2b_iD_i}{h_i + hv_i}} \right] \left\{ \left[-0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{\left[-0.5 + \sqrt{0.25 + \frac{2b_iD_i}{h_i+hv_i}} \right]^2 hv_i\left(1 - \frac{D_i}{P_i}\right) + 2\lambda c_i}} \right] \right\} - Bg = 0 \qquad (17)$$

Given the discrete value of each K_i, then the discrete value for each m_i can be determined by the following equation:

$$m_i = \left\lceil -0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i + K_ib_i)}{K_i\left[K_ihv_i\left(1 - \frac{D_i}{P_i}\right) + h_i + hv_i + 2\lambda c_iK_i\right]}} \right\rceil \quad \text{or} \quad \left\lfloor 0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i + K_ib_i)}{K_i\left[K_ihv_i\left(1 - \frac{D_i}{P_i}\right) + h_i + hv_i + 2\lambda c_iK_i\right]}} \right\rfloor \qquad (18)$$

where K_i is given by Eq. (16) and λ value can be determined by solving the following equation:

$$\sum_{i=1}^{l} c_im_i \left[-0.5 + \sqrt{0.25 + \frac{2D_i(A_i + Av_i)}{m_i^2\left[hv_i\left(1 - \frac{D_i}{P_i}\right) + 2\lambda c_i\right]}} \right] - Bg = 0 \qquad (19)$$

Based on the previous analysis, we describe the heuristic algorithm for solving MINLP2 in detail as follows:

2.2.1. Heuristic algorithm for MINLP2

Step 0. Iterative procedure ← false
Step 1. Determine the initial discrete value for each m_i with Eq. (13) and the discrete value for each K_i with Eq. (14) and ignoring the budget constraint. Given the discrete value of each K_i, then calculate the final discrete value for each m_i with Eq. (15). If the solution satisfies the budget constraint then go to **Step 6**. Otherwise go to **Step 2**.
Step 2. Solve the optimization problem subject to the budget constraint. Determine the discrete value for each K_i variable with Eq. (16) where λ is calculated with Eq. (17).
Step 3. Given the discrete value of each K_i then determine the discrete values for m_i with Eq. (18) where λ is obtained by solving Eq. (19).
Step 4. If **Iterative procedure** is false and the solution satisfies the budget constraint, then go to **Step 6**. Otherwise, **Iterative procedure ← true** and go to **Step 5**.
Step 5. If solution does not satisfy the budget constraint then
  set budget constraint to $Bg_{new} \leftarrow \frac{Bg}{2}$.
 Else
  if $|Bg_{new} - Bg| < \varepsilon$
   go to **Step 6.**
 Else
   Set $Bg_{new} = (Bg + Bg_{new})/2$
 Set $Bg = Bg_{new}$ and go to **Step 1**.
Step 6. Determine the total cost with Eq. (1) and report the solution.

The two heuristic algorithms can be implemented in a spreadsheet or any programming language such as C++, FORTRAN, among others. Both heuristic algorithms can be solved easily and only require evaluating the total cost function one time. In the next section, we present the results of the numerical experimentation.

3. Numerical experimentation

Both heuristics algorithms are illustrated from the data in Table 1 and Widyadana and Wee [2].
Example 1 considers the MINLP1 problem with a budget of \$30,000. The solution is given in Table 2.
Example 2 also considers MINLP1 problem but with a budget of \$20,000. The solution to this problem is given in Table 3. Using the solution procedure from Widyadana and Wee [2], we derive the same solution (see Table 4).
Example 3 also considers MINLP1 problem with a budget of \$20,000. The solution to this example is given in Table 5. Note that our solution is the same as Widyadana and Wee [2].

Table 1
Data for the numerical experimentation.

Product	A_i	D_i	h_i	c_i	Av_i	hv_i	P_i	b_i
1	47	1361	5	17	68	3	2444	14
2	49	1039	9	13	71	5	2355	15
3	58	1434	8	16	75	6	2392	13
4	49	1113	6	14	55	5	2440	18

Table 2
Solution to Example 1.

Product	m_i	K_i
1	69.1817	7
2	48.6220	6
3	50.5699	8
4	59.2575	5
Total cost (Z)	5830.7128	

Table 3
Solution to Example 2.

Product	Proposed m_i	Algorithm K_i
1	68.39359073	6
2	46.72992628	6
3	51.59875406	7
4	64.28319461	4
Total cost (Z)		5852.808723

Table 4
Data for the numerical experimentation from Widyadana and Wee [2].

Product	A_i	D_i	h_i	c_i	Av_i	hv_i	P_i	b_i
1	48	1255	8	17	74	3	2183	13
2	42	1342	5	13	57	3	2375	15
3	54	1395	9	16	52	6	2292	12
4	47	1169	6	14	78	4	2206	12

Table 5
Solution to Example 3.

Product	Proposed m_i	Algorithm K_i
1	52.5848	7
2	70.6239155	5
3	48.42575748	6
4	53.60692303	6
Total cost (Z)		5269.656386

Table 6
Data for the instances.

Parameter	Values
D_i	U(1000,15000)
P_i	$D + U$(10,100)
A_i	U(100,5000)
Av_i	U(100,5000)
b_i	U(20,80)
h_i	U(5,100)
hv_i	U(5,100)
c_i	U(10,200)

To make both heuristic algorithms more practical and relevant, we randomly generate 100 instances. We test both heuristic algorithms considering five levels of l (i.e., number of products): 50, 100, 250, 500, and 1000. At each level of l we generate 20 instances. The parameter values were generated from uniform distributions with ranges as shown in Table 6.

With regards to each instance, the budget values are determined as follows:

$$Bg = U(0.75, 1.1)c_i m_i K_i$$

Table 7
Results of the numerical experimentation for $l = 50$ products.

Instance	Total cost MINLP1	Lower bound	K_i discrete m_i continuous Difference	% percentage of penalty	CPU time in seconds	Total cost MINLP2	K_i discrete m_i discrete Difference	% percentage of penalty	CPU time in seconds
1	718930.112	718929.7974	0.3146	0.000044	0.0048	718933.4335	3.3215	0.000462	0.0107
2	647381.1844	647380.7459	0.4385	0.000068	0.0948	647386.2986	5.1142	0.00079	1.5209
3	667865.9868	667865.6893	0.2975	0.000045	0.0408	667872.8611	6.8743	0.001029	1.0183
4	694904.9807	694904.6311	0.3496	0.000050	0.0033	694908.9602	3.9795	0.000573	0.0035
5	643326.5934	643326.3194	0.274	0.000043	0.0498	643333.9443	7.3509	0.001143	1.1853
6	590461.4065	590461.2143	0.1922	0.000033	0.0034	590465.9221	4.5156	0.000765	0.0033
7	618129.6185	618129.118	0.5005	0.000081	0.0602	618136.8581	7.2396	0.001171	1.3636
8	685792.4025	685792.038	0.3645	0.000053	0.052	685796.7866	4.3841	0.000639	1.0554
9	696866.2593	696866.0645	0.1948	0.000028	0.0546	696870.204	3.9447	0.000566	0.0623
10	688356.5247	688356.2137	0.311	0.000045	0.0587	688360.8918	4.3671	0.000634	1.1676
11	665085.5275	665085.1979	0.3296	0.000050	0.0722	665091.0088	5.4813	0.000824	1.678
12	632040.6511	632040.2787	0.3724	0.000059	0.0509	632044.8463	4.1952	0.000664	0.0592
13	681535.0228	681534.7239	0.2989	0.000044	0.0584	681539.5239	4.5011	0.00066	1.2769
14	687949.7783	687949.2194	0.5589	0.000081	0.0623	687952.9886	3.2103	0.000467	0.0788
15	663776.7791	663776.313	0.4661	0.000070	0.0464	663781.4775	4.6984	0.000708	1.0845
16	683487.6103	683487.1194	0.4909	0.000072	0.051	683493.2646	5.6543	0.000827	1.1473
17	632935.7941	632935.405	0.3891	0.000061	0.0036	632939.5638	3.7697	0.000596	0.0034
18	644641.0084	644640.6528	0.3556	0.000055	0.03	644644.5604	3.552	0.000551	0.8983
19	627087.9843	627087.6743	0.31	0.000049	0.0034	627090.6814	2.6971	0.00043	0.0068
20	624905.9608	624905.536	0.4248	0.000068	0.0033	624910.2346	4.2738	0.000684	0.0054

Table 8
Results of the numerical experimentation for $l = 100$ products.

Instance	Total cost MINLP1	Lower bound	K_i discrete m_i continuous Difference	% percentage of penalty	CPU time in seconds	Total cost MINLP2	K_i discrete m_i discrete Difference	% percentage of penalty	CPU time in seconds
1	1297891.156	1297890.54	0.616	0.000047	0.0068	1297898.276	7.12	0.000549	0.0117
2	1282080.143	1282079.327	0.816	0.000064	0.0719	1282088.767	8.6239	0.000673	0.0891
3	1304284.227	1304283.268	0.9593	0.000074	0.0879	1304295.075	10.8475	0.000832	2.3573
4	1278617.641	1278616.56	1.0808	0.000085	0.0064	1278624.149	6.5087	0.000509	0.0073
5	1372179.116	1372178.329	0.7873	0.000057	0.0655	1372186.862	7.7464	0.000565	0.0804
6	1269249.374	1269248.624	0.7492	0.000059	0.083	1269257.575	8.2013	0.000646	0.1039
7	1346158.633	1346157.725	0.908	0.000067	0.0441	1346167.517	8.8838	0.00066	1.0953
8	1401339.2	1401338.035	1.165	0.000083	0.0743	1401350.043	10.8426	0.000774	1.9468
9	1335201.429	1335200.804	0.6247	0.000047	0.0749	1335208.827	7.3983	0.000554	2.2521
10	1339757.863	1339757.157	0.7058	0.000053	0.0819	1339765.632	7.7692	0.00058	0.107
11	1245298.846	1245298.214	0.6319	0.000051	0.05	1245305.514	6.6685	0.000535	1.3136
12	1308043.638	1308042.655	0.9836	0.000075	0.0656	1308051.401	7.763	0.000593	0.08
13	1343661.025	1343660.367	0.658	0.000049	0.0065	1343666.511	5.4859	0.000408	0.0072
14	1343226.17	1343225.534	0.6362	0.000047	0.0724	1343232.758	6.5882	0.00049	1.7446
15	1318052.239	1318051.496	0.7426	0.000056	0.0832	1318058.797	6.5575	0.000498	0.095
16	1323780.037	1323779.156	0.8817	0.000067	0.0713	1323786.306	6.2682	0.000474	0.0922
17	1327740.706	1327739.949	0.7571	0.000057	0.0761	1327750.362	9.656	0.000727	1.758
18	1309550.887	1309550.36	0.5275	0.000040	0.0101	1309556.549	5.6624	0.000432	0.0077
19	1415022.169	1415021.627	0.5419	0.000038	0.0065	1415029.615	7.4457	0.000526	0.0072
20	1286072.727	1286071.637	1.0901	0.000085	0.0065	1286079.197	6.4707	0.000503	0.007

The actual data sets may be obtained from any of the first two authors upon request. We test the computation time performance of both heuristic algorithms on a laptop with the following technical specifications: Intel® Core™ 2 Duo CPU, P8700 @ 2.53 GHz, 3.45 GB of RAM.

The results of the numerical experimentation for both heuristic algorithms are shown in Tables 7–11.

For the MINLP1 the percentage of penalty is determined as follows:

$$Percentage\ of\ penalty = \left[\frac{(TotalCost(Z)\ of\ MINLP1 - LowerBound)}{Lower\ Bound}\right] * 100\%$$

The Lower Bound is determined by solving the relaxed problem considering all variables as continuous and satisfying the budget constraint.

In the MINLP2, the percentage of penalty is calculated as

Table 9
Results of the numerical experimentation for $l = 250$ products.

Instance	Total cost MINLP1	Lower bound	K_i discrete m_i continuous Difference	% percentage of penalty	CPU time in seconds	Total cost MINLP2	K_i discrete m_i discrete Difference	% percentage of penalty	CPU time in seconds
1	3310704.336	3310702.707	1.6292	0.000049	0.0969	3310721.665	17.3287	0.000523	0.1191
2	3302428.175	3302426.577	1.5974	0.000048	0.016	3302444.974	16.7992	0.000509	0.0171
3	3230262.109	3230260.357	1.752	0.000054	0.1543	3230281.685	19.5755	0.000606	3.6394
4	3207502.691	3207501.561	1.1299	0.000035	0.0159	3207518.733	16.0419	0.0005	0.018
5	3312765.559	3312763.876	1.6831	0.000051	0.0198	3312784.388	18.8297	0.000568	0.017
6	3258972.3	3258970.75	1.55	0.000048	0.1028	3258990.695	18.395	0.000564	0.1284
7	3292525.675	3292523.875	1.8004	0.000055	0.1064	3292542.998	17.3228	0.000526	0.1385
8	3235762.82	3235760.855	1.965	0.000061	0.0198	3235780.997	18.1772	0.000562	0.0162
9	3296467.073	3296464.923	2.1494	0.000065	0.1606	3296485.404	18.3312	0.000556	0.1654
10	3185027.138	3185025.322	1.8157	0.000057	0.0166	3185044.032	16.8935	0.00053	0.0161
11	3326129.604	3326127.868	1.736	0.000052	0.1065	3326147.251	17.6469	0.000531	0.1339
12	3249829.207	3249827.408	1.7992	0.000055	0.0188	3249849.381	20.1743	0.000621	0.0175
13	3278570.651	3278568.655	1.9958	0.000061	0.1623	3278588.532	17.8814	0.000545	0.1806
14	3280804.876	3280803.013	1.8627	0.000057	0.1031	3280825.381	20.5051	0.000625	0.131
15	3340613.94	3340612.297	1.6432	0.000049	0.0967	3340630.216	16.2757	0.000487	0.1177
16	3215728.995	3215727.079	1.9163	0.000060	0.0992	3215744.611	15.6158	0.000486	0.1173
17	3271687.61	3271685.911	1.6985	0.000052	0.1442	3271701.841	14.2309	0.000435	3.4284
18	3176454.819	3176452.871	1.9485	0.000061	0.0156	3176469.767	14.9475	0.000471	0.0163
19	3287899.559	3287897.565	1.9944	0.000061	0.0187	3287917.691	18.1322	0.000551	0.0186
20	3412322.681	3412320.783	1.8982	0.000056	0.0155	3412343.29	20.6092	0.000604	0.0172

Table 10
Results of the numerical experimentation for $l = 500$ products.

Instance	Total cost MINLP1	Lower bound	K_i discrete m_i continuous Difference	% percentage of penalty	CPU time in seconds	Total cost MINLP2	K_i discrete m_i discrete Difference	% percentage of penalty	CPU time in seconds
1	6729944.296	6729940.703	3.593	0.000053	0.0367	6729981.075	36.7792	0.000547	0.0319
2	6494360.52	6494356.846	3.6744	0.000057	0.2714	6494394.937	34.4163	0.00053	0.3263
3	6640896.035	6640892.142	3.89330	0.000059	0.0324	6640930.169	34.1336	0.000514	0.0425
4	6568061.842	6568058.273	3.5696	0.000054	0.2407	6568097.341	35.4983	0.00054	6.2726
5	6751048.136	6751044.05	4.0858	0.000061	0.269	6751084.796	36.6603	0.000543	0.3
6	6583611.29	6583607.267	4.02279	0.000061	0.0307	6583647.156	35.8658	0.000545	0.0326
7	6506531.923	6506526.988	4.9345	0.000076	0.18	6506568.346	36.4231	0.00056	4.7937
8	6409793.668	6409790.333	3.3355	0.000052	0.0309	6409827.175	33.5063	0.000523	0.0319
9	6823969.11	6823964.759	4.3511	0.000064	0.0336	6824005.694	36.5839	0.000536	0.0333
10	6463125.573	6463121.245	4.3279	0.000067	0.2559	6463161.515	35.9421	0.000556	0.3003
11	6516981.845	6516978.793	3.0513	0.000047	0.2566	6517017.258	35.4129	0.000543	0.2855
12	6660463.008	6660459.126	3.8817	0.000058	0.2512	6660502.325	39.3169	0.00059	5.7179
13	6578479.257	6578474.59	4.6666	0.000071	0.2654	6578511.902	32.6453	0.000496	0.3042
14	6593560.004	6593555.916	4.08740	0.000062	0.0402	6593595.909	35.9053	0.000545	0.0305
15	6687670.091	6687666.693	3.3974	0.000051	0.0417	6687705.675	35.5844	0.000532	0.0336
16	6600664.882	6600660.743	4.1397	0.000063	0.2788	6600700.822	35.9396	0.000544	6.2237
17	6685410.901	6685407.058	3.8433	0.000057	0.2795	6685448.882	37.9803	0.000568	6.6248
18	6408940.924	6408937.533	3.3914	0.000053	0.2086	6408976.761	35.8361	0.000559	5.4053
19	6610198.049	6610193.764	4.285	0.000065	0.0335	6610230.79	32.7411	0.000495	0.0324
20	6440814.677	6440810.762	3.9151	0.000061	0.3425	6440848.939	34.2626	0.000532	0.3879

$$Percentage\ of\ penalty = \left[\frac{(Total\ Cost\ (Z)\ of\ MINLP2 - Total\ Cost\ (Z)\ of\ MINLP1)}{Total\ Cost\ (Z)\ of\ MINLP1}\right] * 100\%.$$

The results of the empirical experimentation show that the proposed heuristic algorithms perform very well. From the results in Tables 12 and 14, the minimum, the maximum and the average percentage penalties are very small; the heuristic algorithms obtain a near optimal solution.

The results in Table 13 show that the CPU times are very short (less than 0.6 s) in the heuristic algorithm for MINLP1. On the other hand, in Table 15 the CPU times for the heuristic algorithm for MINLP2 are also small (less than 15 s for very large problems).

Finally, both heuristic algorithms produce good results in the following three performance measures: the percentage of penalty, the number of evaluations of the total cost, and the computational time.

Table 11
Results of the numerical experimentation for l = 1000 products.

Instance	Total cost MINLP1	Lower bound	K_i discrete m_i continuous Difference	% percentage of penalty	CPU time in seconds	Total cost MINLP2	K_i discrete m_i discrete Difference	% percentage of penalty	CPU time in seconds
1	13449860.3	13449853.06	7.2378	0.000054	0.4735	13449932.74	72.4404	0.000539	13.3319
2	13158126.47	13158118.38	8.0854	0.000061	0.5428	13158195.18	68.7164	0.000522	0.6023
3	13407801.47	13407792.79	8.67710	0.000065	0.5131	13407864.87	63.3986	0.000473	14.3552
4	13089769.31	13089760.52	8.7917	0.000067	0.4969	13089841.12	71.8062	0.000549	11.9
5	13384993.07	13384985.14	7.92940	0.000059	0.3674	13385061.19	68.1258	0.000509	0.4507
6	13012535.32	13012527.68	7.6424	0.000059	0.0628	13012604.88	69.5612	0.000535	0.0633
7	13286454.62	13286446.5	8.1208	0.000061	0.5118	13286527.96	73.3422	0.000552	14.4067
8	13412643.31	13412635.53	7.78429	0.000058	0.0754	13412712.78	69.4674	0.000518	0.0674
9	13259354.02	13259345.47	8.55629	0.000065	0.5105	13259426.92	72.8946	0.00055	12.9997
10	13300364.21	13300357.48	6.7310	0.000051	0.0652	13300435.76	71.5504	0.000538	0.0663
11	13495885.64	13495877.35	8.2926	0.000061	0.3707	13495955.81	70.1707	0.00052	0.4515
12	13442394.13	13442386.52	7.61260	0.000057	0.0735	13442463.62	69.4951	0.000517	0.0681
13	13205903.67	13205895.8	7.8621	0.000060	0.0636	13205974.32	70.6481	0.000535	0.066
14	13168241.11	13168233.35	7.76269	0.000059	0.0765	13168308.61	67.5004	0.000513	0.0667
15	13275288.8	13275281.35	7.4451	0.000056	0.4423	13275360.58	71.7883	0.000541	0.5379
16	13352337.23	13352330.23	7.00359	0.000052	0.3753	13352402.56	65.3282	0.000489	9.8421
17	13350739.88	13350732.44	7.4405	0.000056	0.4038	13350814.21	74.3271	0.000557	0.4229
18	13340260.8	13340253.14	7.66049	0.000057	0.5158	13340335.61	74.8126	0.000561	13.1459
19	13485537.87	13485529.82	8.057	0.000060	0.5125	13485611.41	73.5385	0.000545	13.0633
20	13406542.76	13406534.4	8.36219	0.000062	0.5012	13406614.27	71.505	0.000533	0.5975

Table 12
Minimum, average, and maximum percentage of penalty for MINLP1.

l number of products	Minimum	Average	Maximum
50	0.000028	0.000055	0.000081
100	0.000038	0.000060	0.000085
250	0.000035	0.000054	0.000065
500	0.000047	0.000060	0.000076
1000	0.000051	0.000059	0.000067

Table 13
Minimum, average, and maximum CPU time in seconds for MINLP1.

l number of products	Minimum	Average	Maximum
50	0.0033	0.040195	0.0948
100	0.0064	0.052245	0.0879
250	0.0155	0.074485	0.1623
500	0.0307	0.168965	0.3425
1000	0.0628	0.34773	0.5428

Table 14
Minimum, average, and maximum percentage of penalty for MINLP2.

l number of products	Minimum	Average	Maximum
50	0.00043	0.000709	0.001171
100	0.000408	0.000576	0.000832
250	0.000435	0.00054	0.000625
500	0.000495	0.00054	0.00059
1000	0.000473	0.00053	0.000561

4. Conclusions

The main contribution of this paper is to present two heuristic algorithms and determine the integer values of the discrete variables for a multi-products EPQ vendor–buyer integrated model with JIT philosophy and a budget constraint. The proposed heuristic algorithms are simple and practically relevant and require calculating the total cost just once. We compare

Table 15
Minimum, average, and maximum CPU time in seconds for MINLP2.

l number of products	Minimum	Average	Maximum
50	0.0033	0.681475	1.678
100	0.007	0.65817	2.3573
250	0.0161	0.422685	3.6394
500	0.0305	1.860545	6.6248
1000	0.0633	5.32527	14.4067

our algorithm with Widyadana and Wee [2] and derive the same result. However, our algorithms run faster than Widyadana and Wee [2] who used the branch and bound algorithm. Moreover, the running time for the algorithm developed by Widyadana and Wee [2] increases exponentially as the problem size increases, while the running time for our algorithm increases linearly. Future research can be done to simultaneously consider constraints on space, budget, and the number of orders. Also, an *n*-stage-multi-customer integrated inventory system with planned backorders for all members with both fixed and linear backordering costs should be of interest to develop.

Acknowledgments

This research was supported by the School of Business of the Tecnológico de Monterrey and the Tecnológico de Monterrey research fund numbers CAT128 and CAT185. The authors would also like to thank Dr. J. David Porter from Oregon State University and the anonymous referees for their constructive comments and suggestions that enhanced this paper. The first author would like to thank Sarai Rodríguez Payán, Italia Tatnaí Cárdenas Rodríguez, David Nahúm Cárdenas Rodríguez, and Zuriel Eluzai Cárdenas Rodríguez for their valuable support during the development of this paper.

Appendix A

In the vendor–buyer inventory problem with discrete delivery order and lot size, the vendor delivers in a discrete frequency (K times) in one replenishment period. Buyer inventory level of product i can be represented in Fig. A1. Vendor delivers m units when buyer's inventory level reach zero. There is no lead time between vendor and buyer.

The buyer's total inventory cost consists of setup cost, transportation cost, and holding cost. The total inventory cost per unit of product i can be shown as:

$$TBUC_i = \frac{A_iD_i}{Q_i} + \frac{b_iK_iD_i}{Q_i} + \frac{h_im_i}{2} \tag{A1}$$

Substitute Q_i with m_iK_i to (A1), one has:

$$TBUC_i = \frac{A_iD_i}{K_im_i} + \frac{b_iD_i}{m_i} + \frac{h_im_i}{2} \tag{A2}$$

Fig. A2 shows the vendor inventory model. The vendor produces product i until w_iT_i/K_i period and delivers m unit of product i every T_i/K_i period in K_i delivery times. The vendor's average inventory can be represented as:

$$IP_i = \frac{m_i(K_i - w_i + 1)}{2} \tag{A3}$$

The vendor total cost consists of setup cost and inventory holding cost. It can be modeled as:

$$TVUC_i = \frac{Av_iD_i}{Q_i} + \frac{hv_im_i(K_i - w_i + 1)}{2} \tag{A4}$$

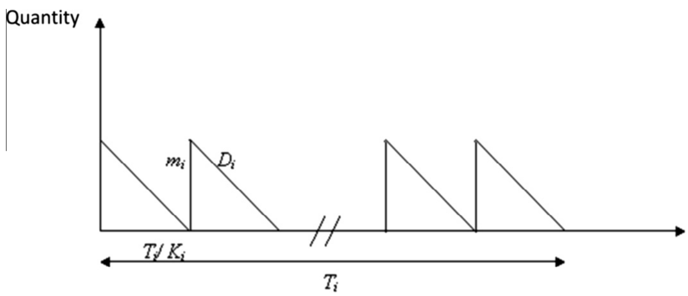

Figure A1. Buyer inventory level.

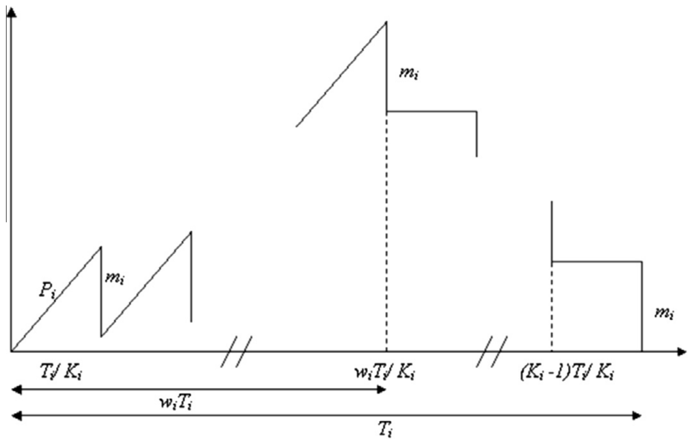

Figure A2. Vendor inventory model.

Substitute Q_i with m_iK_i to (A4), one has:

$$TVUC_i = Av_i\frac{D_i}{m_iK_i} + \frac{hv_im_i(K_i - w_i + 1)}{2} \tag{A5}$$

Total vendor–buyer inventory cost is equal to total buyer cost and total vendor cost.

$$TUC = \sum_{i=1}^{l} \frac{A_iD_i}{K_im_i} + \frac{b_iD_i}{m_i} + \frac{h_im_i}{2} + Av_i\frac{D_i}{m_iK_i} + \frac{hv_im_i(K_i - w_i + 1)}{2} \tag{A6}$$

When the buyer setup the quantity order, she or he has to spend cQ dollar to pay for the order. Since Q order quantity can be represented as mK and the buyer has a budget of Bg, one has:

$$\sum_{i=1}^{v} c_im_iK_i \leqslant Bg \tag{A7}$$

References

[1] F.W. Harris, How many parts to make at once factory, Mag. Manage. 10 (1913) 135–136 and 152.
[2] G.A. Widyadana, H.M. Wee, A multi-products EPQ model with discrete delivery order: a Lagrangean solution approach, in: S.Y. Chou, A. Trappey, J. Pokojski, S. Smith (Eds.), Global Perspective for Competitive Enterprise, Economy and Ecology: Proceedings of the 16th ISPE International Conference on Concurrent Engineering, Springer-Verlag, London, 2009, pp. 601–608.
[3] S.H.R. Pasandideh, S.T.A. Niaki, A genetic algorithm approach to optimize a multi-products EPQ model with discrete delivery orders and constrained space, Appl. Math. Comput. 195 (2) (2008) 506–514.
[4] S.H.R. Pasandideh, S.T.A. Niaki, J.A. Yeganeh, A parameter-tuned genetic algorithm for multi-product economic production quantity model with space constraint, discrete delivery orders and shortages, Adv. Eng. Softw. 41 (2) (2010) 306–314.
[5] L.E. Cárdenas-Barrón, G. Treviño-Garza, H.M. Wee, A simple and better algorithm to solve the vendor managed inventory control system of multi-product multi-constraint economic order quantity model, Expert Syst. Appl. 39 (3) (2012) 3888–3895.
[6] S. Islam, T.K. Roy, A fuzzy EPQ model with flexibility and reliability consideration and demand dependent unit production cost under a space constraint: a fuzzy geometric programming approach, Appl. Math. Comput. 176 (2) (2006) 531–544.
[7] S. Islam, T.K. Roy, Fuzzy multi-item economic production quantity model under space constraint: a geometric programming approach, Appl. Math. Comput. 184 (2) (2007) 326–335.
[8] M. Pourakbar, R.Z. Farahani, N. Asgari, A joint economic lot-size model for an integrated supply network using genetic algorithm, Appl. Math. Comput. 189 (1) (2007) 583–596.
[9] S.H.R. Pasandideh, S.T.A. Niaki, A.R. Nia, A genetic algorithm for vendor managed inventory control system of multi-product multi-constraint economic order quantity model, Expert Syst. Appl. 38 (3) (2011) 2708–2716.
[10] L.E. Cárdenas-Barrón, The derivation of EOQ/EPQ inventory models with two backorders costs using analytic geometry and algebra, Appl. Math. Model. 35 (5) (2011) 2394–2407.
[11] J. García-Laguna, L.A. San-José, L.E. Cárdenas-Barrón, J. Sicilia, The integrality of the lot size in the basic EOQ and EPQ models: applications to other production-inventory models, Appl. Math. Comput. 216 (5) (2010) 1660–1672.
[12] L.E. Cárdenas-Barrón, H.M. Wee, M.F. Blos, Solving the vendor-buyer integrated inventory system with arithmetic-geometric inequality, Math. Comput. Model. 53 (5–6) (2011) 991–997.
[13] L.E. Cárdenas-Barrón, H.M. Wee, J.-T. Teng, A supplement to "Using the EPQ for coordinated planning of a product with partial backordering and its components", Math. Comput. Model. 54 (1–2) (2011) 852–857.
[14] J.-T. Teng, L.E. Cárdenas-Barrón, K.-R. Lou, The economic lot size of the integrated vendor–buyer inventory system derived without derivatives: a simple derivation, Appl. Math. Comput. 217 (12) (2011) 5972–5977.