Expert Systems with Applications 39 (2012) 3888–3895

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Short Communication

A simple and better algorithm to solve the vendor managed inventory control system of multi-product multi-constraint economic order quantity model

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

[a] Department of Industrial and Systems Engineering, School of Engineering, Instituto Tecnológico y de Estudios Superiores de Monterrey, E. Garza Sada 2501 Sur, CP 64 849, Monterrey, NL, Mexico

[b] Department of Management, School of Business, Instituto Tecnológico y de Estudios Superiores de Monterrey, E. Garza Sada 2501 Sur, CP 64 849, Monterrey, NL, Mexico

[c] Department of Industrial and Systems Engineering, Chung Yuan Christian University, Chungli 32023, Taiwan, ROC

ARTICLE INFO

Keywords:
Genetic algorithms
Heuristic algorithms
EOQ
Vendor management inventory
Nonlinear integer programming

ABSTRACT

This research presents an alternative heuristic algorithm to solve the vendor management inventory system with multi-product and multi-constraint based on EOQ with backorders considering two classical backorders costs: linear and fixed. For this type of inventory system, the optimization problem is a nonlinear integer programming (NLIP). Several numerical examples are given to demonstrate that the proposed heuristic algorithm is better than the previous genetic algorithm published based on three aspects: the total cost, the number of evaluations of the total cost function and computational time. Furthermore, the proposed algorithm is simpler and can be implemented by any people.

1. Introduction

Almost a century ago, the economic order quantity (EOQ) inventory model without backorders was proposed by Harris (1913). Five years later, the economic production quantity (EPQ) inventory model without backorders was proposed by Taft (1918). Later, Hadley and Whitin (1963) proposed the EOQ/EPQ inventory models with backorders. A complete review of different optimization methods used in inventory field can be seen in Cárdenas-Barrón (2011).

Recently, Pasandideh, Niaki, and Nia (2011) presented a genetic algorithm for vendor management inventory system with multi-product, multi-constraint based on EOQ with backorders considering two classical backorders costs: linear and fixed. In the vendor management inventory system both retailer and supplier manage their inventories in a win to win manner. The vendor management inventory has attracted the attention of the researchers and practitioners, e.g. Kwak, Choi, Kim, and Kwon (2009), Lin, Chang, Hung, and Pai (2010), Arora, Chan, and Tiwari (2010), just to name a few recently works.

We have read the paper with considerable interest. Pasandideh et al. (2011) solved a hard optimization problem thorough a valuable and elegant approach based on genetic algorithms. Genetic algorithms have received an increasing attention from the researchers and practitioners since they give us an alternative to traditional optimization techniques. This approach uses a directed random search to locate very good solutions for complex optimization problems in many different fields of study. The use of genetic algorithms to solve inventory problems has been a common research topic recently. However, in some cases, the genetic algorithms are computationally expensive as Cárdenas-Barrón (2010) stated.

We found some interesting points to discuss: (1) the mathematical formulation of the problem has some shortcomings, (2) some of solutions to the test problems are infeasible and (3) there exist better solutions for the test problems.

2. Discussion

2.1. The mathematical formulation of the problem has some shortcomings

The Pasandideh et al.'s (2011) formulation is:

$$Min \sum_{j=1}^{n} \left[\frac{D_j}{Q_j}(A_{jS} + A_{jR}) + \frac{h_{jR}}{2Q_j}(Q_j - b_j)^2 + \frac{\hat{\pi} b_j^2}{2Q_j} + \frac{\pi b_j D_j}{Q_j} \right]$$

$$\sum_{j=1}^{n} f_j Q_j \leqslant F$$

$$\sum_{j=1}^{n} \frac{D_j}{Q_j} \leqslant M$$

$$Q_j, b_j \geqslant 0: \ Integer, \ j = 1, 2, 3, \ldots, n$$

* Corresponding author. Tel.: +52 8183284235; fax: +52 8183284153.
E-mail address: lecarden@itesm.mx (L.E. Cárdenas-Barrón).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.09.057

which is a nonlinear integer programming (NLIP) model, and this problem is hard to solve by a exact method. They developed a genetic algorithm to solve the NLIP problem. However, the NLIP model has some shortcomings, for example the first constraint $\sum_{j=1}^{n} f_j Q_j \leqslant F$ should be $\sum_{j=1}^{n} f_j (Q_j - b_j) \leqslant F$. The last constraints; should be $Q_j > 0$, $b_j \geqslant 0$; *integer*, $j = 1, 2, 3, \ldots, n$. Technically, the lot size cannot be zero. Additionally, there is missing the following constraint $b_j \leqslant Q_j$. The backorders size cannot be greater than the lot size Q_j.

Therefore, the correct NLIP model should be:

$$MinZ = \sum_{j=1}^{n} \left[\frac{D_j}{Q_j}(A_{jS} + A_{jR}) + \frac{h_{jR}}{2Q_j}(Q_j - b_j)^2 + \frac{\hat{\pi}_j b_j^2}{2Q_j} + \frac{\pi_j b_j D_j}{Q_j} \right] \quad (1)$$

$$\sum_{j=1}^{n} (Q_i - b_j) \leqslant F \quad (2)$$

$$\sum_{j=1}^{n} \frac{D_j}{Q_j} \leqslant M \quad (3)$$

$$b_j \leqslant Q_j \quad (4)$$

$$Q_j > 0; \quad Integer, \ j = 1, 2, 3, \ldots, n \quad (5)$$

$$b_j \geqslant 0; \quad Integer, \ j = 1, 2, 3, \ldots, n \quad (6)$$

where the nomenclature is:

- D_j = Retailer's demand rate of the *j*th product (units/time unit).
- A_{jS} = Supplier's ordering cost of the *j*th product ($/order).
- A_{jR} = Retailer's ordering cost of the *j*th product ($/order).
- h_{jR} = holding cost of the *j*th product ($/unit/time unit).
- $\hat{\pi}_j$ = linear backorder cost of the *j*th product ($/unit/time unit).
- π_j = fixed backorder cost of the *j*th product ($/unit).
- F = Total available storage space for all products.
- f_j = Space occupied by one unit of the *j*th product.
- M = Total number of orders for all products.
- n = Total of products.
- Q_j = Order quantity of the *j*th product (units); a decision variable.
- b_j = Maximum backorders level of the *j*th product (units); a decision variable.
- $j = 1, 2, 3, \ldots, n$.

Note that our proposed NLIP model permits that both backorders costs, linear and fixed, to be different for all products.

In the formulation, the objective (1) is to minimize the integrated total inventory cost. Constraint (2) is the space restriction. Constraint (3) states that total number or orders for all products is constrained to be less or equal to M; and constraint (4) establishes that all b_i's are less or equal to Q_j. Constraints (5) and (6) defines that all b_i's and Q_j's are discrete variables. This problem contains $2n$ discrete variables. The previous optimization problem is hard to solve. We notice that the difficult of the problem lies in the quantity of discrete variables and the nonlinearity of objective function (1) and constraint (3) in the NILP formulation.

2.2. Some of solutions to the test problems are infeasible

Pasandideh et al. (2011) solve only four test problems in order to test the efficiency of their genetic algorithm. In the solutions reported for the test problems, we found that there are some solutions that are infeasible. The data of the test problems is reported in Table 1.

From the solutions reported in Tables 2 and 3, it is important to remark that Pasandideh et al. (2011) showed 12 alternatives solutions for each test problem. For the first test problem ($n = 3$), there are six feasible solutions and 6 infeasible solutions. The best feasible solution for the first test problem has a total cost of 925.731. For the second test problem ($n = 5$), there are eight feasible solutions and four infeasible solutions. The best feasible solution for second test problem has a total cost of 1,191.057. For the third test problem ($n = 8$), there are four feasible solutions and 8 infeasible solutions. The best feasible solution for third test problem has a total cost of 4,346.243. Finally, for the fourth test problem ($n = 10$), there is only 1 feasible solution and 11 infeasible solutions. The unique feasible solution for the fourth test problem has a total cost of 6,654.628.

Table 1
Data for the test problems.

Product j	D_j	A_{jS}	A_{jR}	h_{jR}	f_j
1	420	1	3	4	3
2	360	2	2	9	2
3	540	3	1	7	3
4	390	5	4	2	1
5	480	2	2	4	4
6	510	4	2	6	3
7	530	1	3	5	2
8	380	2	1	3	1
9	430	3	4	2	3
10	580	4	2	8	4
		$F = 18000$	$M = 12$	$\hat{\pi} = 3$	$\pi = 0$

2.3. There are better solutions to test problems

Recently, there is a lot of interest in developing heuristics algorithms that can find near-optimal solutions within a reasonable computation time for complex problems. The optimization problem (1)–(6) is a hard problem which in fact has two main constraints: space (2) and orders (3). According to Hadley and Whitin (1963) the continuous optimization problem with two or more constraints is more complicated.

When there are two constraints imposed simultaneously, the main procedure to solve the problem is as follows:

Main procedure

Step 1: Determine the optimal continuous values ignoring both constraints. If Q_j and b_j satisfy both constraints, a continuous solution has been found. Then calculate the integer values for Q_j and b_j with the ***heuristic algorithm*** and go to step 5.
Step 2: Else solve the optimization problem subject to constraint orders and ignore constraint space. If Q_j and b_j satisfy the space constraint, a continuous solution has been found. Then calculate the integer values for Q_j and b_j with the ***heuristic algorithm*** and go to step 5.
Step 3: Else solve the optimization problem subject to constraint space and ignore constraint orders. If Q_j and b_j satisfy the orders constraint, a continuous solution has been found. Then calculate the integer values for Q_j and b_j with the ***heuristic algorithm*** and go to step 5.
Step 4: If none of the three steps aforementioned is applicable, both constraints are active. Then solve the optimization problem subject to both constraints orders and space. Determine Q_j and b_j and the continuous solution has been found. Then calculate the integer values for Q_j and b_j with the ***heuristic algorithm.***
Step 5: Solution found.

In Step 1, we optimize the objective function (1) ignoring constrains (2)–(6). In other words, considering the variables as continuous values we solve the relaxed problem. Therefore, we minimize over each Q_j and b_j separately. The optimal Q_j and b_j can be