Int J Adv Manuf Technol (2010) 51:1209–1223
DOI 10.1007/s00170-010-2689-0

ORIGINAL ARTICLE

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

Ata Allah Taleizadeh · Seyed Taghi Akhavan Niaki · Nima Shafii · Ramak Ghavamizadeh Meibodi · Armin Jabbarzadeh

Received: 13 December 2008 / Accepted: 20 April 2010 / Published online: 15 May 2010

Abstract In this paper, the chance-constraint joint single vendor-single buyer inventory problem is considered in which the demand is stochastic and the lead time is assumed to vary linearly with respect to the lot size. The shortage in combination of back order and lost sale is considered and the demand follows a uniform distribution. The order should be placed in multiple of packets, the service rate limitation on each product is considered a chance constraint, and there is a limited budget for the buyer to purchase the products. The goal is to determine the re-order point and the order quantity of each product such that the chain total cost is minimized. The model of this problem is shown to be an integer nonlinear programming type and in order to solve it, a particle swarm optimization (PSO) approach is used. To assess the efficiency of the proposed algorithm, the model is solved using both genetic algorithm and simulated annealing approaches as well. The results of the comparisons by a numerical example, in which a sensitivity analysis on the model parameters is also performed, show that the proposed PSO algorithm performs better than the other two methods in terms of the total supply chain costs.

Keywords Multi-product inventory · Stochastic demand · Changeable lead time · Supply chain · Integer nonlinear programming · Particle swarm optimization

A. A. Taleizadeh · A. Jabbarzadeh
Department of Industrial Management,
Raja University,
Qazvin, Iran

A. A. Taleizadeh
e-mail: taleizadeh@iust.ac.ir

A. Jabbarzadeh
e-mail: arminj@raja.ac.ir

S. T. A. Niaki (✉)
Department of Industrial Engineering,
Sharif University of Technology,
P.O. Box 11155-9414, Azadi Ave.,
Tehran 1458889694, Iran
e-mail: Niaki@Sharif.edu

N. Shafii
Faculty of Engineering, University of Porto,
Rua Dr. Roberto Frais,
Porto, Portugal
e-mail: Nima.Shafii@Gmail.com

R. G. Meibodi
Department of Electrical and Computer Engineering,
Shahid Beheshti University,
Tehran, Iran
e-mail: r-ghavami@sbu.ac.ir

1 Introduction and literature review

As the firms realize that a more efficient management of inventories across the entire supply chain through better coordination and more cooperation are in the joint benefit of all parties involved, the joint single-vendor single-buyer inventory problem (J-SVSB-IP) has received an extensive attention in the literature. While some of the previous research on the joint vendor-buyer problem focused on the production shipment with constant demand and lead time, in some research there is no constraint and the simplest form of the joint single vendor-single buyer problem is assumed.

One of the earliest researches on the J-SVSB-IP is due to Banerjee [3], in which he assumed that the vendor played the role of a manufacturing company with a finite production rate and considered a lot-for-lot model to satisfy the buyer requests as separate batches. Moreover, various types of J-

SVSB-IP were considered by Goyal [12], Goyal and Gupta [13], and Lu [30]. Hill [18] investigated an unequal shipment policy for the J-SVSB-IP and concluded that an optimal policy for this problem is to use shipment sizes that increase by a fixed factor in the beginning and then remaining constant after a well-specified number of shipments. Ouyang et al. [32] extended the Ben-Daya and Raouf's [4] model in which shortages were allowed and the total amount of stock-outs was considered as a mixture of back orders and lost sales. Hsiao and Lin [19] investigated an economic order quantity model on Stackelberg game in supply chain; that is, a distribution channel system containing one supplier and a single retailer such that the supplier in the channel holds monopolistic status, in which he not only owns cost information about the retailer but also has the decision making right of the lead time. Hariga and Ben-Daya [16] developed a continuous review inventory model where the reorder point, the ordering quantity, and the lead time were the decision variables.

On the joint single-vendor multiple-buyer inventory problem, Siajadi et al. [38] presented a new methodology to obtain the joint economic lot size in the case where multiple buyers are demanding one type of item from a single vendor. They found the shipment policy and proposed a new model to minimize the joint total relevant cost for both vendor and buyer(s). Wee and Yang [48] developed an optimal pricing and replenishment policies in a supply chain system for a single vendor and multiple buyers. They showed since it benefits the vendor more than the buyers in the integrated system, a pricing strategy with price reduction is incorporated to entice the buyers to accept the minimum total cost integrated system. Kim et al. [24] proposed centralized and decentralized adaptive inventory control models for a supply chain consisting of one supplier and multiple retailers. The objectives of the models were to satisfy a target service level predefined for each retailer. The inventory-control parameters of the supplier and retailers were safety lead time and safety stocks, respectively.

For more complex supply chains, Su et al. [39] considered a chain system where facilities produce intermediate or end items that were shipped to other facilities or customers. They considered a complicated production process that can be convergent, divergent, and circulatory. Moreover, Heydari et al. [17] investigated lead time variability in a serially connected supply chain with four levels.

On the variable lead time researches, Lodree et al. [29] investigated a supply chain system in an uncertain demand and variable lead time setting that encompasses customer waiting costs as well as traditional plant costs (i.e., production and inventory costs). Wang [47] provided a study of examining two aspects of supply chain flexibility: order quantity and lead time flexibilities, which have been clarified as the two most common changes which occur in supply chains. Wu [49] analytically studied the implication of a controllable lead time and a random supplier capacity on the continuous review inventory policy, in which the order quantity, reorder point, and lead time were decision variables. Ouyang and Chuang [33] considered a mixture periodic review inventory model in which both the lead time and the review period were considered decision variables. Instead of having a stock-out term in the objective function, a service level constraint was added to the model.

Meta-heuristic algorithms have been proposed to solve some of the existing developed inventory models in the literature. Some of these algorithms are: simulating annealing [1, 40], threshold accepting [7], Tabu search [20], genetic algorithms [2, 41, 44], neural networks [9], ant colony optimization [6], fuzzy simulation [43], evolutionary algorithm [25, 45], and harmony search [10, 26, 42]. Examples of researches on particle swarm optimization (PSO) algorithm can be found in [14, 27, 36, 50], and Rahimi-Vahed et al. [34]. Furthermore, there are some applications of the PSO algorithm on constraint programming of supply chain problems in the literature (see for example [21, 51]).

In this paper, a constraint joint single vendor-single buyer inventory problem is considered in which besides the assumptions of Ben-Daya and Hariga [5] research, the demand follows a uniform probability distribution, the lead time is variable, and there exists a linear relation between the lead time and the lot size. Furthermore, the improvements of the current research over the Ben-Daya and Hariga's research [5] are: (1) a combination of back order and lost sale is considered, (2) service rate is a chance-constraint for each product, (3) multi-product inventory system is assumed, (4) orders should be placed in multiple of packets, (5) the buyer has a limited budget to purchase the products, and (6) a different meta-heuristic solution algorithm is proposed to solve the model.

The remainder of the paper is organized as follows. In Section 2, the problem is defined in details. While in Section 3, the problem is modeled, the meta-heuristic solution algorithm of particle swarm is presented in section 4. Section 5 contains a numerical example along with the obtained benchmarking the model using different meta-heuristic algorithms. In Section 6, the proposed PSO algorithm is illustrated for the given numerical example. Section 7 contains a sensitivity analysis on the model parameters, and finally the conclusion comes in Section 8.

2 Problem definition

Consider a J-SVSB-IP in which the buyer is using the classical (r, Q) continuous review inventory policy and the demand is stochastic. Assume that the lead time is a function of the production lot size. In particular, let the lead time of each product be proportional to the corresponding lot size produced by the vendor plus a fixed delay due to transportation, nonproductive time, etc.

The relationship between the vendor and the buyer is described as follows: for the ith product, i=1, 2,..., p, the buyer orders a lot of size Q_i to the vendor and incurs an ordering cost A_i. The vendor manufactures the product in lots with a finite production rate P_i and incurs a setup cost C_{A_v}. Then, the buyer receives n_i^o shipments, each containing M_i packets of n_i products, $Q_i=n_iM_i$, and incurs a transportation cost for each shipment A_i^t. The buyer places his order when his on-hand inventory of the ith product reaches a reorder point r_i. Moreover, there are lower limits on the service rate of each product as chance constraints and that the buyer has a limited amount of budget to purchase the products TB. Shortages are allowed and incur in combination of back order and lost sale. The elements of the buyer cost function are fixed order, holding, shortage, and transportation costs. The transportation cost is constant for each shipment. For the vendor, the set-up and holding costs are considered. The objective of this research is to determine the reorder points, the order quantities and the number of shipments for each product such that the expected total cost of the supply chain is minimized.

3 Modeling

To model the problem, let us to first define the parameters and the variables of the model. Then based on these definitions, the buyer's, the vendor's, and the chain total costs are derived in Sections 3.2, 3.3, and 3.4, respectively. The constraints are next introduced in Section 3.5, and finally the mathematical model of the problem is given in Section 3.6.

3.1 Parameters and variables

For a specific product i, i=1, 2,..., p, let define the parameters and the variables of the model as:

r_i	reorder point of the ith product (a decision variable)
n_i^o	number of shipments of the ith product from the vendor to the buyer (a decision variable)
M_i	expected number of the packets for the ith product order (a decision variable)
n_i	number of the ith product in each packet
Q_i	expected amount of the ith product order (a decision variable in which, $Q_i=n_iM_i$)
SS_i	safety stock of the ith product (a decision variable)
D_i	expected demand quantity of the ith product, i.e., $D_i=\frac{D_i^{\mathrm{Min}}+D_i^{\mathrm{Max}}}{2}$
$f_{D_i}(d_i)$	probability density functions of D_i (a Uniform density function with parameters D_i^{Min}, D_i^{Max})
P_i	constant production rate of the ith product ($P_i \geq D_i$)
SL_i	the lower limit of the service level for the ith product
β_i	percentage of unsatisfied demands of the ith product that is back ordered
π_i	the back-order cost per unit demand of the ith product
$\widehat{\pi}_i$	shortage cost for each unit of the ith product that is lost sale
A_i	constant cost per order of the ith product
A_i^t	buyer's constant transportation cost per shipment of the ith product
A_i^p	buyer's constant production cost for each setup of the ith product
h_i^v	vendor's holding cost per unit time per unit of the ith product
h_i^b	buyer's holding cost per unit time per unit of the ith product
$\overline{b}_i(r,Q)$	expected amount of the ith product shortage
B_i	expected amount of the ith product back order $\left(B_i=\beta_i\overline{b}_i(r,Q)\right)$
L_i	Expected amount of the ith product lost sale $\left(L_i=(1-\beta_i)\overline{b}_i(r,Q)\right)$
I_i	expected amount of the ith product inventory
LT_i	the ith product lead time is assumed to be $\mathrm{LT}_i=\frac{Q_i}{P_i}+\gamma_i$, where γ_i denotes a fixed delay due to transportation, production time of other products scheduled during the lead time on the same facility, etc.
TB	buyer's total available budget
C_i	purchasing price per unit of the ith product
C_{H_b}	expected total holding cost of the buyer
C_{H_v}	total holding cost of the vendor
C_{B_b}	buyer's expected total shortage cost in back-order state
C_{L_b}	buyer's expected total shortage cost in lost sale state
C_{A_b}	buyer's expected total order cost
C_{A_v}	vendor's expected total set up cost
C_{T_b}	buyer's expected total transportation cost
TC_b	buyer's expected total cost
TC_v	vendor's expected total cost
TC	expected total cost of the supply chain

3.2 The buyer's total costs

In spite of Ben-Daya and Hariga [5] who modeled the J-SVSB-IP in the case of the stochastic demand and variable lead time, in this research, the J-SVSB-IP is formulated: allowing shortages as combinations of back orders and lost sales. Furthermore, there exit transportation costs; service rate is a chance-constraint for each product; multi-product inventory system is assumed; orders should be placed in multiple of packets, and the buyer has a limited budget to purchase the products. To model the problem, a model for a single product i is first developed and then it is extended to include several products.

The buyer's total expected cost per unit time is given by:

$$\mathrm{TC}_b = C_{A_b} + C_{H_b} + C_{B_b} + C_{L_b} + C_{T_b} \tag{1}$$

Referring to Fig. 1, C_{A_b} and C_{T_b} will be calculated as [5]:

$$C_{A_b} = \frac{A_i D_i}{n_i^o Q_i} \tag{2}$$

$$C_{T_b} = \frac{D_i}{Q_i} A_i^t \tag{3}$$

Since the demand is stochastic, according to Hadley–Whitin's [15] the expected on hand inventory per unit time is:

$$I_i = \frac{Q_i}{2} + \mathrm{SS}_i \tag{4}$$

Where SS_i is obtained as (see Appendix 1):

$$\mathrm{SS}_i = r_i - D_i\left(\frac{Q_i}{P_i} + \gamma_i\right) + \frac{(1-\beta_i)\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2\left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)} \tag{5}$$

Hence,

$$C_{H_b} = h_i^b\left(\frac{Q_i}{2} + r_i - D_i\left(\frac{Q_i}{P_i} + \gamma_i\right) + \frac{(1-\beta_i)\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2\left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)}\right) \tag{6}$$

Furthermore, the shortage costs in both back-order and lost-sale states are (see Appendix 2):

$$C_{B_b} = \frac{\pi_i \beta_i D_i \left(D_i^{\mathrm{Max}} - r_i\right)^2}{2 n_i^o Q_i \left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)} \tag{7}$$

$$C_{L_b} = \frac{\widehat{\pi}_i (1-\beta_i) D_i \left(D_i^{\mathrm{Max}} - r_i\right)^2}{2 n_i^o Q_i \left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)} \tag{8}$$

Finally, the buyer's total expected total cost per unit time for the ith product becomes

$$\begin{aligned}\mathrm{TC}_b &= C_{A_b} + C_{H_b} + C_{B_b} + C_{L_b} + C_{T_b} \\ &= \frac{A_i D_i}{n_i^o Q_i} + h_i^b\left(\frac{Q_i}{2} + r_i - D_i\left(\frac{Q_i}{P_i} + \gamma_i\right) + \frac{(1-\beta_i)\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2\left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)}\right) \\ &\quad + \frac{\pi_i \beta_i D_i \left(D_i^{\mathrm{Max}} - r_i\right)^2}{2 n_i^o Q_i \left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)} + \frac{\widehat{\pi}_i (1-\beta_i) D_i \left(D_i^{\mathrm{Max}} - r_i\right)^2}{2 n_i^o Q_i \left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)} + \frac{D_i}{Q_i} A_i^t\end{aligned} \tag{9}$$

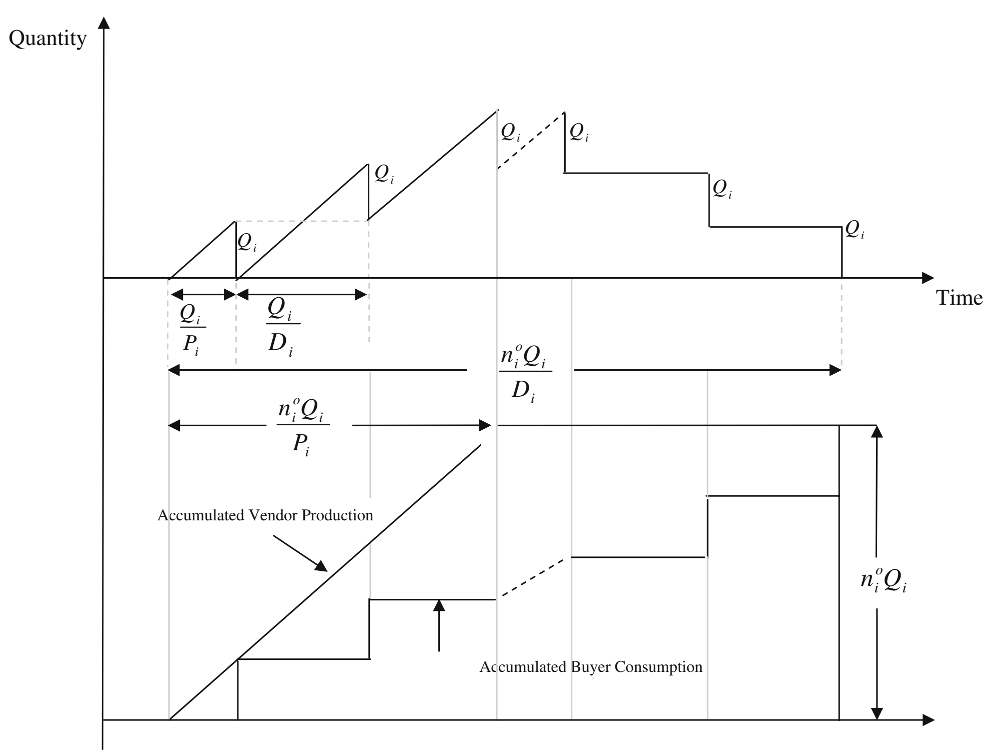

Fig. 1 The vendor's inventory graph

3.3 The vendor's total costs

Ben-Daya and Hariga [5] developed the vendor's total costs by calculating C_{A_v} and C_{H_v} as:

$$C_{A_v} = \frac{A_i^p D_i}{n_i^o Q_i} \tag{10}$$

$$C_{H_v} = h_i^v \frac{Q_i}{2}\left[n_i^o\left(1-\frac{D_i}{P_i}\right)-1+2\frac{D_i}{P_i}\right] \tag{11}$$

So, the vendor's total cost for the ith product becomes:

$$\mathrm{TC}_v = C_{A_v} + C_{H_v} = \frac{A_i^p D_i}{n_i^o Q_i} + h_i^v \frac{Q_i}{2}\left[n_i^o\left(1-\frac{D_i}{P_i}\right)-1+2\frac{D_i}{P_i}\right] \tag{12}$$

3.4 The chain's total costs

According to Eqs. 9 and 12, the chain's total expected cost for the ith product is:

$$\begin{aligned}\mathrm{TC} &= \mathrm{TC}_b + \mathrm{TC}_v = C_{A_b} + C_{H_b} + C_{B_b} + C_{L_b} + C_{T_b} + C_{A_v} + C_{H_v} \\ &= \tfrac{A_i D_i}{n_i^o Q_i} + h_i^b\left(\tfrac{Q_i}{2} + r_i - D_i\left(\tfrac{Q_i}{P_i}+\gamma_i\right) + \tfrac{(1-\beta_i)\left(D_i^{\mathrm{Max}}-r_i\right)^2}{2\left(D_i^{\mathrm{Max}}-D_i^{\mathrm{Min}}\right)}\right) \\ &\quad + \frac{\pi_i\beta_i D_i\left(D_i^{\mathrm{Max}}-r_i\right)^2}{2n_i^o Q_i\left(D_i^{\mathrm{Max}}-D_i^{\mathrm{Min}}\right)} + \frac{\widehat{\pi}_i(1-\beta_i)D_i\left(D_i^{\mathrm{Max}}-r_i\right)^2}{2n_i^o Q_i\left(D_i^{\mathrm{Max}}-D_i^{\mathrm{Min}}\right)} \\ &\quad + \frac{D_i}{Q_i}A_i^t + \frac{A_i^p D_i}{n_i^o Q_i} + h_i^v\frac{Q_i}{2}\left[n_i^o\left(1-\frac{D_i}{P_i}\right)-1+2\frac{D_i}{P_i}\right]\end{aligned} \tag{13}$$

3.5 The constraints

In order to model the service level as a chance constraint, Liu [28] first defined a stochastic constraint function in the form of $G(X, Y)$ in which X and Y were decision and stochastic vectors, respectively. Since $G(X, Y)\leq 0$ does not define a deterministic feasible set, he then introduced a confidence level α at which it was desired the stochastic constraint $G(X, Y)\leq 0$ hold. In other words, he defined a chance constraint as $P\{G(X,Y)\leq 0\}\geq\alpha$. For the problem at hand since the shortages of the ith product only occur when the demand is more than the reorder point and that the lower limit for the service level is SL_i, then (see Appendix 3):

$$r_i \geq \left(\frac{Q_i}{P_i}+\gamma_i\right)\left[\left(D_i^{\mathrm{Max}}-D_i^{\mathrm{Min}}\right)SL_i + \left(D_i^{\mathrm{Min}}\right)\right] \tag{14}$$

However, the orders to be placed in packets of size n_i require:

$$Q_i = n_i M_i \tag{15}$$

Furthermore, since the purchasing price per unit of the ith product is C_i, the order quantity of the ith product is Q_i and the total budget of the buyer is TB, then the budget constraint will be:

$$\sum_{i=1}^{p} C_i Q_i \leq \mathrm{TB} \tag{16}$$

3.6 The supply chain multi-constraint multi-product model

Based on Eqs. 13, 14, 15, and 16, the multi-product multi-constraint inventory model of the supply chain can be easily obtained as:

$$\begin{aligned}\mathrm{MinTC} = &\sum_{i=1}^{p}\tfrac{\left(A_i+A_i^p\right)D_i}{n_i^o n_i M_i} + \sum_{i=1}^{p} h_i^b \\ &\left[\tfrac{n_i M_i}{2} + r_i - D_i\left(\tfrac{n_i M_i}{P_i}+\gamma_i\right) + \tfrac{(1-\beta_i)\left(D_i^{\mathrm{Max}}-r_i\right)^2}{2\left(D_i^{\mathrm{Max}}-D_i^{\mathrm{Min}}\right)}\right] \\ &+ \sum_{i=1}^{p}\tfrac{\left(\pi_i\beta_i+\widehat{\pi}_{ii}(1-\beta_i)\right)D_i\left(D_i^{\mathrm{Max}}-r_i\right)^2}{2n_i^o n_i M_i\left(D_i^{\mathrm{Max}}-D_i^{\mathrm{Min}}\right)} + \sum_{i=1}^{p}\tfrac{D_i}{n_i M_i}A_i^t \\ &+ \sum_{i=1}^{p} h_i^v \tfrac{n_i M_i}{2}\left[n_i^o\left(1-\tfrac{D_i}{P_i}\right)-1+\tfrac{2D_i}{P_i}\right]\end{aligned} \tag{17}$$

$$\begin{aligned} s.t.:\quad &\sum_{i=1}^{p} C_i n_i M_i \leq TB \\ &Q_i = n_i M_i && \forall i;\ i=1,2,\cdots,p \\ &r_i \geq \left(\tfrac{n_i M_i}{P_i}+\gamma_i\right)\left[\left(D_i^{\mathrm{Max}}-D_i^{\mathrm{Min}}\right)SL_i + \left(D_i^{\mathrm{Min}}\right)\right] \\ & && \forall i;\ i=1,2,\cdots,p \\ &\mathrm{SS}_i = r_i - D_i\left(\tfrac{n_i M_i}{P_i}+\gamma_i\right) + \tfrac{(1-\beta_i)\left(D_i^{\mathrm{Max}}-r_i\right)^2}{2\left(D_i^{\mathrm{Max}}-D_i^{\mathrm{Min}}\right)} \\ & && \forall i;\ i=1,2,\cdots,p \\ &Q_i, n_i^o, r_i, M_i \geq 0 \text{ Integer} && \forall i;\ i=1,2,\cdots,p\end{aligned}$$

In the next section, a solution algorithm is proposed to solve the model in (17).

4 A solution algorithm

In most inventory models that have been developed so far, researchers have tried to consider some constraints such as defective items, shortages, back orders, and so on such that the objective function of the model becomes concave and the model can easily be solved by some mathematical approaches like the Lagrangian or the derivative methods. However, since the model in (17) is integer-nonlinear in nature, reaching an analytical solution (if any) to the problem is difficult [11]. In addition, efficient treatment of integer nonlinear optimization is one of the most difficult problems in practical optimization [8].

Many researchers have successfully used meta-heuristic methods to solve complicated optimization problems in different fields of scientific and engineering disciplines; among them, the particle swarm optimization algorithm is one of the most efficient methods. Accordingly, in the next section, a particle swarm meta-heuristic algorithm is proposed.

4.1 Particle swarm optimization

New ways have been found to optimize problems for less than a century, but nature has used various ways of optimization for millions of million years. Recently scientists mimicked nature to solve different kinds of complex optimization problems. Most of these problems are so complicated and time consuming that one cannot use an exact algorithm to solve them. Thus, typically some non-precise algorithms are used to find a near-optimum solution in a shorter period. These algorithms are called heuristic. Furthermore, a meta-heuristic algorithm is a heuristic method to solve a very general class of computational problems by combining user-given black box procedures (usually heuristics themselves) in the hope of obtaining a more efficient or more robust procedure. These algorithms are generally applied to problems for which there is no satisfactory problem-specific algorithm or heuristic; or when it is not practical to implement such a method [6].

PSO was proposed by Kennedy and Eberhart [22] in the mid-1990s while attempting to simulate the choreographed, graceful motion of swarms of birds as part of a socio-cognitive study investigating the notion of "collective intelligence" in biological populations. In PSO, a set of randomly generated solutions (initial swarm) propagates in the design space towards the optimal solution over a number of iterations (moves) based on large amount of information about the design space that is assimilated and shared by all members of the swarm. PSO is inspired by the ability of flocks of birds, schools of fish, and herds of animals to adapt to their environment, find rich sources of food, and avoid predators by implementing an "information sharing" approaches [23].

4.2 The proposed PSO algorithm

The PSO algorithm consists of three main steps; *generating particle's positions and exploration velocities, updating exploration velocity, and position update*. These parts will be described in Sections 4.3, 4.4 and 4.5, respectively.

4.3 Initializing particles' positions and exploration velocities

A particle refers to a point in the designed 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. In this research, there are three decision variables (r_i, M_i, and n_i^o) for each product. As PSO is a population-based optimization algorithm, each particle is an individual and the swarm is composed of particles. Mapping between swarm and particles in PSO is similar to the relationship between population and chromosomes in genetic algorithm. The swarm size will be denoted by N. Using the upper and the lower bounds on the design variables' values, $X_{\min}$ and $X_{\max}$, the positions, X_k^i, and the exploration velocities, V_k^i, of the initial swarm of particles can be first randomly generated. The positions and exploration velocities are given in a vector format where the superscript and subscript denote the ith particle in the population at kth 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. Equations 18 and 19 are used to initialize particles, in which Δt is the constant time increment and assumed to be 1.

$$X_0^i = X_{\min} + \mathrm{Rand}(X_{\max} - X_{\min}) \tag{18}$$

$$V_0^i = \frac{X_{\min} + \mathrm{Rand}(X_{\max} - X_{\min})}{\Delta t} = \frac{\text{position}}{\text{time}} \tag{19}$$

The initialization is a very important process of the PSO algorithm to become convergent. Two common methodologies to generate the initial solution are: (1) generating feasible solutions or (2) random generation. In this paper, to generate the initial solution, the first methodology is used. In other words, since a solution vector, because of its exploration velocity, may dissatisfy a constraint, the feasibility of a generated solution in each step is checked. As a result, if a solution vector does not satisfy a constraint, the related vector solution will be punished by a big penalty on its fitness.

4.4 Updating the exploration velocities

Using the fitness values that are obtained based on the particles current positions in the design space at time k, the exploration velocities of all particles at time k+1 are updated. The fitness function value of a particle not only determines which particle has the best global value in the current swarm or population, P_k^g, but also determines the best position of each particle over time, P^i, in the current and all previous moves. The exploration velocity update formula uses these two pieces of information for each particle in the swarm along with the effect of current motion, V_k^i, to provide a search direction, V_{k+1}^i for the next iteration and to ensure good coverage of the design space and avoid entrapment in local optima.

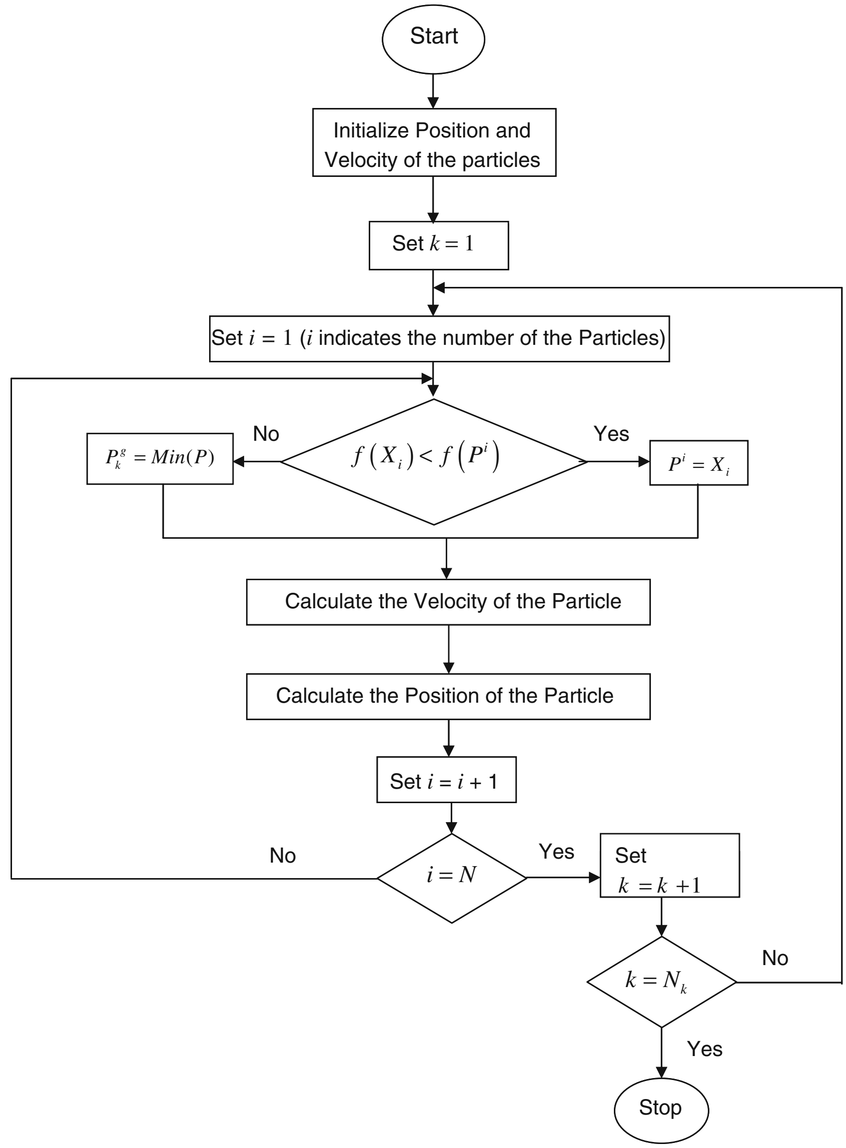

Fig. 2 The flowchart of the proposed PSO algorithm

The exploration velocity update formula includes some random parameters, rand, represented by the uniformly distributed variables. The three values that affect the new search direction, namely, current motion, particle own memory, and swarm influence, are incorporated via a summation approach as shown in Eq. 20 with three weight factors, namely, inertia factor, w, self-confidence factor, C_1, and swarm confidence factor, C_2, respectively. The constraint in Eq. 21 formed by $V_{\max}$, a maximum exploration velocity, is specified to clamp the excessive accelerations.

$$\underbrace{V_{k+1}^{i}}_{\substack{Velocity\ of\ Particle\\ i\ at\ time\ k+1}} = \overbrace{w}^{[0.4,1.4]}\ \underbrace{V_k^i}_{\substack{Current\\ Motion}} + \underbrace{\overbrace{C_1}^{[1.5,2]}\ Rand\,\frac{(P^i - X_k^i)}{\Delta t}}_{Particle\ Memory\ Influence} + \underbrace{\overbrace{C_2}^{[2,2.5]}\ Rand\,\frac{(P_k^g - X_k^i)}{\Delta t}}_{Swarm\ Influence} \tag{20}$$

$$\begin{aligned} &\text{if}\left(V_{k+1}^i > V_{\max}\right); \quad V_{k+1}^i = V_{\max} \\ &\text{if}\left(V_{k+1}^i < -V_{\max}\right); \; V_{k+1}^i = -V_{\max} \end{aligned} \tag{21}$$

The inertia weight w controls how much of the previous exploration velocity should be retained from the previous step. A larger inertia weight facilitates a global

Table 1 General data of the example

Prod.	n_i	D_i^{Min}	D_i^{Max}	μ_i	γ_i	P_i	SL_i	β_i	C_i	π_i	$\widehat{\pi}_i$	A_i^o	A_i^t	A_i^p	h_i^v	h_i^b
1	1	800	120	1,000	0.01	3,500	0.5	0.5	60	10	12	50	25	400	5	6
2	2	1,300	1,700	1,500	0.01	3,500	0.5	0.05	60	10	12	50	25	400	5	6
3	5	1,800	2,200	2,000	0.01	3,500	0.5	0.5	60	10	12	50	25	400	5	6
4	6	2,300	2,700	2,500	0.01	3,500	0.5	0.5	60	10	12	50	25	400	5	6
5	10	2,800	3,200	3,000	0.01	3,500	0.5	0.5	60	10	12	50	25	400	5	6
6	1	3,300	3,700	3,500	0.02	6,000	0.8	0.9	30	5	7	70	50	600	2	3
7	2	3,800	4,200	4,000	0.02	6,000	0.8	0.9	30	5	7	70	50	600	2	3
8	5	4,300	4,700	4,500	0.02	6,000	0.8	0.9	30	5	7	70	50	600	2	3
9	6	4,800	5,200	5,000	0.02	6,000	0.8	0.9	30	5	7	70	50	600	2	3
10	10	5,300	5,700	5,500	0.02	6,000	0.8	0.9	30	5	7	70	50	600	2	3

search, while a smaller inertia weight facilitates a local search. A balance can be achieved between global and local 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 [31, 37]. The linear distribution of the inertia weight is expressed as follows [31]:

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} \times k \tag{22}$$

where, $w_{\max}$ and $w_{\min}$ are the initial and final values of weighting coefficient, and $k_{\max}$ and k are the maximum iteration number and iteration counter, respectively. The related results of the two parameters $w_{\max}$=0.9 and $w_{\min}$= 0.4 were investigated by Shi and Eberhart [37] and Naka et al. [31].

Salman et al. [35] used the values of 0.9, 2, and 2 for w, C_1 and C_2 respectively, and suggested upper and lower bounds on these values shown in Eq. 20. Other combinations of the parameter values usually lead to much slower convergence or sometimes non-convergence at all. The tuning of the PSO algorithm weight factors is a topic that warrants proper investigation but is outside the scope of this work. Meanwhile, according to some suggestions in the literature (see [31, 37]) in this research Eq. 22 for w is considered. Also different values for C_1, C_2 and N (population size) are considered and all possible combinations of them are examined. Finally, the best combination of the parameters from all of the considered values in Table 3 is chosen.

4.5 Updating the position

Position update is the final step of a PSO-iteration and is performed using the current particle position and its own updated exploration velocity vector shown in Eq. 23.

$$X_{K+1}^i = X_K^i + V_{K+1}^i \Delta t \tag{23}$$

In short, the pseudo code of the proposed PSO algorithm is:

Initialize position (X_0) and velocity of N particles
P^1=X_0
DO

k=1
FOR i=1 to N particles

IF $f(X_i)$<$f(P^i)$ THEN P^i=X_i
Calculate new velocity of the particle
Calculate new position of particle
$P_k^g = \min(P)$
END FOR
$k = k + 1$

UNTIL a sufficient good criterion is met.

Figure 2 shows the flowchart of the proposed algorithm.

In the next section, a numerical example is given to illustrate the application of the proposed PSO algorithm in a real world environment.

5 A numerical example

Consider a multi-product inventory control problem with ten products and general data given in Tables 1 along with

Table 2 The parameters of the PSO algorithm

C_2	C_1	N
1.5	1.5	10
2	1.75	50
2.5	2	100

Table 3 The best result of the PSO algorithm

Product	Buyer					Vendor		Chain
	r_i	M_i	Q_i	SS_i	TC(b)	n_i^o	TC (v)	TC = TC(b) + TC(v)
1	137	371	371	73	182,400	11	71,540	253,940
2	199	195	390	1,425		15		
3	603	179	895	167		10		
4	2,245	165	990	164		13		
5	2,543	149	1,490	150		8		
6	697	956	956	119		15		
7	706	394	788	162		15		
8	1,482	331	1,655	144		12		
9	1,819	270	1,620	180		7		
10	4,647	342	3,420	154		10		

TB=500,000. Each particle has 30 variables (three decision variables for each of the 10 products). Table 2 shows the different values of the PSO parameters used to obtain the solution. In this research, all of the possible combinations of the PSO parameters in Table 2 are employed and using the *min* (*min*) criterion, the best combination of the parameters has been selected. The best combination of the PSO parameters for this example is obtained as $C_1=C_2=2$, $N=100$.

In order to show that the proposed PSO algorithm is an effective means of solving the complicated inventory model of this research, the genetic algorithm and simulated annealing approaches of Taleizadeh et al. [46] are also employed to solve the numerical example. The best results obtained by the three methods are given in Tables 3, 4, and 5. In these tables, SS_i and Q_i are the safety stock and the order quantity given in Eqs. 5 and 15, respectively.

A comparison of the results in Tables 3, 4, and 5 show that the PSO algorithm performs better than the genetic algorithm (GA) and the simulated annealing (SA) algorithms in terms of the objective function values. Furthermore, the convergence paths of the best result of the objective function for all methods are shown in Figs. 3, 4, and 5.

In order to show how the proposed PSO algorithm works, in the next section, the first two iterations are described in details.

6 Illustration of the proposed PSO algorithm

The illustration of the first two iterations of the PSO algorithm for the numerical example is as follows:

6.1 Particles initialization

Since 100 particles were found as the best population size in this problem, they are randomly generated for the first iteration employing Eqs. 18 and 19 and using the lower and the upper bounds on M_i, n_i^o, and r_i that are given in Table 6.

Table 4 The best result of the GA

Product	Buyer					Vendor		Chain
	r_i	M_i	Q_i	SS_i	TC (b)	n_i^o	TC (v)	TC = TC(b) + TC(v)
1	85	203	203	794	201,152	15	67,220	268,372
2	320	329	658	1,213		9		
3	445	142	710	1,944		11		
4	1,792	171	1,026	1,549		11		
5	2,580	161	1,630	1,393		9		
6	368	488	488	1,401		15		
7	685	430	860	1,576		15		
8	1,358	328	1,640	1,434		15		
9	2,135	351	2,106	1,454		15		
10	4,290	297	2,970	1,706		11		

Table 5 The best result of the SA algorithm

Product	Buyer					Vendor		Chain
	r_i	M_i	Q_i	SS_i	TC (*b*)	n_i^o	TC (*v*)	$TC = TC(b) + TC(v)$
1	120	281	281	759	240,620	10	62,660	303,280
2	310	321	642	1,227		7		
3	589	160	800	1,734		10		
4	887	175	1,050	2,166		10		
5	1,735	181	1,810	1,495		11		
6	363	472	472	1,409		12		
7	627	378	756	1,639		12		
8	1,700	385	1,925	1,291		15		
9	2,058	378	2,268	1,302		12		
10	4,840	213	2,130	2,870		13		

Based on the fact that there is a relation between r_i and M_i (see Eqs. 14 and 15), the minimum feasible value of r_i for a particle can be calculated based on its M_i. Moreover, using the lower and the upper bounds given in Table 6, r_i can then be randomly generated. Hence, the calculation of M_1, n_1^o and r_1 are described as follows:

$$M_1 = 100.00 + (1000.00 - 0.00) \times 0.698 = 998$$
$$n_1^o = 1.00 + (20.00 - 0.00) \times 0.5 = 11$$
$$r_1 \geq \left[\left(\frac{998 \times 1}{3500}\right) + 0.01\right] \times [((1200 - 800) \times 0.5) + 800] = 296$$

The value of r_1 is then randomly generated between 296 and 6,000 as r_1=347.

Similarly, all 100 particles are generated using different random numbers. Based on the lower and the upper bounds of the variables, the initial exploration velocity vector of the first particle in the initial iteration is shown in Table 7.

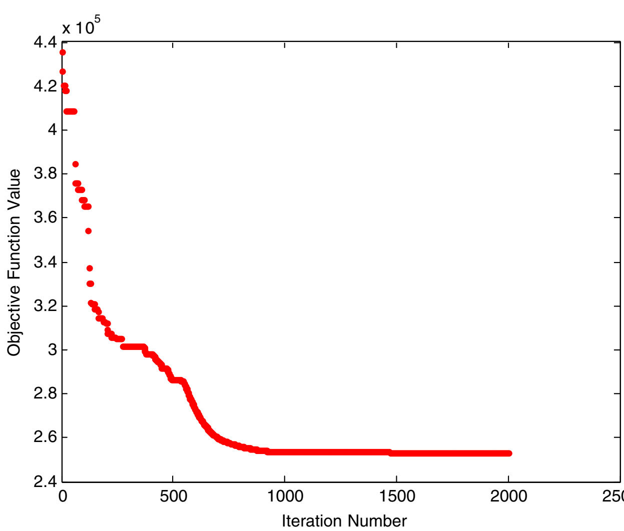

Fig. 3 Convergence path of the objective function by PSO

6.2 Particle updating

In successive iterations, each particle is updated by changing its position using Eqs. 20 and 21. For the first iteration, Table 8 shows the first five particles of 100 randomly generated particles along with their evaluated objective function values.

In the first iteration, P^i values of the particles are their initial values. However, in the next generations, the particles with minimum objective function values are chosen to be the P^i values of the particles. Using Eq. 20 and according to the best parameter combination, C_1=C_2=2, N=100, the exploration velocity calculation for position updating process is derived as follows:

$$V_2^1 = [0.9 \times V_1^1] + [2.0 \times 0.53834 \times (1\text{ st particle} - 1\text{ st particle})] + [2.0 \times 0.9961 \times (78\text{th particle} - 1\text{ st particle})] \quad (24)$$

In the second iteration, the exploration velocity of the first particle is calculated using Eq. 24 and is shown in Table 9 along with its objective function value of 428,989. The velocities and the objective function values of the remaining particles are updated in a similar way and P^i and P_k^g are chosen for the second iterations. In this process, if an updated particle is not feasible, its corresponding objective function value takes a big value as 10,000,000.

7 A sensitivity analysis

To study the effects of the parameter changes on the best result derived by the proposed method, a sensitivity analysis is performed by changing (increasing or decreasing) the parameters D_i, P_i, SL_i, π_i, $\widehat{\pi}_i$, h_h^v and h_i^b by 10% and 40% and taking one parameter at a time, keeping the

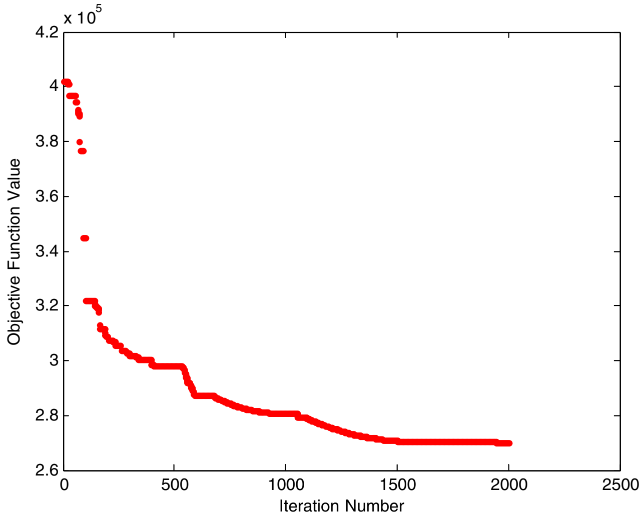

Fig. 4 Convergence path of the objective function by GA

remaining parameters at their original levels. This analysis is performed on the numerical examples given in Section 5. The results of the sensitivity analysis are shown in Table 10.

A careful study of Table 10 reveals the followings in the uniform distribution case:

- The buyer's total cost (TC_b) is highly sensitive to the changes in the expected values of demand rate, and moderately sensitive to the changes in the values of production rate, lost sale cost and holding cost of the buyer. Also, TC_b is slightly sensitive to the changes in the values of service level, back order, and holding costs of the vendor.
- The vendor's total cost (TC_v) is highly sensitive to both changes in the expected value of demand and the value of productions rate, and moderately sensitive to the changes in the value of vendor holding cost. Moreover, TC_v is slightly sensitive to the changes in the value of the service level, back-order cost, lost sale cost, and holding cost of the buyer.
- The TC is highly sensitive to the changes in the expected value of the demand and moderately sensitive to the changes in the value of the production rate, lost sale cost, and holding cost of the buyer. Furthermore, TC is slightly sensitive to the changes in the value of service level, back order, and holding cost of the vendor.

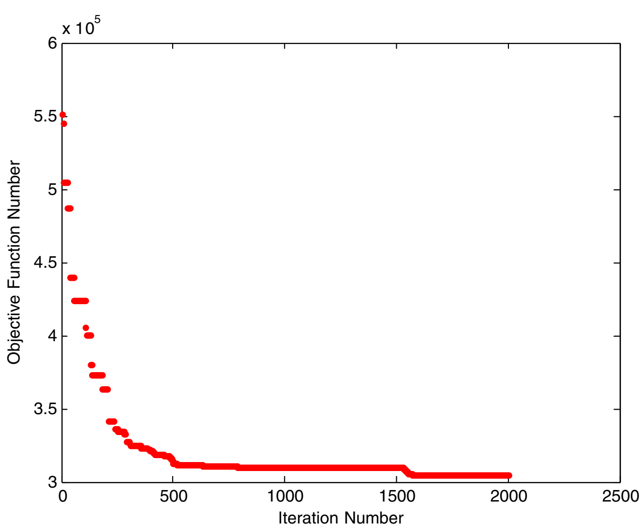

Fig. 5 Convergence path of the objective function by SA

Table 6 Lower and upper bounds for the variables in the initialization step

Variable	Lower limit	Upper limit
M_i	0	1,000
n_i^o	0	20
r_i	Based on Eq. 14	6,000

It should also be mentioned that in the above example, D_i and P_i have the main affects on the system performances.

8 Conclusion and recommendations for future research

In this paper, a multi-product multi-chance constraint joint single buyer-single vendor inventory problem was investigated. The customer demand was assumed stochastic and the lead time varied linearly with respect to the lot size. The orders were assumed to be placed in multiple of packets, the service rate limitation of each product was considered a chance constraint, and that there was a limited budget for the buyer to purchase the products. The aim of the study was to determine the optimal re-order point and the order quantity of each product to minimize the total supply chain cost. A mathematical model that includes the costs of transportation, fixed order, holding, and shortage in both back-order and lost-sale states for the buyer and the set-up and holding costs for the vendor was introduced. As the model was shown to be integer nonlinear, the particle swarm optimization, genetic algorithm, and simulated annealing techniques have been used to benchmark the model. A practical numerical example along with the illustration of the first two iterations of the PSO algorithm was given to demonstrate the applicability of the proposed methodology. A comparison of the results shows that the PSO algorithm performs better than the GA and SA algorithms in terms of the objective function values. Finally, a sensitivity analysis on some model parameters was performed to bring some managerial insights.

Some recommendations for future works are to deploy multi-buyer or multi-vendor cases, assuming fuzzy demands, and considering stochastic or fuzzy lead times.

Table 7 Initial velocity vector of the first particle

Product (i)	1	2	3	4	5	6	7	8	9	10
$V_1^1(M_i)$	556	239	582	578	170	496	117	439	296	453
$V_1^1(n_i^o)$	5	2	7	7	6	14	5	7	6	2
$V_1^1(r_i)$	416	373	178	342	2,810	579	524	478	2,367	2,491

Acknowledgment The authors are thankful for the constructive comments of the reviewers that improved the presentation of the paper.

Appendix 1: the safety stock

$SS_i^{\text{Backorder}}$ and $SS_i^{\text{Lost-Sale}}$ can be calculated in different ways such as:

For the back order, we have:

$$SS_i^{\text{Backorder}} = \int_0^{\infty} (r_i - D_i) f_{D_i}(d_i) dD_i = \int_0^{\infty} r_i f_{D_i}(d_i) dD_i - \int_0^{\infty} D_i f_{D_i}(d_i) dD_i = r_i - \mu_D = r_i - D_i(LT_i)$$

Since all shortages are lost in the lost-sale case, when the demand during the lead time is greater than the re-order point, the safety stock becomes zero. Hence:

$$SS_i^{\text{Lost-Sale}} = \int_0^{r_i} (r_i - D_i) f_{D_i}(d_i) dD_i = \underbrace{\int_0^{\infty} (r_i - D_i) f_{D_i}(d_i) dD_i}_{r_i - D_i(\text{LT}_i)} \underbrace{- \int_{r_i}^{\infty} [-(D_i - r_i)] f_{D_i}(d_i) dD_i}_{-\overline{b}_i(r,Q)}$$
$$= r_i - D_i(\text{LT}_i) + \overline{b}_i(r, Q)$$

Since the expected amount of the ith product back order is $B_i = \beta_i \overline{b}_i(r, Q)$, in combination form of the back order and lost sale we have:

$$\begin{aligned} SS_i^{\text{Combination}} &= SS_i^{\text{Lost-Sale}} - \beta_i \overline{b}_i(r, Q) \\ &= r_i - D_i(\text{LT}_i) + \overline{b}_i(r, Q) - \beta_i \overline{b}_i(r, Q) \\ &= r_i - D_i(\text{LT}_i) + (1 - \beta_i) \overline{b}_i(r, Q) \end{aligned}$$

In other way,

$$\begin{aligned} SS_i^{\text{Combination}} &= \beta_i SS_i^{\text{Backorder}} + (1 - \beta_i) SS_i^{\text{Lost-Sale}} \\ &= \beta_i r_i - \beta_i D_i(\text{LT}_i) + (1 - \beta_i) r_i \\ &\quad - (1 - \beta_i) D_i(\text{LT}_i) + (1 - \beta_i) \overline{b}_i(r, Q) \\ &= r_i - D_i(\text{LT}_i) + (1 - \beta_i) \overline{b}_i(r, Q) \end{aligned}$$

Now, since the demand distribution function is a continuous uniform and shortage occur when the demand is more than the reorder point, the expected shortage will be:

$$\overline{b}_i(r, Q) = \int_{r_i}^{D_i^{\text{Max}}} (X_i - r_i) \frac{1}{D_i^{\text{Max}} - D_i^{\text{Min}}} dD_i$$

$$= \frac{(X_i - r_i)^2}{2(D_i^{\text{Max}} - D_i^{\text{Min}})} \Bigg|_{r_i}^{D_i^{\text{Max}}} = \frac{\left(D_i^{\text{Max}} - r_i\right)^2}{2(D_i^{\text{Max}} - D_i^{\text{Min}})}$$

Table 8 The first five particles of the first iteration

Product (i)	First particle			Second particle			Third particle			Fourth particle			Fifth particle		
	M_i	n_i^o	r_i	M_i	n_i^o	r_i	M_i	n_i^o	r_i	M_i	n_i^o	r_i	M_i	n_i^o	r_i
1	998	11	347	115	6	569	227	7	101	580	3	176	179	6	110
2	567	6	507	148	2	183	545	5.8	530	508	6	494	336	4	322
3	334	4	976	258	5	806	169	4.8	511	299	9	889	238	5	718
4	181	3	840	319	1	1,412	520	2.8	2,267	315	8	1,423	105	6	490
5	364	3	3,160	497	6	4,304	221	6.6	1,977	231	12	2,025	118	8	1,083
6	231	5	244	322	4	299	198	3.4	204	534	11	423	477	11	386
7	474	5	755	477	8	751	336	5.3	562	172	5	361	188	19	363
8	214	1	963	427	4	1,746	392	6.8	1,631	275	6	1,178	436	5	1,181
9	512	2	2,755	349	1	1,938	242	7.4	1,384	137	4	820	131.9	17	809
10	139	7	2,755	392	1	3,801	290	6.2	2,859	157	1	1,922	243.7	14	2,431
Objective unction	468,905			490,356			480,572			493,256			437,296		

Table 9 The first particle of the second iteration

First particle	1	2	3	4	5	6	7	8	9	10
M_i	757	1,148	904	575	606	1,501	782	1,399	640	918
n_i^o	6	4	11	8	6	3	14	15	12	7
r_i	585	1,115	2,102	1,533	3,010	1,274	1,217	4,844	2,328	5,978

Finally, knowing that $\mathrm{LT}_i = \left(\frac{Q_i}{P_i} + \gamma_i\right)$ we have:

$$\mathrm{SS}_i = r_i - D_i\left(\frac{Q_i}{P_i} + \gamma_i\right) + \frac{(1-\beta_i)\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})}$$

Appendix 2: the buyer back order and lost sale costs

According to Appendix (1), we have:

$$\overline{b}_i(r,Q) = \frac{\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})}$$

Table 10 The results of the sensitivity analysis

Parameters	% Changes	% Changes in		
		TC (*b*)	TC (*v*)	*TC*
D_i	+40	+80	−84	+39
	+10	+30	−19	+16
	−10	−16	−6	−13
	−40	−50	−39	−47
P_i	+40	+10	+31	+16
	+10	+17	+8	+15
	−10	+11	−31	−0.65
	−40	Infeasible	Infeasible	Infeasible
SL_i	+40	−4	−4	−4
	+10	+15	−3	+10
	−10	+8	+5	+7
	−40	+7	−3	+4
π_i	+40	+6	+2	+4
	+10	−1	+1	−0.3
	−10	−7	−1	−5
	−40	−12	−10	−12
$\hat{\pi}_i$	+40	+18	−0.7	+12
	+10	+5	−0.5	+2
	−10	−12	−2	−9
	−40	−16	−11	−13
h_i^v	+40	+6	+27	+10
	+10	+3	+12	+7
	−10	−3	−10	−2
	−40	−5	−22	−5
h_i^b	+40	+25	+3	+19
	+10	+7	+0.3	+4
	−10	−3	−3	−3
	−40	−29	−2	−22

Knowing that:

$$B_i = \beta_i \frac{D_i}{n_i^o Q_i}\overline{b}_i(r,Q) = \beta_i \frac{D_i}{n_i^o Q_i}\frac{\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})}$$

$$L_i = (1-\beta_i)\frac{D_i}{n_i^o Q_i}\overline{b}_i(r,Q) = (1-\beta_i)\frac{D_i}{n_i^o Q_i}\frac{\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})}$$

The expected costs of back order and lost sale based on the expressions for π_i and $\hat{\pi}_i$ are:

$$C_{B_b} = \frac{\pi_i \beta_i D_i\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2n_i^o Q_i(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})}$$

$$C_{L_b} = \frac{\hat{\pi}_i(1-\beta_i)D_i\left(D_i^{\mathrm{Max}} - r_i\right)^2}{2n_i^o Q_i(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})}$$

Appendix 3: service rate as a chance constraint

Knowing that if $X \sim U[a,b]$ then $Y = KX \sim U[Ka, Kb]$, the distribution function of $D_i(LT_i)$ is uniform on $\left[(\mathrm{LT}_i)D_i^{\mathrm{Max}}, (\mathrm{LT}_i)D_i^{\mathrm{Min}}\right]$. Since the shortages of the *i*th product only occur when the demand during the lead time is more than the reorder point and the lower limit for the service level is SL_i, then:

$$P(D_i(\mathrm{LT}_i) \le r_i) \ge \mathrm{SL}_i$$

$$\Rightarrow \int_{\mathrm{LT}_i\left(D_i^{\mathrm{Min}}\right)}^{r_i} \frac{1}{\mathrm{LT}_i(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})} d_i \ge \mathrm{SL}_i$$

$$\Rightarrow \frac{r_i - \mathrm{LT}_i\left(D_i^{\mathrm{Min}}\right)}{LT_i(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})} \ge \mathrm{SL}_i$$

$$\Rightarrow r_i \ge LT_i(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}})\mathrm{SL}_i + \mathrm{LT}_i\left(D_i^{\mathrm{Min}}\right)$$

$$\Rightarrow r_i \ge LT_i\left[\left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)\mathrm{SL}_i + \left(D_i^{\mathrm{Min}}\right)\right]$$

$$\Rightarrow r_i \ge \left(\frac{Q_i}{P_i} + \gamma_i\right)\left[\left(D_i^{\mathrm{Max}} - D_i^{\mathrm{Min}}\right)\mathrm{SL}_i + \left(D_i^{\mathrm{Min}}\right)\right]$$

References

1. Aarts EHL, Korst JHM (1989) Simulated annealing and Boltzmann machine; a stochastic approach to computing, 1st edn. Wiley, Chichester, UK
2. Al-Tabtabai H, Alex AP (1999) Using genetic algorithms to solve optimization problems in construction. Eng Con Arch Man 6:121–132
3. Banerjee A (1986) A joint economic lot size model for purchaser and vendor. Decis Sci 17:292–311
4. Ben-Daya M, Raouf A (1994) Inventory models involving lead time as a decision variable. Journal of the Operational Research Society 45:579–582
5. Ben-Daya M, Hariga M (2004) Integrated single vendor single buyer model with stochastic demand and variable lead time. Int J Prod Econ 92:75–80
6. Dorigo M, Stutzle T (2004) Ant colony optimization. MIT Press, Cambridge
7. Dueck G, Scheuer T (1990) Threshold accepting: a general purpose algorithm appearing superior to simulated annealing. J Comput Phys 90:161–175
8. El-Sharkawi L (2008) Modern heuristic optimization techniques, 1st edn. Wiley Inter Science, New Jersey
9. Gaiduk AR, Vershinin YA, West MJ (2002) Neural networks and optimization problems. In Proceedings of IEEE 2002 International Conference on Control Applications, 1:37–41
10. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76:60–68
11. Gen M, Cheng R (1997) Genetic algorithm and engineering design, 1st edn. Wiley, New York
12. Goyal SK (1988) Joint economic lot size model for purchaser and vendor: a comment. Decis Sci 19:236–241
13. Goyal SK, Gupta YP (1989) Integrated inventory models: the vendor–buyer coordination. Eur J Oper Res 41:261–269
14. Guo YW, Li WD, Mileham AR, Owen GW (2008) Optimization of integrated process planning and scheduling using a particle swarm optimization approach. Int J Prod Res 40:1–22
15. Hadley G, Whitin TM (1963) Analysis of inventory systems. Prentice-Hall, Englewood Cliffs
16. Hariga M, Ben-Daya M (1999) Some stochastic inventory models with deterministic variable lead time. Eur J Oper Res 113:42–51
17. Heydari J, Baradaran-Kazemzadeh R, Chaharsooghi SK (2009) A study of lead time variation impact on supply chain performance. Int J Adv Manuf Technol 40:1206–1215
18. Hill R (1999) The optimal production and shipment policy for the single-vendor single-buyer integrated production–inventory problem. Int J Prod Res 37:2463–2475
19. Hsiao JM, Lin C (2005) A buyer-vendor EOQ model with changeable lead time in supply chain. Int J Adv Manuf Technol 26:917–921
20. Joo SJ, Bong JY (1996) Construction of exact D-optimal designs by Tabu search. Comput Stat Data Anal 21:181–191
21. Kannan G, Noorul-Haq A, Devik M (2009) Analysis of closed loop supply chain using genetic algorithm and particle swarm optimization. Int J of Prod Res 47:1175–1200
22. Kennedy J, Eberhart R (1995) Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 1942–1945
23. Kennedy J, Eberhart R (2001) Swarm intelligence. Academic, San Diego
24. Kim CO, Jun J, Baek JK, Smith RL, Kim YD (2005) Adaptive inventory control models for supply chain management. Int J Adv Manuf Technol 26:1184–1192
25. Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining convergence and diversity in evolutionary multi-objective optimization. Evol Comput 10:263–282
26. Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82:781–798
27. Liu B, Wang L, Jin Y (2008) An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. Comput Oper Res 35:2791–2806
28. Liu B (2004) Uncertainty theory: an introduction to its axiomatic foundations. Springer, Berlin
29. Lodree E, Jang W, Klein CM (2004) Minimizing response time in a two-stage supply chain system with variable lead time and stochastic demand. Int J of Prod Res 42:2263–2278
30. Lu L (1995) A one-vendor multi-buyer integrated inventory model. Eur J Oper Res 81:312–323
31. Naka S, Genji T, Yura T, Fukuyama Y (2001) Practical distribution state estimation using hybrid particle swarm optimization. Proceedings of the IEEE Power Engineering Society Winter Meeting
32. Ouyang LY, Yeh NC, Wu KS (1996) Mixture inventory models with backorders and lost sales for variable lead time. J Oper Res Soc 47:829–832
33. Ouyang LY, Chuang BR (2000) A periodic review inventory model involving variable lead time with a service level constraint. Int J Sys Sci 31:1209–1215
34. Rahimi-Vahed AR, Mirghorbani SM, Rabbani M (2007) A hybrid multi-objective particle swarm algorithm for a mixed-model assembly line sequencing problem. Eng Optim 39:877–898
35. Salman A, Ahmad I, Al-Madani S (2003) Particle swarm optimization for fast assignment problem. Microprocess Microsyst 26:363–371
36. Sha DY, Hsu C (2008) A new particle swarm optimization for the open shop scheduling problem. Comput Oper Res 35:3243–3261
37. Shi Y, Eberhart RC (1999) Empirical study of particle swarm optimization. Proceedings of the 1999 Congress on Evolutionary Computation, 1945–1950
38. Siajadi H, Ibrahim RN, Lochert PB (2006) A single-vendor multiple-buyer inventory model with a multiple-shipment policy. Int J Adv Manuf Technol 27:1030–1037
39. Su S, Zhan D, Xu X (2008) An extended state task network formulation for integrated production-distribution planning in supply chain. Int J Adv Manuf Technol 37:1232–1249
40. Taleizadeh AA, Aryanezhad MB, Niaki STA (2008) Optimizing multi-product multi-constraint inventory control systems with stochastic replenishment. J Appl Sci 8:1228–1234
41. Taleizadeh AA, Niaki STA, Hosseini V (2008) The multi-product multi-constraint newsboy problem with incremental discount and batch order. As J Appl Sci 1:110–122
42. Taleizadeh AA, Moghadasi H, Niaki STA, Eftekhari A (2008) An EOQ-joint replenishment policy to supply expensive imported raw materials with payment in advance. J Appl Sci 8:4263–4273
43. Taleizadeh AA, Niaki STA, Aryanezhad MB (2009) Multi-product multi-constraint inventory control systems with stochastic replenishment and discount under fuzzy purchasing price and holding costs. Am J Appl Sci 6:1–12
44. Taleizadeh AA, Niaki STA, Hosseini V (2009) Optimizing multi-product multi-constraint bi-objective newsboy problem with discount by a hybrid method of goal programming and genetic algorithm. Eng Optim 41:437–457
45. Taleizadeh AA, Niaki STA, Aryanezhad MB (2009) A hybrid method of Pareto, TOPSIS and genetic algorithm to optimize multi-product multi-constraint inventory systems with random fuzzy replenishment. Math Comput Model 49:1044–1057
46. Taleizadeh AA, Niaki STA, Aryanezhad MB (2009d) Multi-product multi-constraint inventory control systems with stochastic

period length and total discount under fuzzy purchasing price and holding costs. Int J Sys Sci (in press)
47. Wang YC (2008) Evaluating flexibility on order quantity and delivery lead time for a supply chain system. Int J Sys Sci 39:1193–1202
48. Wee HM, Yang PC (2007) A mutual beneficial pricing strategy of an integrated vendor-buyers inventory system. Int J Adv Manuf Technol 34:179–187
49. Wu KS (2001) A mixed inventory model with variable lead time and random supplier capacity. Prod Plan Control 12:353–361
50. Yapicioglu H, Smith AE, Dozier G (2007) Solving the semi-desirable facility location problem using bi-objective particle swarm. Eur J Oper Res 177:733–749
51. Zahara E, Hu CH (2008) Solving constrained optimization problems with hybrid particle swarm optimization. Eng Optim 40:1031–1049
52. Zequeira RI, Durn A, Gutirrez G (2005) A mixed inventory model with variable lead time and random back-order rate. Int J Sys Sci 36:329–339