Joint Single Vendor Single Buyer Supply Chain Problem with Stochastic Demand and Fuzzy Lead-Time

Ata Allah Taleizadeh
Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran Email: Ata.taleizadeh@gmail.com

Seyed Taghi Akhavan Niaki
Department of Industrial Engineering, Sharif University of Technology, Tehran Iran niaki@Sharif.edu

Hui-Ming Wee*
Department of Industrial and Systems Engineering, Chung Yuan Christian University, Chungli, Taiwan, Email: weehm@cycu.edu.tw, Tel: 886-3-2654409, Fax: 886-3-2654499

Abstract
In this paper, a chance-constraint joint single vendor, single buyer supply chain problem with stochastic demand and fuzzy lead-time is investigated. The lead-time is lot size dependent; and delay times such as moving, waiting and setting time are fuzzy. The shortage costs result from backorder and/or lost sale, and the uncertain demand follows a uniform distribution. The buyer’s cost function consists of fixed ordering, holding, shortage and transportation costs, while the vendor is responsible for the set up and holding costs. The service rate of each product has a chance constraint and the buyer has a limited budget. The goal is to determine the re-order point and the order quantity of the products such that the total cost is minimized. The mathematical formulation of the problem is shown to be uncertain integer-nonlinear; hence two hybrid procedures of artificial bee colony and particle swarm optimization, with fuzzy simulation and approximate simulation methods are developed to solve the problems. Finally, three numerical examples are given to demonstrate the applicability of the proposed methodologies in a real world supply chain problem.

Keywords: Inventory Control; Supply Chain; Fuzzy Simulation; Artificial Bee Colony; Particle Swarm Optimization

* Corresponding author (HM Wee)
1. Introduction and Literature Review

For an efficient management and coordination of inventories across the entire supply chain, the joint single vendor single buyer (JSVSB) inventory problems have began to receive much attention in the literature recently. Joint single vendor single buyer inventory control problem is one of the most important and famous inventory management model in which the total cost function of both buyer and vendor should be optimized jointly. That means a unique cost function for both players should be derived to minimize the total cost function. The previous researches on the joint vendor-buyer problem assume constant demand and lead time with no constraint. Banerjee (1986) assumed 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’s demand. Other JSVSB inventory models were developed by Goyal (1988) and Yang et al. (2007). Hill (1999) investigated an unequal shipment policy for JSVSB inventory problems with increasing shipment sizes. Ben-Daya and Raouf (1994) and Wu (2001) studied the lead time optimization for JSVSB inventory problems. Ouyang et al. (1996) extended Ben-Daya and Raouf’s (1994) model to allow for partial backordering. Hsiao and Lin (2005) investigated an economic order quantity model using Stackelberg game theory. Hariga and Ben-Daya (1999) developed a continuous review inventory model where the reorder point, the ordering quantity and the lead time were the decision variables. Maity and Maiti (2008) developed an inventory control model for deteriorating items in which fuzzy inflation rate is assumed. Ben-Daya and Hariga (2004) examined the single vendor single buyer integrated production inventory problem. They relaxed the deterministic demand assumption and assume stochastic demand and lead time. Moreover, they also take into consideration nonproductive time. Taleizadeh et al. (2010) extended Ben-Daya and Hariga (2004)'s model to multi chance-constraints multi products joint single vendor single buyer supply chain problem where the demand is assumed to be stochastic and the lead-time is assumed to vary linearly with lot size. An and Zhao (2010) developed a single buyer single vendor system under vendor managed inventory system with fuzzy demand. Mahnam et al. (2009) developed supply chain network model with fuzzy demand and external suppliers. Petrović et al. (1996) presented two models with discrete fuzzy demand and imprecise costs. Li et al. (2002) proposed models with fuzzy demand and deterministic costs. They used fuzzy ordering of fuzzy numbers to obtain the optimal order quantity. Ishill and Konno (1998) assumed that the demand is stochastic and fuzziness was restricted to shortage cost. Kao and Hsu (2002) developed a model where demand was described by a fuzzy number. Shao and Ji (2006) studied a multi-product constrained fuzzy newsboy problem. Panda et al. (2008) extended the single period inventory problem into a multi-product manufacturing system under chance and imprecise constraints. Paksoy and Yapıcı Pehlivan (2012) developed fuzzy linear programming model for the optimization of multi-stage supply chain networks with triangular and trapezoidal membership functions.

In this paper, a chance-constraint multi products joint single vendor single buyer (CCJSVSB) inventory model is developed. The study extends the work by Ben-Daya and Hariga’s (2004) as well as that by Taleizadeh et al. (2010). Our work is different from the preceding study in the following constructs: 1) partial backordering is assumed, 2) service rate is a chance-constraint for each product, 3) multi-product inventory system is assumed, 4) order quantity is placed in multiple packets, 5) lead-time is considered fuzzy, 6) deterministic cost factors are replaced by approximate ones, 7) fuzzy simulation and approximate simulation
methods are considered, and 8) two different hybrid meta-heuristic solution algorithms, Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO), with fuzzy simulation (FS) and approximate simulation (AS) methods are employed to solve the model.

The rest of the paper is organized as follows. In Section 2, some definitions of fuzzy and approximate environment are given. The problem is defined and modeled in Section 3 and 4. In Section 5, the two hybrid meta-heuristic solution algorithms are developed. Three numerical examples, sensitivity analysis and managerial insights of the study are provided in Section 6 and 7. Finally, the concluding remarks are given in Section 8.

2. Definitions of Fuzzy Environments

The credibility, the possibility and the expected value concepts of the fuzzy variables are based on Liu (2002, 2004), Liu and Liu (2002) and Xu and Zha (2008).

Definition 1: A Fuzzy number is of LR-Type, if there exist reference functions L (for the left), R (for the right), and scalars $\alpha > 0, \beta > 0$ with

$$
\mu(\bar{\xi}) = \begin{cases} 
1 & \bar{\xi} \in [m, n] \\
L \left( \frac{m - \bar{\xi}}{\alpha} \right) & \bar{\xi} \leq m \\
R \left( \frac{\bar{\xi} - n}{\beta} \right) & \bar{\xi} \geq n
\end{cases}
$$

(1)

In which $\bar{\xi}$ is abbreviated for $\bar{\xi} = (m, n, \alpha, \beta)_{L-R}$. The triangular and trapezoidal fuzzy variables are specific types of LR-Type.

Definition 2: Let $\bar{\xi}$ be a fuzzy number with the membership function $\mu(\bar{\xi})$. The possibility, necessity, and credibility measure of the fuzzy event $\bar{\xi} \geq r$ can be represented, respectively, by:

$$
\text{Pos} \left\{ \bar{\xi} > r \right\} = \sup_{\bar{\xi} \geq r} \mu(\bar{\xi}) 
$$

(2)

$$
\text{Nec} \left\{ \bar{\xi} \geq r \right\} = 1 - \sup_{\bar{\xi} < r} \mu(\bar{\xi}) 
$$

(3)

$$
\text{Cr} \left\{ \bar{\xi} \geq r \right\} = \frac{1}{2} \left[ \text{Pos} \left\{ \bar{\xi} \geq r \right\} + \text{Nec} \left\{ \bar{\xi} \geq r \right\} \right] 
$$

(4)

Definition 3: The expected value of a fuzzy variable $\bar{\xi}$ is defined as:

$$
E[\bar{\xi}] = \int_{0}^{\infty} \text{Cr} \left\{ \bar{\xi} \geq r \right\} dr - \int_{-\infty}^{0} \text{Cr} \left\{ \bar{\xi} \leq r \right\} dr
$$

(5)

The expected value of a triangular fuzzy variable $\bar{\xi} = (\xi_1, \xi_2, \xi_3)$ is
\[ E[\tilde{\xi}] = \frac{1}{4}(\tilde{\xi}_1 + 2\tilde{\xi}_2 + \tilde{\xi}_3) \]  \hspace{1cm} (6)

**Definition 4:** Let \( \tilde{\xi} \) be a fuzzy variable. Then the optimistic function of \( \alpha \) is defined as:
\[
\tilde{\xi}_{\text{sup}}(\alpha) = \sup \left[ r \left| Cr \{ \tilde{\xi} \geq r \} \geq \alpha \right. \right] , \quad \alpha \in (0,1]
\]  \hspace{1cm} (7)

**Definition 5:** Let \( (C_1, C_2, \ldots, C_k) \) are real constant and \( G_1(\tilde{\xi}), G_2(\tilde{\xi}), \ldots, G_k(\tilde{\xi}) \) are functions of fuzzy variable then,
\[
E \left[ \sum_{k=1}^{K} C_k G_k(\tilde{\xi}) \right] = \sum_{k=1}^{K} C_k E(G_k(\tilde{\xi}))
\]  \hspace{1cm} (8)

The required definitions in approximate environment are presented in (Liu 2004).

**Definition 6:** let \( \Lambda \) be a nonempty set, \( A \) be a \( \sigma \)-Algebra of subset of \( \Lambda \), \( \Delta \) be an element in \( A \), and \( \pi \) be a nonnegative, real valued, additive set function, then \( (\Lambda, \Delta, A, \pi) \) is called approximate space. Further, an approximate variable is defined as a measurable function from an approximate space to the real line.

**Definition 7:** An approximate variable \( \delta \) on the approximate space \( (\Lambda, \Delta, A, \pi) \) is a function from \( \Lambda \) to the real line \( \Re \) such that for every Borel set \( O \) of \( \Re \), we have \( \{ \vartheta \in \Lambda | \delta(\vartheta) \in O \} \in A \). The lower and the upper approximation of the approximate variable \( \delta \) are then respectively defined as follows,
\[
\tilde{\delta} = \{ \delta(\vartheta) | \vartheta \in \Delta \} \quad \text{and} \quad \bar{\delta} = \{ \delta(\vartheta) | \vartheta \in \Lambda \}
\]  \hspace{1cm} (9)

An approximate variable \( ([a,b],[c,d]) \) with \( c \leq a \leq b \leq d \) is a measurable function from an approximate space \( (\Lambda, \Delta, A, \pi) \) to the real line, where \( \Lambda = \{ \chi | c \leq \chi \leq d \} \), \( \Delta = \{ \chi | a \leq \chi \leq b \} \) and \( \delta(\chi) = \chi \) for all \( \chi \in \Lambda \).

**Definition 8:** The expected value of assumed approximate variable will be:
\[
E[\delta] = \frac{1}{4}[a + b + c + d]
\]  \hspace{1cm} (10)

3. Problem Definition and Mathematical Modeling

Similar to Taleizadeh et al. (2010) and Ben-Daya and Hariga (2004), we assume the buyer’s classical \((r, Q)\) continuous review inventory policy considers stochastic demand. The lead time is a function of the production lot size. In particular, we assume the lead time of each product is proportional to the corresponding lot size produced by the vendor plus a fuzzy delay due to transportation and nonproductive time.
The relationship between the vendor and the buyer is described as follows: for product \( i, i = 1, 2, \ldots, p \) the buyer orders a lot of size \( Q_i \) to the vendor and incurs an ordering cost. The vendor manufactures the product in lots with a finite production rate and incurs a setup cost. Then, the buyer receives \( n_i \) shipments, each containing \( M_i \) packets of \( n_i \) products, \( Q_i = n_i M_i \) and incurs an approximate transportation cost with each shipment. The buyer places his order when his on-hand inventory of product \( i \) reaches to reorder point \( r_i \). Moreover, there is a service-rate lower limit for each product, and that the buyer has a limited budget. Partial backordering is considered. The objective of this study is to determine the reorder point and order quantity of each product such that the expected total cost of the supply chain is minimized.

The following parameters and the variables are used in model development.

3.1. Parameter and Variables

For \( i = 1, 2, \ldots, p \), the parameters and the variables are defined as:
- \( r_i \): The re-order point of product \( i \) (a decision variable).
- \( n_i^* \): Number of shipments of product \( i \) from the vendor to the buyer (a decision variable).
- \( M_i \): The expected number of packets for product \( i \) order (a decision variable).
- \( n_i \): The number of product \( i \) in each packet.
- \( Q_i \): The expected amount of product \( i \) order (a decision variable in which, \( Q_i = n_i M_i \)).
- \( SS_i \): The safety stock of product \( i \) (a decision variable).
- \( D_i \): The expected demand quantity of product \( i \), i.e., \( D_i = \frac{D_i^{\text{Min}} + D_i^{\text{Max}}}{2} \).
- \( f_{D_i}(d_i) \): The probability density functions of \( D_i \) (a Uniform density function with parameters \( D_i^{\text{Min}}, D_i^{\text{Max}} \)).
- \( P_i \): The constant production rate of product \( i \) (\( P_i \geq D_i \)).
- \( SL_i \): The lower limit of the service level for product \( i \).
- \( \beta_i \): The percentage of unsatisfied demands of product \( i \) that is backordered.
- \( \tilde{\pi}_i \): The approximate backorder cost per unit demand of product \( i \).
- \( \tilde{\xi}_i \): The approximate shortage cost for each unit of product \( i \) that is lost sale.
- \( A_i \): The constant cost of each order.
- \( \tilde{A}_i \): The approximate buyer's transportation cost per shipment of product \( i \).
- \( A_i^p \): The buyer's constant production cost of each setup.
- \( \tilde{h}_i^v \): The approximate vendor's holding cost per unit per unit time of product \( i \).
- \( \tilde{h}_i^b \): The approximate buyer's holding cost per unit per unit time of product \( i \).
- \( \tilde{h}_i(r,Q) \): The expected amount of product \( i \) shortage.
\( B_i \): The expected amount of product \( i \) back-ordered (\( B_i = \beta_i \hat{B}(r, Q) \)).

\( L_i \): The expected amount of product \( i \) lost sale (\( L_i = (1 - \beta_i) \tilde{L}(r, Q) \)).

\( I_i \): The expected amount of product \( i \) inventory.

\( LT_i \): Product \( i \) lead time is assumed to be \( LT_i = \frac{Q}{P_i} + \hat{\gamma}_i \), where \( \hat{\gamma}_i \) denotes a fuzzy delay due to transportation and production time of other products scheduled during the lead time on the same facility.

\( C_{H_b} \): The expected total holding cost of the buyer.

\( C_{H_v} \): The total holding cost of the vendor.

\( C_{B_b} \): The buyer's expected total shortage cost in back-ordered state.

\( C_{B_L} \): The buyer's expected total shortage cost in lost sale state.

\( C_{A_b} \): The buyer's expected total constant order cost.

\( C_{A_v} \): The vendor's expected total constant set up cost.

\( C_{T_b} \): The buyer's expected total transportation cost.

\( TB \): Buyer's total available budget

\( C_i \): Purchasing price per unit of product \( i \)

\( TC(b) \): The buyer's expected total cost.

\( TC(v) \): The vendor's expected total cost.

\( TC \): The expected total cost of the supply chain.

### 3.2. The Buyer's Total Costs

We start by developing a model for a single product, and then we extend it to consider multiple products.

For a single product, the buyer's total expected cost per unit time is given by:

\[
TC(b) = C_{A_b} + C_{H_b} + C_{B_b} + C_{L_b} + C_{T_b}
\]  

(11)

From figure (1), \( C_{A_b} \) and \( C_{T_b} \) are derived from Ben-Daya and Hariga (2004):

\[
C_{A_b} = \frac{AD_i}{n_i Q_i}
\]

(12)

\[
C_{T_b} = \frac{D_i}{Q_i} A_i^i
\]

(13)

Insert Figure (1) about here
Since the demand is stochastic, the expected on hand inventory per unit time (Hadley-Whitin, 1963) is:

$$I_i = \frac{Q_i}{2} + SS_i$$  \hspace{1cm} (14)

Where $SS_i$ (from Taleizadeh et al. 2010) is:

$$SS_i = r_i - D_i \left( \frac{Q_i}{P_i} + \hat{\gamma}_i \right) + \frac{(1 - \beta_i)(D_i^{Max} - r_i)^2}{2(D_i^{Max} - D_i^{Min})}$$  \hspace{1cm} (15)

Because $SS_i^{Back Ordered} = r_i - D_i(LT_i)$ and $SS_i^{Lost Sale} = r_i - D_i(LT_i) + \overline{b_i}(r_i, Q_i)$ sum up the total shortage (backordered plus lost sale), one has:

$$SS_i^{Combination} = \beta_i SS_i^{Back Ordered} + (1 - \beta_i) SS_i^{Lost Sale} = r_i - D_i(LT_i) + (1 - \beta_i) \overline{b_i}(r_i, Q_i)$$  \hspace{1cm} (16)

Moreover, since the demand distribution function is a continuous uniform distribution, and shortage occurs when the demand is more than the reorder point, the expected shortage is:

$$\bar{b}_i(r_i, Q_i) = \int_{r_i}^{D_{i}^{Max}} \frac{1}{D_{i}^{Max} - D_{i}^{Min}} dx_i = \frac{(X_i - r_i)^2}{2(D_{i}^{Max} - D_{i}^{Min})} \bigg|_{r_i}^{D_{i}^{Max}} = \frac{(D_{i}^{Max} - r_i)^2}{2(D_{i}^{Max} - D_{i}^{Min})}$$  \hspace{1cm} (17)

Hence, knowing $LT_i = (\frac{Q_i}{P_i} + \hat{\gamma}_i)$, one has

$$C_{H_b} = \overline{b}_i \left( \frac{Q_i}{2} + r_i - D_i \left( \frac{Q_i}{P_i} + \hat{\gamma}_i \right) \right) + \frac{(1 - \beta_i)(D_i^{Max} - r_i)^2}{2(D_i^{Max} - D_i^{Min})}$$  \hspace{1cm} (18)

The backorder and lost sale costs, $B_i = \beta_i \frac{D_i}{n_i^o Q_i} \overline{b}_i(r_i, Q_i)$ and $L_i = (1 - \beta_i) \frac{D_i}{n_i^o Q_i} \overline{b}_i(r_i, Q_i)$, using equation (17) is:

$$C_{B_b} = \frac{\overline{b}_i \beta_i D_i (D_i^{Max} - r_i)^2}{2n_i^o Q_i (D_i^{Max} - D_i^{Min})}$$  \hspace{1cm} (19)

$$C_{L_b} = \frac{\overline{b}_i (1 - \beta_i) D_i (D_i^{Max} - r_i)^2}{2n_i^o Q_i (D_i^{Max} - D_i^{Min})}$$  \hspace{1cm} (20)

The buyer's total expected total cost per unit time for product $i$ is:
3.3. The Vendor's Total Costs

Similarly, the vendor's backorder and lost sale costs. $C_{A_v}$ and $C_{H_v}$, and the total cost for product $i$ are respectively as follows:

$$C_{A_v} = \frac{A^v D_i}{n_i^o Q_i}$$

$$C_{H_v} = \frac{\hat{h}_i^o}{2} \left[ n_i^o \left( 1 - \frac{D_i}{P_i} \right) - 1 + 2 \frac{D_i}{P_i} \right]$$

$$TC(v) = C_{A_v} + C_{H_v} = \frac{A^v D_i}{n_i^o Q_i} + \hat{h}_i^o \left[ n_i^o \left( 1 - \frac{D_i}{P_i} \right) - 1 + 2 \frac{D_i}{P_i} \right]$$

3.4. The Total Supply Chain Cost

From equations (18) and (21), the total supply chain expected cost for product $i$ is:

$$TC = TC(b) + TC(v) = C_{A_v} + C_{H_v} + C_{A_k} + C_{H_k} + C_{L_v} + C_{L_k} + C_{A_v} + C_{A_k} + C_{H_v} + C_{H_k}$$

$$= \frac{A^v D_i}{n_i^o Q_i} + \hat{h}_i^o \left[ \frac{Q_i}{2} + r_i - D_i \left( \frac{Q_i}{P_i} + \hat{\gamma}_i \right) + \frac{(1 - \beta_i) \left( D_i^{Max} - r_i \right)^2}{2 \left( D_i^{Max} - D_i^{Min} \right)} \right] + \frac{\hat{h}_i^o}{2n_i^o Q_i} \left( D_i^{Max} - D_i^{Min} \right)^2$$

$$+ \frac{2 \left( D_i^{Max} - D_i^{Min} \right)^2}{2n_i^o Q_i} \left[ n_i^o \left( 1 - \frac{D_i}{P_i} \right) - 1 + 2 \frac{D_i}{P_i} \right]$$

$$+ \frac{2 \left( D_i^{Max} - D_i^{Min} \right)^2}{2n_i^o Q_i} \left[ n_i^o \left( 1 - \frac{D_i}{P_i} \right) - 1 + 2 \frac{D_i}{P_i} \right]$$

3.5. The Multi-Constraint Multi-Product Supply Chain Model

Based on previous equations, the multi-product multi-constraint supply chain model can be derived as:
\[
\begin{align*}
\text{Min } TC &= \sum_{i=1}^{p} \left( \frac{A_i + A_i^p}{n_i^* Q_i} \right) D_i + \sum_{i=1}^{p} \bar{r}_i^* \left[ \frac{Q_i}{2} + r_i - D_i \left( \frac{Q_i}{P_i} + \hat{\gamma}_i \right) \right] + \frac{(1 - \beta_i)(D_i^{Max} - r_i)^2}{2(D_i^{Max} - D_i^{Min})} \\
&\quad + \sum_{i=1}^{p} \left[ \bar{\bar{r}}_i^* \beta_i + \bar{\bar{r}}_i^* (1 - \beta_i) \right] D_i \left( \frac{D_i^{Max} - r_i}{Q_i (D_i^{Max} - D_i^{Min})} \right) + \sum_{i=1}^{p} \frac{Q_i}{A_i^p} D_i + \sum_{i=1}^{p} \bar{r}_i^* \frac{Q_i}{2} \left[ n_i^* \left( 1 - \frac{D_i}{P_i} \right) - 1 + \frac{2D_i}{P_i} \right] \\
\text{s.t.:} &
\sum_{i=1}^{p} C_i Q_i \leq TB\\
Q_i &= n_i M_i \quad \forall i ; i = 1, 2, \ldots, p\\
r_i \geq \left( \frac{Q_i}{P_i} + \hat{\gamma}_i \right) \left[ (D_i^{Max} - D_i^{Min}) SL_i + D_i^{Min} \right] \quad \forall i ; i = 1, 2, \ldots, p\\
Q_i, n_i^*, r_i, M_i &\geq 0 \text{ Integer} \quad \forall i ; i = 1, 2, \ldots, p
\end{align*}
\]
Upon arrival, the foraging bee takes a load of nectar and returns to the hive relinquishing the nectar. After she relinquishes the food, the bee can: (a) abandon the food source and become again uncommitted follower, (b) continue to forage at the food source without recruiting the nest mates, or (c) dance and thus recruit the nest mates before the return to the food source. The bee opts for one of the above alternatives with a certain probability. Within the dance area, the bee dancers “advertise” different food areas. The mechanisms by which the bee decides to follow a specific dancer are not well understood, but it is considered that “the recruitment among bees is always a function of the quality of the food source (Teodorovic 2009, Karaboga and Basturk 2008).

4.1.1. Real Bees Behavior

In searching for food sources, a colony of honeybees can extend itself over space and direction simultaneously (Von Frisch 1976, Seeley 1996 and Pham et al. 2006). A colony prospers by deploying its foragers to good fields. In principle, flower patches with much nectar or pollen would be visited by more bees (Bonabeau et al. 1999 and Camazine et al. 2003). The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees move randomly from one patch to another.

During the harvesting season, a percentage of the population in a colony continues its exploration (Pham et al. 2006). When they return to the hive, those scout bees that found quality threshold deposit their nectar and perform a dance known as the “waggle dance” (Von Frisch 1976 and Pham et al. 2006). This mysterious dance is essential for colony communication, and contains three pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive and its quality rating (or fitness) (Von Frisch 1976, Pham et al. 2006 and Camazine et al. 2003). This information directs bees to flower patches precisely. The dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it (Pham et al. 2006). After waggle dancing, the scout bee goes back to the flower patch with follower bees that were waiting inside the hive. This allows the colony to gather food quickly and efficiently. While harvesting from a patch, the bees monitor its food level. If the patch is still good enough as a food source, it will be advertised by the waggle dance, and more bees will be recruited to that source (Teodorovic 2009 and Singh 2009).

4.1.2. The Proposed ABC Algorithm

ABC is a swarm intelligent optimization algorithm inspired by honeybee foraging. Employee bees, onlookers and scouts are three groups of the artificial bee colony. A food source position is a possible optimal solution and the nectar amount corresponds to the objective values of the associated solution (Kang et al. 2009).

First, the algorithm initializes total population size of $N = N_{eb} + N_{ob}$ where the initial positions of food sources $N_{eb}$ is equal to the number of employee bees and $N_{ob}$ is the number of onlooker bees. Each solution vector $Y_j \ (j = 1, 2, \cdots, N)$ is a $p$-dimensional vector in which $i = 1, 2, \cdots, p$ is the number of variables in the optimization problem. Each onlooker bee will choose a food source
depending on the probability value $P_j$ associated with the food source, where (Karaboga and Basturk 2008 and Singh 2009):

$$P_j = \frac{TC_j}{\sum_{k=1}^{N} TC_k}$$

(27)

in which $TC_j$ is the objective function of the $j^{th}$ solution. A candidate solution $X_j$ from the neighborhood of old solution $Y_j$ will be generated as (Singh 2009),

$$X_{ji} = Y_{ji} + \text{Rand} \cdot (1,1)(Y_{ji} - Y_{ki})$$

(28)

Where $Y_j$ and $Y_k$ are randomly chosen and $k$ is not equal to $j$. After each generation, the artificial bee compares and evaluates the food source position and quality. If the new food source has a better value than the previous source, the previous one is replaced by the new one. Otherwise, the old one is retained. If a position cannot be improved further through a predetermined number $N_{ABC}$ (limited cycles), then that food source is assumed to be abandoned and the corresponding employed bee becomes a scout. Finally, the abandoned position will be replaced by a new food source found by the scout. Assuming the abandoned source is $Y_j$, the new food source discovered by the scout is described as (Karaboga and Basturk 2008),

$$Y_j = Y_{\text{min}} + \text{Rand} \cdot (0,1)(Y_{\text{max}} - Y_{\text{min}})$$

(29)

It should be noted that the stopping criteria in this method is reaching to maximum iteration. The parameter values of the ABC algorithm is shown in Table (8), where it contains the parameter values of the second hybrid algorithm described in Section 5.2.

In short, the steps involved in the ABC algorithm are shown as follows:

1. Generate $N$ food sources randomly; based on the objective function, evaluate them.
2. Select $N_{eb}$, and assign them to each employee bee until a stopping criteria is met.
3. Determine the abandoned food sources and move scouts to a new search by equation (29).
4. Each employee bee will find a candidate food source according to equation (28).
5. Evaluate the candidate food source and replace the new better food source.
6. Each onlooker bee selects a food source according to equation (27).
7. Each onlooker bee generates a candidate solution according to equation (28).
8. Evaluate the candidate food source and select a better new food source.
9. Store the best-known food source found.

4.2. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was invented by Kennedy and Eberhart (1995) 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 (Taleizadeh et al. 2010). PSO updates a population (called swarm) of individuals (called particles). A particle is treated as a point in a multidimensional space and the status of the particle is characterized by its position and velocity in the search space. The position of a particle represents a candidate solution to the problem at hand. PSO basically relies on the exchange of information between particles of the swarm [Kennedy and Eberhart 2001, Dye and Hsieh 2010]. Each particle flies over to an optimal position according to its own trajectory (Taleizadeh et al. 2010). For each iteration, it adjusts its trajectory towards its own previous best position (local best) and towards the current best experience or position found by all particles in its neighborhood (global best). Thus global information sharing takes place and particles benefit from their own discoveries (i.e., local bests) and the previous experience of all other members (i.e., global bests) during the search process. Search process continues until a relatively steady position or the iteration limit is reached (Taleizadeh et al. 2010). The local bests and global bests are determined by evaluating the fitness or objectives of the current particles at each iteration. The basic PSO algorithm includes three phases: particle’s positions and velocities generations, velocity update, and position update, which are defined in following subsections. The definition of the parameters of PSO algorithm is shown in Table (2).

Insert Table (2) about here

Equations (30) and (31) are used to initialize particles, in which $\Delta t$ is the constant time increment.

\[
X_0^i = X_{\text{min}} + \text{Rand} \left( X_{\text{max}} - X_{\text{min}} \right) \quad (30)
\]

\[
V_0^i = \frac{X_{\text{min}} + \text{Rand} \left( X_{\text{max}} - X_{\text{min}} \right)}{\Delta t} = \frac{\text{Position}}{\text{time}} \quad (31)
\]

In order to update the velocities, the following equation is being used,

\[
V_{k+1}^i = wV_k^i + C_1 R a n d \frac{(P_i^k - X_i^k)}{\Delta t} + C_2 R a n d \frac{(P^g_k - X_i^k)}{\Delta t} \quad (32)
\]

The constraint in equation (33) is specified to clamp the excessive accelerations.

\[
\text{if} \quad (V_{k+1}^i > V_{\text{max}}) \quad ; \quad V_{k+1}^i = V_{\text{max}}
\]

\[
\text{if} \quad (V_{k+1}^i < -V_{\text{max}}) \quad ; \quad V_{k+1}^i = -V_{\text{max}} \quad (33)
\]

The inertia weight $w$ controls how much of the previous velocity should be retained from the previous step. Introducing a linearly decreasing inertia weight into the original PSO significantly improves its performance through the parameter study of inertia weight (Taleizadeh
et al. 2010). The linear distribution of the inertia weight is expressed as follows (Taleizadeh et al. 2010, Dye and Hsieh 2010 and Naka et al. 2001):

\begin{equation}
W = W_{\text{max}} - \left( \frac{W_{\text{max}} - W_{\text{min}}}{k_{\text{max}}} \right) k
\end{equation}

Finally, position update is the last step involved in the iteration and is done by using the current particle position and its own updated velocity vector shown in the Equation (35).

\begin{equation}
X_{K+1}^i = X_K^i + V_{K+1}^i \Delta t
\end{equation}

In short, the steps involved in the PSO algorithm are shown as follows:

Repeat:
   - For each particle
     1. Update the velocities.
     2. Update the positions.
     3. Evaluate the fitness according to the desired optimization model.
     4. Update the best global value and position of each particle over time if necessary.
End: when a maximum number of iterations is reached.

### 4.3. Fuzzy Simulation (FS)

In order to estimate the uncertain lead-time of the fuzzy model, we employ a fuzzy simulation technique. Denoting \( \hat{\gamma} \) by \( \hat{\gamma} = (\hat{\gamma}_1, \hat{\gamma}_2, \cdots, \hat{\gamma}_p) \), \( \mu \) as the membership function of \( \hat{\gamma} \), and \( \mu_i \) are the membership functions of \( \hat{\gamma}_i \), we randomly generate \( \gamma_{\alpha} \) from the \( \alpha \)-level sets of fuzzy variables \( \hat{\gamma}_i \), \( i = 1, 2, \cdots, p \), \( k = 1, 2, \cdots, K \) and \( \gamma_i = (\gamma_{\alpha_1}, \cdots, \gamma_{\alpha_p}) \) and \( \mu(\hat{\gamma}_i) = \mu_1(\gamma_{\alpha_1}) \land \mu_2(\gamma_{\alpha_2}) \land \cdots \land \mu_p(\gamma_{\alpha_p}) \), where \( \alpha \) is a sufficiently small positive number. Based on the definition in equation (3), the expected value of the fuzzy variable is (Liu 2004):

\begin{equation}
E[\hat{\gamma}_i] = \int_{r_i}^{\infty} Cr\{\hat{\gamma}_i \geq r_i\} dr - \int_{-\infty}^{r_i} Cr\{\hat{\gamma}_i \leq r_i\} dr
\end{equation}

Provided \( N \) is sufficiently large, for any number \( r_i \geq 0 \), \( Cr\{\hat{\gamma}_i \geq r_i\} \) can be estimated by:

\begin{equation}
Cr\{\hat{\gamma}_i \geq r_i\} = \frac{1}{2} \left( \text{Max}_{k=1,2,\cdots,N} \{\mu_{\alpha_k} \gamma_i \geq r_i\} + 1 - \text{Max}_{k=1,2,\cdots,N} \{\mu_{\alpha_k} \gamma_i < r_i\} \right)
\end{equation}

And for any number \( r_i < 0 \), \( Cr\{\hat{\gamma}_i \leq r_i\} \) can be estimated by:
\[ Cr \{ \hat{y}_i \leq r_i \} = \frac{1}{2} \left\{ \max_{k=1,2,...,N} \left\{ \mu_{ik} \mid \hat{y}_i \leq r_i \right\} + 1 - \max_{k=1,2,...,N} \left\{ \mu_{ik} \mid \hat{y}_i > r_i \right\} \right\} \] (38)

4.4. Approximate Simulation (AS)

In order to estimate the uncertain cost factors defined in approximate variables, we employ an approximate simulation technique. Approximate simulation plays an important role in approximate system. If \( \hat{h}^b, \hat{h}^v, \hat{\pi}^b, \hat{\pi}^v \) are approximate vectors defined on approximate spaces \( (\Lambda^b, \Delta^b, A^b, \pi^b) \), \( (\Lambda^v, \Delta^v, A^v, \pi^v) \), \( (\Lambda^\delta, \Delta^\delta, A^\delta, \pi^\delta) \), and \( (\Lambda^\pi, \Delta^\pi, A^\pi, \pi^\pi) \), respectively, in order to estimate their expected values, approximate simulation approach may be employed. The other approximate variables estimations are made in a similar manner as \( \hat{\pi}_i \) estimation. In this approach, by denoting \( \pi_i = (\pi_{i1}, \pi_{i2}, \ldots, \pi_{ik}) \), we generate \( \hat{\pi}_i = (\hat{\pi}_{i1}, \hat{\pi}_{i2}, \ldots, \hat{\pi}_{ik}) \) from \( \Lambda^\delta \), and \( \hat{\pi}_i = (\hat{\pi}_{i1}, \hat{\pi}_{i2}, \ldots, \hat{\pi}_{ik}) \) from \( \Lambda^{\pi} \) according to the measure \( \pi^\pi \). Thus \( \hat{\pi}_i \) is a function of \( \pi_i \) and \( \hat{\pi}_i \).

Figure (2) shows a flowchart of either the proposed hybrid method of BCO, FS and AS method or the proposed hybrid method of PSO, FS and AS, in which \( N_{AS} \) and \( N_{FS} \) indicate the number of iterations of approximate and fuzzy simulations, respectively.

5. Numerical Examples and Sensitivity Analysis

Consider three multiproduct problems with different quantity of products. The first one has 10 products and its general data is given in Tables (3) and (4) and with \( TB = 500,000 \). The second example includes 25 products with \( TB = 1,000,000 \), and the third one includes 50 products with \( TB = 2,000,000 \). Tables (5) and (6) show the best results obtained from the first numerical example with detailed decision variables information. Table (7) shows the results obtained from the second and the third examples in which the total cost and related costs of the single buyer and the single vendor are shown separately. Table (8) shows the sensitivity of varying the ABC and PSO parameters values on the optimal solutions. In this research, all of the possible combinations of the ABC and PSO parameters given in Table (8) are employed and the best results are obtained using the \textit{min (min)} criterion. The best combination of the ABC parameters for all examples is
obtained as $N_{eb} = N_{ob} = 50, N_{ABC} = 100$; and the best combination of the PSO parameters is obtained as $C_1 = C_2 = 2, N_{PSO} = 100$. In addition, we considered $N_{AS} = N_{FS} = 20$.

In the case of CPU time, there is no significant difference between the hybrid ABC and the hybrid PSO. To show this, the best, the mean, and the variances of CPU time for all three examples based on solution methods are provided in Table (9).

To study the effects of parameter changes on the best result derived by the proposed methods and on the required CPU time to obtain it, a sensitivity analysis is performed to investigate the effect of increasing or decreasing the parameters.

The parameters investigated are demand and production rates. In the ABC algorithm; the number of employee bees, $N_{eb}$, the number of onlooker bees, $N_{ob}$, and the number of predefined iterations, $N_{ABC}$, are investigated as well. For the parameters corresponding to the simulations parts; $N_{AS}$ and $N_{FS}$ have been analyzed and the effect of number of product, $N_p$ on CPU time is investigated.

Table (10) shows the effects of the model parameter changes on the objective function value and Table (11) shows the effects of the parameters changes of the ABC algorithm on CPU time.

Based on the results given in Table (10), the total cost of the buyer and the vendor is highly sensitive to the demand rate, while they are slightly sensitive with respect to the changes in value of the production rate. When the value of demand rate is increased by 50%, there is no feasible solution because the demand rate must be less than the production rate. Further, when the production rate is decreased by 50%, it becomes smaller than the demand rate and there is no feasible solution as well.

Based on the results in Table (11), the CPU time of the hybrid ABC-FS-AS algorithm is highly sensitive to the iteration number of the approximate simulation part of the algorithm and on the number of employee and onlooker bees. Furthermore, the CPU time is moderately sensitive with respect to the changes in the values of iteration number of the ABC algorithm and the fuzzy simulation part of the hybrid algorithm.
All runs are performed using MATLAB on a pentium5 computer with Core 2 Due 2.93 GHZ processor.

6. Conclusion and Recommendations for Future Research

In this paper, a chance constraint single buyer single vendor inventory problem was investigated. A mathematical model considering deterministic, approximate and fuzzy variables is developed. Since the model is an uncertain integer-nonlinear problem, the hybrid of ABC and PSO with Fuzzy Simulation technique and approximate simulation are proposed to solve the model. Three practical numerical examples are given to demonstrate the feasibility of the proposed methodologies. Further research can be done to consider developing the model and algorithms for the multi objective problems.

7. References


