Modeling fuzzy multi-period production planning and sourcing problem with credibility service levels

Article *in* Journal of Computational and Applied Mathematics · September 2009

DOI: 10.1016/j.cam.2009.02.009 · Source: DBLP

Yanfei Lan
Tianjin University

Gao-Ji Sun

Journal of Computational and Applied Mathematics 231 (2009) 208–221

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

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

Modeling fuzzy multi-period production planning and sourcing problem with credibility service levels

Yan-Fei Lan, Yan-Kui Liu *, Gao-Ji Sun

College of Mathematics & Computer Science, Hebei University, Baoding 071002, Hebei, China

ARTICLE INFO

Article history:
Received 11 June 2008
Received in revised form 17 December 2008

Keywords:
Fuzzy programming
Production planning
Credibility service level
Approximation approach
Particle swarm optimization

ABSTRACT

A great deal of research has been done on production planning and sourcing problems, most of which concern deterministic or stochastic demand and cost situations and single period systems. In this paper, we consider a new class of multi-period production planning and sourcing problem with credibility service levels, in which a manufacturer has a number of plants and subcontractors and has to meet the product demand according to the credibility service levels set by its customers. In the proposed problem, demands and costs are uncertain and assumed to be fuzzy variables with known possibility distributions. The objective of the problem is to minimize the total expected cost, including the expected value of the sum of the inventory holding and production cost in the planning horizon. Because the proposed problem is too complex to apply conventional optimization algorithms, we suggest an approximation approach (AA) to evaluate the objective function. After that, two algorithms are designed to solve the proposed production planning problem. The first is a PSO algorithm combining the AA, and the second is a hybrid PSO algorithm integrating the AA, neural network (NN) and PSO. Finally, one numerical example is provided to compare the effectiveness of the proposed two algorithms.

1. Introduction

Production planning is viewed as the plans and arrangements of the production mission and progress in production scheduled time. In recent years, production planning – especially uncertain production planning – has been studied widely in the field of production planning management. Galbraith [1] defined uncertainty as the difference between the amount of information required to perform a task and the amount of information already possessed. In the real world, there are many forms of uncertainty that affect production processes. Ho [2] categorized them into two groups: (i) environmental uncertainty and (ii) system uncertainty. Environmental uncertainty included uncertainties beyond the production process such as demand uncertainty and supply uncertainty. System uncertainty was related to uncertainties within the production process such as operation yield uncertainty, production lead time uncertainty, quality uncertainty, failure of production system and change to product structure. Uncertainty can be present as randomness and fuzziness in the production environment. This uncertainty will result in more realistic production planning models. However, the inclusion of uncertainty in the production system parameters is a more difficult task in terms of modeling and solving. Over the years, there has been much research and many applications with the aim of modeling the uncertainty in production planning problems, such as the material requirements planning (MRP) model [3,4], the hierarchical production planning (HPP) model [5,6], the aggregate production planning (APP) model [7–9], the supply chain (SC) model [10,11] and other well-known production planning models in the literature [12–15].

* Corresponding author.
E-mail addresses: yanfei-lan@163.com (Y.-F. Lan), liuyankui@tsinghua.org.cn (Y.-K. Liu), gsunmath@126.com (G.-J. Sun).

0377-0427/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2009.02.009

In order to handle probabilistic uncertainty in the production decision systems, some meaningful stochastic production planning models have been proposed in the literature such as [12]. They dealt with a stochastic production planning problem with service level requirements, and provided non-sequential and deterministic equivalent formulations of the model. Kelly, Clendenen and Dardeau [16] extended the economic lot scheduling problem for the single-machine multi-product case with random demands. Their objective was to find the optimal length of production cycles that minimizes the sum of set-up costs and inventory holding costs per unit of time, and satisfy the demand of products at the required service levels. Zäpfel [14] claimed that MRP II systems could be inadequate for the solution of production planning problems with uncertain demand because of the insufficiently supported aggregation/disaggregation process. The paper then proposed a procedure to generate an aggregate plan and a consistent disaggregate plan for the Master Production Schedule.

In fuzzy decision systems, fuzzy production planning models have been considered by many researchers. Based on fuzzy set theory and possibility theory [17–20], many researchers applied them to fuzzy optimization models such as [7,9,21,22]. Among them, Wang and Fang [7] presented a fuzzy linear programming model for solving the aggregate production planning problem with multiple objectives. Gen and Tsujimura [9] proposed a fuzzy model with multiple objectives for aggregate planning, with objective function coefficients, technological coefficients, and resource right-hand side constraints represented by triangular fuzzy numbers. Tanaka et al. [21] transformed possibilistic linear programming problems based on exponential possibility distributions into non-linear optimization problems. In order to solve optimization problems easily, algorithms for obtaining center vectors and distribution matrices in sequence were proposed. Shih [22] resolved the cement transportation planning problem using fuzzy linear programming methods. Three types of fuzzy linear programming models were used to determine the optimal transportation amount and the capacity of new facilities.

The purpose of this paper is to present a realistic production planning model. We take credibility theory [23–25] as the theoretical foundation of fuzzy optimization and develop a multi-period production planning and sourcing problem with a credibility service levels model, in which demands and costs are uncertain and assumed to be fuzzy variables with known possibility distributions. The objective of the problem is to minimize the total expected cost, including the expected value of the sum of the inventory holding and production cost in the planning horizon. Also, we transform the credibility constraint into its crisp equivalent form when demands are independent normal fuzzy variables. Then, we suggest an AA to evaluate the objective function. Since the approximating production planning problem is neither linear nor convex, conventional optimization algorithms cannot be applied. Therefore, two approximation-based algorithms are designed to solve the proposed production planning problem. The first is the PSO algorithm which integrates the AA [26] and PSO [27,28], and the second is the hybrid PSO algorithm which combines the AA, neural network (NN) and PSO. One numerical example is also provided to compare the effectiveness of the two algorithms.

The rest of this paper is organized as follows. In Section 2, we recall some preliminary knowledge. Section 3 proposes a new class of fuzzy production planning model. In Section 4, we employ the AA to discretize the objective function of the fuzzy production planning model, and deal with the convergence of the AA. The convergent result allows us to design two approximation-based PSO algorithms to solve the proposed fuzzy production planning problem in Section 5, and one numerical example is provided in this section to compare the effectiveness of the two algorithms. Section 6 summarizes the main results in this paper.

2. Preliminaries

Given a universe Γ, $\mathcal{P}(\Gamma)$ is the power set of Γ and Pos is a set function defined on $\mathcal{P}(\Gamma)$. Let ξ be a fuzzy variable with membership function $\mu(x)$ and r a real number. Then the possibility measure of a fuzzy event $\{\xi \le r\}$ is defined as

$$\mathrm{Pos}\{\xi \le r\} = \sup_{x\le r} \mu(x)$$

for any real number r.

The credibility measure [23] of the fuzzy event $\{\xi \le r\}$ was defined as

$$\mathrm{Cr}\{\xi \le r\} = \frac{1}{2}\left(1 + \sup_{x\le r}\mu(x) - \sup_{x>r}\mu(x)\right)$$

for any real number r.

Using credibility measure, the expected value of the fuzzy variable ξ, denoted by $E[\xi]$, was defined as

$$E[\xi] = \int_0^\infty \mathrm{Cr}\{\xi \ge r\}\mathrm{d}r - \int_{-\infty}^0 \mathrm{Cr}\{\xi \le r\}\mathrm{d}r$$

provided that at least one of the two integrals is finite.

In particular, if ξ is a finite discrete fuzzy variable with the following membership function

$$\mu_\xi(x) = \begin{cases} \mu_1, & \text{if } x = \widehat{\xi}_1 \\ \mu_2, & \text{if } x = \widehat{\xi}_2 \\ \cdots & \\ \mu_n, & \text{if } x = \widehat{\xi}_n \end{cases}$$

such that $\mu_i = \text{Pos}\left\{\xi = \widehat{\xi}_i\right\} > 0$, and $\max_{i=1}^n \mu_i = 1$. Assume that $\widehat{\xi}_i$, $i = 1, 2, \ldots, n$ satisfy the condition $\widehat{\xi}_1 \le \widehat{\xi}_2 \le \cdots \le \widehat{\xi}_n$, then the expected value becomes

$$E[\xi] = \sum_{i=1}^{n} w_i \widehat{\xi}_i,$$

where the weights w_i, $i = 1, 2, \ldots, n$ are given by

$$w_i = \frac{1}{2}\left(\max_{j=1}^{i} \mu_j - \max_{j=0}^{i-1} \mu_j\right) + \frac{1}{2}\left(\max_{j=i}^{n} \mu_j - \max_{j=i+1}^{n+1} \mu_j\right)$$

with $\mu_0 = \mu_{n+1} = 0$. It is easy to verify that all $w_i \ge 0$, and $\sum_{i=1}^n w_i = \max_{i=1}^n \mu_i = 1$.

3. Problem formulation

In this section, we will construct a new type of fuzzy programming model of a multi-period production planning and sourcing problem with fuzzy parameters. The characteristic of this manufacturing system can be summarized as follows.

• There are N types of production sources (plants and subcontractors) in the system, and the decision of production levels to meet market demand with the minimum cost must be taken for T periods. The demand in each period is uncertain and is characterized by a fuzzy variable with known possibility distribution.

• The costs that are used in the model's objective function consist of production cost and inventory carrying cost. The production and inventory cost coefficients are not known exactly and assumed to be represented by fuzzy variables. In general, we assume fuzzy demands and production and inventory cost coefficients in different periods are mutually independent [29].

• Constraints on the performance (related to backorders) of the system are imposed by requiring service levels which force the credibility of having no stock out to be greater than or equal to a predetermined service level requirement in each period.

The following indices and parameters are used to describe the model.

Indices:
i: index of sources, $i = 1, 2, \ldots N$;
t: index of periods, $t = 1, 2, \ldots T$.
Parameters:
$\tau_{i,t}$: the fuzzy unit production cost at source i in period t;
η_t: the fuzzy unit cost of inventories in period t;
I_t: the inventory level at the end of period t;
ξ_t: the fuzzy demand for the specific product in period t;
α_t: the credibility service level requirement in period t.
Decision variables:
$x_{i,t}$: the production quantities at source i in period t.
Objective function:
The objective function includes the following costs:
The total inventory cost during T periods:

$$\sum_{t=1}^{T} \eta_t I_t^+;$$

The total production cost from N sources during T periods:

$$\sum_{i=1}^{N}\sum_{t=1}^{T} \tau_{i,t} x_{i,t}.$$

As a consequence, the objective is to minimize the total cost

$$\sum_{t=1}^{T} \eta_t I_t^+ + \sum_{i=1}^{N}\sum_{t=1}^{T} \tau_{i,t} x_{i,t}.$$

Constraints:
I: The inventory balance equation for each period is

$$I_t = I_{t-1} + \sum_{i=1}^{N} x_{i,t} - \xi_t, \quad t = 1, \ldots, T.$$

According to the recursive relation in inventory balance equation, it can be represented as

$$I_t = I_0 + \sum_{i=1}^{N}\sum_{j=1}^{t} x_{i,j} - \sum_{j=1}^{t}\xi_j, \quad t = 1, \ldots, T,$$

which denotes the inventory level at the end of period t.
II: The credibility service level constraint in each period is

$$\mathrm{Cr}\{I_t \geq 0\} \geq \alpha_t, \quad t = 1, \ldots, T,$$

which imposes the credibility of the fuzzy event that inventory level at the end of period t is not negative more than the predetermined service level requirement in each period.
III:

$$x_{i,t} \geq 0, \quad i = 1, \ldots, N, t = 1, \ldots, T.$$

The constraints state that the production quantities cannot be negative.

In this paper, we provide a new expected value approach to establishing a meaningful production planning and sourcing problem with credibility service levels. We adopt the expected value criterion on the objective function to build the N-product source, T-period production planning model, as follows:

$$\begin{array}{ll} \min & E\left[\sum_{t=1}^{T}\eta_t I_t^+ + \sum_{i=1}^{N}\sum_{t=1}^{T}\tau_{i,t}x_{i,t}\right] \\ \text{subject to:} & \\ & \mathrm{Cr}\{I_t \geq 0\} \geq \alpha_t, \qquad t = 1, \ldots, T \\ & x_{i,t} \geq 0, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T \\ & I_t^+ = \max\{0, I_t\}, \qquad t = 1, \ldots, T, \end{array} \tag{1}$$

where $I_t = I_{t-1} + \sum_{i=1}^{N} x_{i,t} - \xi_t$, fuzzy vector $\xi(\gamma) = (\eta_1(\gamma), \ldots \eta_T(\gamma), \tau_{1,1}(\gamma), \ldots \tau_{N,T}(\gamma), \xi_1(\gamma), \ldots \xi_T(\gamma))$ is obtained by piecing together the fuzzy components of the production planning problem data $\eta_t(\gamma)$, $\tau_{i,t}(\gamma)$ and $\xi_t(\gamma)$ in problem (1).

From the discussion above, the major differences between the fuzzy production planning problem and the stochastic production planning problem are summarized as follows:

• The uncertain data in fuzzy production planning problem are fuzzy variables with known possibility distributions, while the uncertain data in the stochastic production planning problem are random variables with known probability distributions. From the computation of the expected value of a fuzzy variable in [23], we can see that it is completely different from that of the expected value of a random variable. Therefore the solution method developed for stochastic programming problems cannot be applied to fuzzy ones.

• The service level requirements in the fuzzy production planning problem are the computation of the credibility levels of fuzzy events, while in the stochastic problem, it is the computation of the probability levels of stochastic events. From the computation of credibility in [23], we can see it is quite different from that of probability.

These differences lead to an inability to apply the solution techniques developed for stochastic production planning problems to the fuzzy ones. To overcome the difficulties, we attempt to employ a PSO algorithm in Section 4 for solving the proposed production planning problem.

4. Solution methods

Since the fuzzy production planning problem in Section 3 is not generally a convex programming one, the conventional optimization methods usually fail to find a global optimal solution of the problem. In order to solve the problem, we suggest two algorithms to solve the proposed fuzzy production planning problem. The first is an approximation-based PSO algorithm, and the second is a hybrid PSO algorithm combining AA, NN and PSO.

4.1. Handling the credibility constraints

In some special cases, we may transform the credibility constraint into its crisp equivalent form.

Theorem 1. *Let ξ be a normal fuzzy variable with the possibility distribution $\mu_\xi(r) = \exp\left(-(r-a)^2/\sigma^2\right)$, $a \in \Re$, $\sigma > 0$. Then, for any given credibility level $\alpha \in (0, 1]$, we have:*

(i) *When $\alpha < 0.5$, $\mathrm{Cr}\{\xi \leq t\} \geq \alpha$ if and only if $(t-a)^2 + \sigma^2 \ln 2\alpha \leq 0$;*
(ii) *When $\alpha \geq 0.5$, $\mathrm{Cr}\{\xi \leq t\} \geq \alpha$ if and only if $(t-a)^2 + \sigma^2 \ln 2(1-\alpha) \geq 0$.*

Proof. From the possibility distribution of ξ and the computation method of $\mathrm{Cr}\{\xi \leq t\}$, we can obtain

$$\mathrm{Cr}\{\xi \leq t\} = \begin{cases} 1 - \dfrac{1}{2}\exp\left(-(t-a)^2/\sigma^2\right) & \text{if } t \geq a \\ \dfrac{1}{2}\exp\left(-(t-a)^2/\sigma^2\right) & \text{if } t < a. \end{cases}$$

When $\alpha < 0.5$, we have:

$$\frac{1}{2}\exp\left(-(t-a)^2/\sigma^2\right) \geq \alpha,$$

i.e.,

$$\exp\left(-(t-a)^2/\sigma^2\right) \geq 2\alpha,$$

therefore, we can obtain

$$-(t-a)^2/\sigma^2 \geq \ln 2\alpha,$$

i.e.

$$(t-a)^2 + \sigma^2 \ln 2\alpha \leq 0.$$

Similarly, when $\alpha \geq 0.5$, we can obtain the deterministic form of $\mathrm{Cr}\{\xi \leq t\} \geq \alpha$ as follows:

$$(t-a)^2 + \sigma^2 \ln 2(1-\alpha) \geq 0. \quad \square$$

According to Theorem 1, we can obtain the more general results.

Theorem 2. *Let*

$$g(\boldsymbol{x}, \boldsymbol{\xi}) = f_1(\boldsymbol{x})\xi_1 + f_2(\boldsymbol{x})\xi_2 + \cdots f_n(\boldsymbol{x})\xi_n + f_0(\boldsymbol{x}),$$

where ξ_k, $k = 1, 2, \ldots n$, are mutually independent normal fuzzy variables with the following possibility distribution functions

$$\mu_\xi(r) = \exp\left(-(r-a_k)^2/\sigma_k^2\right), a_k \in \Re, \sigma_k > 0, \quad k = 1, 2, \ldots n.$$

If $f_k^+(\boldsymbol{x}) = f_k(\boldsymbol{x}) \bigvee 0$ and $f_k^-(\boldsymbol{x}) = -f_k(\boldsymbol{x}) \bigvee 0$, $k = 1, 2, \ldots n$, then, for any given credibility level $\alpha \in (0, 1]$, we have:

(a) *When $\alpha < 0.5$, $\mathrm{Cr}\{g(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha$ if and only if*

$$\left(-f_0(\boldsymbol{x}) - \sum_{k=1}^n a_k f_k^+(\boldsymbol{x}) + \sum_{k=1}^n a_k f_k^-(\boldsymbol{x})\right)^2 + \left(\sum_{k=1}^n \sigma_k f_k^+(\boldsymbol{x}) - \sum_{k=1}^n \sigma_k f_k^-(\boldsymbol{x})\right)^2 \ln 2\alpha \leq 0;$$

(b) *When $\alpha \geq 0.5$, $\mathrm{Cr}\{g(\boldsymbol{x}, \boldsymbol{\xi}) \leq 0\} \geq \alpha$ if and only if*

$$\left(-f_0(\boldsymbol{x}) - \sum_{k=1}^n a_k f_k^+(\boldsymbol{x}) + \sum_{k=1}^n a_k f_k^-(\boldsymbol{x})\right)^2 + \left(\sum_{k=1}^n \sigma_k f_k^+(\boldsymbol{x}) - \sum_{k=1}^n \sigma_k f_k^-(\boldsymbol{x})\right)^2 \ln 2(1-\alpha) \geq 0.$$

Proof. By the negativity of $f_k^+(\boldsymbol{x})$ and $f_k^-(\boldsymbol{x})$, and $f(\boldsymbol{x}) = f_k^+(\boldsymbol{x}) - f_k^-(\boldsymbol{x})$, we have

$$\begin{aligned} g(\boldsymbol{x}, \boldsymbol{\xi}) &= \sum_{k=1}^n f(\boldsymbol{x})\xi_k + f_0(\boldsymbol{x}) \\ &= \sum_{k=1}^n \left[f_k^+(\boldsymbol{x}) - f_k^-(\boldsymbol{x})\right]\xi_k + f_0(\boldsymbol{x}) \\ &= \sum_{k=1}^n \left[f_k^+(\boldsymbol{x})\xi_k + f_k^-(\boldsymbol{x})\xi_k'\right] + f_0(\boldsymbol{x}) \end{aligned}$$

where ξ_k' are normal fuzzy variables with the parameters $(-a_k, -\sigma_k)$, $k = 1, 2, \ldots n$. According to the computation rule of normal fuzzy variables [20], $g(\boldsymbol{x}, \boldsymbol{\xi}) - f_0(\boldsymbol{x})$ is also a normal fuzzy variable with the parameter $\left(\sum_{k=1}^n \left[f_k^+(\boldsymbol{x})a_k - f_k^-(\boldsymbol{x})a_k\right], \sum_{k=1}^n \left[f_k^+(\boldsymbol{x})\sigma_k - f_k^-(\boldsymbol{x})\sigma_k\right]\right)$. It follows from Theorem 1 that the assertion can be proved. $\square$

The credibility service levels

$$\mathrm{Cr}\{I_t \geq 0\} \geq \alpha_t, \quad t = 1, \ldots, T$$

can be represented as

$$\mathrm{Cr}\left\{I_0 + \sum_{i=1}^N \sum_{j=1}^t x_{i,j} - \sum_{j=1}^t \xi_j \geq 0\right\} \geq \alpha_t, \quad t = 1, \ldots, T.$$

For simplicity, assume ξ_j are normal fuzzy variables with the parameters (a_j, σ_j), $j = 1, 2, \ldots T$. When the fuzzy demands are mutually independent normal fuzzy variables and $\alpha_t \geq 0.5$, $t = 1, \ldots, T$, according to Theorem 2, the credibility service levels become the following deterministic equivalent constraints

$$\left(I_0+\sum_{i=1}^{N}\sum_{j=1}^{t}x_{i,j}-\sum_{j=1}^{n}a_j\right)^2+\left(\sum_{j=1}^{T}\sigma_j\right)^2\ln 2(1-\alpha_t)\geq 0,\quad t=1,\ldots,T.$$

Therefore, model (1) becomes the following equivalent one

$$\begin{array}{ll}\min & E\left[\sum_{t=1}^{T}\eta_t I_t^+\right]+E\left[\sum_{i=1}^{N}\sum_{t=1}^{T}\tau_{i,t}x_{i,t}\right]\\ \text{subject to:} & \\ & \left(I_0+\sum_{i=1}^{N}\sum_{j=1}^{t}x_{i,j}-\sum_{j=1}^{T}a_j\right)^2+\left(\sum_{j=1}^{T}\sigma_j\right)^2\ln 2(1-\alpha_t)\geq 0,\quad t=1,\ldots,T\\ & x_{i,t}\geq 0,\quad i=1,\ldots,N,\ t=1,\ldots,T\\ & I_t^+=\max\{0,I_t\},\quad t=1,\ldots,T,\end{array}\tag{2}$$

where $I_t=I_{t-1}+\sum_{i=1}^{N}x_{i,t}-\xi_t,\ t=1,\ldots,T$.

4.2. Evaluating objective function by approximation approach

Let

$$C(\boldsymbol{x},\xi(\gamma))=\sum_{t=1}^{T}\eta_t I_t^+ +\sum_{i=1}^{N}\sum_{t=1}^{T}\tau_{i,t}x_{i,t},\tag{3}$$

$$E_\xi[C(\boldsymbol{x},\xi(\gamma))]=E\left[\sum_{t=1}^{T}\eta_t I_t^+\right]+E\left[\sum_{i=1}^{N}\sum_{t=1}^{T}\tau_{i,t}x_{i,t}\right]\tag{4}$$

where $\eta_t(\gamma)$, $\tau_{i,t}(\gamma)$ and $\xi_t(\gamma)$ are mutually independent fuzzy variables. Denote

$$Q(\boldsymbol{x},\boldsymbol{\xi}(\gamma))=\sum_{t=1}^{T}\eta_t I_t^+,\ \mathcal{Q}(\boldsymbol{x})=E[Q(\boldsymbol{x},\boldsymbol{\xi}(\gamma))].\tag{5}$$

In order to solve the production planning problem, it is only required to evaluate the inventory cost function

$$\mathcal{Q}:\boldsymbol{x}\to E[Q(\boldsymbol{x},\boldsymbol{\xi}(\gamma))]\tag{6}$$

where $\boldsymbol{\xi}(\gamma)=(\eta_1(\gamma),\eta_2(\gamma)\ldots,\eta_T(\gamma),\xi_1(\gamma),\xi_2(\gamma),\ldots\xi_T(\gamma))$ is the fuzzy vector obtained by piecing together fuzzy inventory cost and fuzzy demands in fuzzy production planning problem (2). For any given $\boldsymbol{x}$, we can evaluate the value of the function $\mathcal{Q}(\boldsymbol{x})$ at $\boldsymbol{x}$ according to the following two cases.

Case I: If $\boldsymbol{\xi}=(\xi_1,\ldots,\xi_{2T})$ is a discrete fuzzy vector taking values with possibility μ_i, $i=1,2,\ldots 2T$, and $\max_{1\leq i\leq 2T}\mu_i=1$, then, for each outcome value $\widehat{\xi}_i$ of $\boldsymbol{\xi}$, we can obtain the value of $Q(\boldsymbol{x},\widehat{\xi}_i)$.

Without any loss of generality, we assume that the indexing of the values of $Q(\boldsymbol{x},\widehat{\xi}_i)$ has been done in an increasing order, i.e. $Q(\boldsymbol{x},\widehat{\xi}_1)\leq Q(\boldsymbol{x},\widehat{\xi}_2)\leq\cdots\leq Q(\boldsymbol{x},\widehat{\xi}_{2T})$, then the expected value $Q(\boldsymbol{x})$ is given by

$$\mathcal{Q}(\boldsymbol{x})=\sum_{i=1}^{2T}w_i\,Q\left(\boldsymbol{x},\widehat{\xi}_i\right)\tag{7}$$

where the weights w_i, $i=1,2,\ldots,2T$ are given by

$$w_i=\frac{1}{2}\left(\max_{j=1}^{i}\mu_j-\max_{j=0}^{i-1}\mu_j\right)+\frac{1}{2}\left(\max_{j=i}^{2T}\mu_j-\max_{j=i+1}^{2T+1}\mu_j\right).\tag{8}$$

Case II: Suppose that $\boldsymbol{\xi}=(\xi_1,\xi_2,\ldots,\xi_{2T})$ is a continuous fuzzy vector with the following infinite support $\Xi=\prod_{j=1}^{2T}[a_j,b_j]$, $[a_j,b_j]$ is the support of ξ_j. In this case, we will try to use the approximation approach [26] to approximate the possibility distribution function of $\boldsymbol{\xi}$ by a sequence of possibility distribution functions of discrete fuzzy vectors $\{\boldsymbol{\zeta}_s\}$. The detailed approach can be described as follows.

For each integer s, we will define the discrete fuzzy vector $\boldsymbol{\zeta}_s=(\zeta_{s,1},\zeta_{s,2},\ldots,\zeta_{s,2T})$ by the following method:

For each $j\in\{1,2,\ldots,2T\}$, define fuzzy variables $\zeta_{s,j}=g_{s,j}(\xi_j)$ for $s=1,2,\ldots$, where the function $g_{s,j}$ is as follows

$$g_{s,j}(u_j)=\sup\left\{\frac{k_j}{s}|k_j\in Z,s.t.\frac{k_j}{s}\leq u_j\right\},\quad u_j\in[a_j,b_j]$$

and Z is the set of integers.

Moreover, for each j, $1 \le j \le 2T$, by the definition of $\zeta_{s,j}$, as ξ_j takes its values in $[a_j, b_j]$, the fuzzy vector $\zeta_{s,j}$ takes its values in the set $\{(k_j/s) \mid k_j = [sa_j], [sa_j]+1, \ldots, K_j\}$, where $[r]$ is the maximal integer such that $[r] \le r$, and $K_j = sb_j - 1$ or $[sb_j]$ according as sb_j is an integer or not an integer. What's more, for each integer k_j, the fuzzy vector $\zeta_{s,j}$ takes the value k_j/s as ξ_j takes its values in the interval $[k_j/s, (k_j+1)/s)$. Therefore, the possibility distribution of the fuzzy variable $\zeta_{s,j}$, denoted $\nu_{s,j}$ is

$$\nu_{s,j}\left(\frac{k_j}{s}\right) = \text{Pos}\left\{\gamma \mid \frac{k_j}{s} \le \xi_j(\gamma) < \frac{k_j+1}{s}\right\}$$

for $k_j = [sa_j], [sa_j]+1, \ldots, K_j$. From the construction of $\zeta_{s,j}$, for each $\gamma \in \Gamma$, we have

$$\xi_j(\gamma) - \frac{1}{s} < \zeta_{s,j}(\gamma) \le \xi_j(\gamma)$$

and $j = 1, 2, \ldots, 2T$. Therefore, we have

$$|\xi_j(\gamma) - \zeta_{s,j}(\gamma)| < \frac{1}{s}.$$

Note that $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}_s$ are $2T$-ary fuzzy vectors, and ξ_j and $\zeta_{s,j}$ are their jth components, respectively. Then we have

$$\|\boldsymbol{\zeta}_s(\gamma) - \boldsymbol{\xi}(\gamma)\| = \sqrt{\sum_{j=1}^{2T}(\zeta_{s,j}(\gamma) - \xi_j(\gamma))^2} \le \frac{\sqrt{2T}}{s}, \quad \gamma \in \Gamma,$$

which implies that the sequence $\{\boldsymbol{\zeta}_s\}$ of fuzzy vectors converges to fuzzy vector $\boldsymbol{\xi}$ uniformly.

We now provide an example to illustrate the AA described above.

Example 1. Suppose $T = 2$, $\eta_1 = \eta_2 = 1$, ξ_1 and ξ_2 are mutually independent normal fuzzy variables, their possibility distribution functions are $\exp(-(r-3)^2/0.5^2)$, $r \in [2, 4]$, and $\exp(-(r-6)^2/0.5^2)$, $r \in [5, 7]$, respectively. In this case, we denote that $\boldsymbol{\xi} = (\xi_1, \xi_2)$. Determine the possibility distributions of the discrete fuzzy vectors $\boldsymbol{\zeta}_s = (\zeta_{s,1}, \zeta_{s,2})$, $s = 1, 2, \ldots$, where the fuzzy variables $\zeta_{s,i} = g_{s,i}(\xi_i)$, $i = 1, 2$, with

$$g_{s,1}(u_1) = \sup\left\{\frac{k_1}{s} \mid k_1 \in Z, s.t. \frac{k_1}{s} \le u_1\right\}, \quad u_1 \in [2, 4],$$

and

$$g_{s,2}(u_2) = \sup\left\{\frac{k_2}{s} \mid k_2 \in Z, s.t. \frac{k_2}{s} \le u_2\right\}, \quad u_2 \in [5, 7].$$

We first deduce the possibility distributions of fuzzy variables $\zeta_{s,1}$, $s = 1, 2, \ldots$.

Let $s = 1$. Then fuzzy variable $\zeta_{1,1}$ takes the value 2 as ξ_1 takes its value in [2, 3), and takes the value 3 as ξ_1 takes its value in [3, 4]. Therefore, we have

$$\nu_{1,1}(2) = \text{Pos}\{2 \le \xi_1 < 3\} = 1; \qquad \nu_{1,1}(3) = \text{Pos}\{3 \le \xi_1 \le 4\} = 1$$

i.e., the fuzzy variable $\zeta_{1,1}$ takes on values 2 and 3 with possibility 1 each.

Let $s = 2$. Then fuzzy variable $\zeta_{2,1}$ takes the values: 2, 2.5, 3 and 3.5 as the fuzzy variable ξ_1 takes its values in the intervals [2, 2.5), [2.5, 3), [3, 3.5) and [3.5, 4], respectively. Therefore, we have

$$\nu_{2,1}(2) = \text{Pos}\{2 \le \xi_1 < 2.5\} = \mathrm{e}^{-1}; \qquad \nu_{2,1}(2.5) = \text{Pos}\{2.5 \le \xi_1 < 3\} = 1;$$
$$\nu_{2,1}(3) = \text{Pos}\{3 \le \xi_1 < 3.5\} = 1; \qquad \nu_{2,1}(3.5) = \text{Pos}\{3.5 \le \xi_1 \le 4\} = \mathrm{e}^{-1}$$

i.e., the fuzzy variable $\zeta_{2,1}$ takes on values 2, 2.5, 3, and 3.5 with possibility e^{-1}, 1, 1 and e^{-1}, respectively.

Generally, the fuzzy variable $\zeta_{s,1}$ takes on values k_1/s, $k_1 = 2s, s+1, \ldots, 4s$, and the possibility that $\zeta_{s,1}$ takes the value k_1/s is

$$\nu_{s,1}\left(\frac{k_1}{s}\right) = \begin{cases} \exp\left(-\left(\frac{k_1+1}{s} - 3\right)^2 \Big/ 0.5^2\right), & \text{if } 2s \le k_1 < 3s \\ \exp\left(-\left(\frac{k_1}{s} - 3\right)^2 \Big/ 0.5^2\right), & \text{if } 3s \le k_1 \le 4s \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Also, by the definition of $\zeta_{s,1}$, one has

$$\xi_1 - \frac{1}{s} < \zeta_{s,1} < \xi_1, \quad s = 1, 2, \ldots. \tag{10}$$

Using the similar method, we can obtain the possibility distributions of fuzzy variables $\zeta_{s,2}$

$$\nu_{s,2}\left(\frac{k_2}{s}\right)=\begin{cases}\exp\left(-\left(\frac{k_2+1}{s}-6\right)^2\Big/0.5^2\right), & \text{if } 5s\le k_2<6s\\ \exp\left(-\left(\frac{k_2}{s}-6\right)^2\Big/0.5^2\right), & \text{if } 6s\le k_2\le 7s\\ 0, & \text{otherwise.}\end{cases} \tag{11}$$

and the link between $\zeta_{s,2}$ and ξ_2

$$\xi_2-\frac{1}{s}<\zeta_{s,2}<\xi_2,\quad s=1,2,\ldots. \tag{12}$$

By (9) and (11), the possibility distribution of fuzzy vector $\boldsymbol{\zeta}_s=(\zeta_{s,1},\zeta_{s,2})$, denoted μ_s, is obtained as follows:

$$\mu_s\left(\frac{k_1}{s},\frac{k_2}{s}\right)=\begin{cases}\min\left\{\exp\left(-\left(\frac{k_1+1}{s}-3\right)^2\Big/0.5^2\right),\exp\left(-\left(\frac{k_2+1}{s}-6\right)^2\Big/0.5^2\right)\right\}, & \text{if } 2s\le k_1<3s,\ 5s\le k_2<6s\\ \min\left\{\exp\left(-\left(\frac{k_1+1}{s}-3\right)^2\Big/0.5^2\right),\exp\left(-\left(\frac{k_2}{s}-6\right)^2\Big/0.5^2\right)\right\}, & \text{if } 2s\le k_1<3s,\ 6s\le k_2\le 7s\\ \min\left\{\exp\left(-\left(\frac{k_1}{s}-3\right)^2\Big/0.5^2\right),\exp\left(-\left(\frac{k_2+1}{s}-6\right)^2\Big/0.5^2\right)\right\}, & \text{if } 3s\le k_1\le 4s,\ 5s\le k_2<6s\\ \min\left\{\exp\left(-\left(\frac{k_1}{s}-3\right)^2\Big/0.5^2\right),\exp\left(-\left(\frac{k_2}{s}-6\right)^2\Big/0.5^2\right)\right\}, & \text{if } 3s\le k_1\le 4s,\ 6s\le k_2\le 7s\\ 0, & \text{otherwise.}\end{cases}$$

In addition, it follows from (10) and (12)that

$$\|\boldsymbol{\zeta}_s-\boldsymbol{\xi}\|=\sqrt{(\zeta_{s,1}-\xi_1)^2+(\zeta_{s,2}-\xi_2)^2}<\frac{\sqrt{2}}{s},$$

which implies that the sequence $\{\boldsymbol{\zeta}_s\}$ of the discrete fuzzy vectors converges uniformly to the continuous fuzzy vector $\boldsymbol{\xi}$.

We now give the computation of the objective function according to the method proposed above. Let $\boldsymbol{\zeta}_s$ be the discretization of the fuzzy vector $\boldsymbol{\xi}$. For each fixed s, the vector $\boldsymbol{\zeta}_s$ takes on K values $\hat{\boldsymbol{\zeta}}_s^k=\left(\hat{\zeta}_{s,1}^k,\hat{\zeta}_{s,2}^k,\ldots,\hat{\zeta}_{s,2T}^k\right)$, $k=1,2,\ldots,K$, with $K=K_1K_2\ldots K_{2T}$, where K_i is the number of discrete points of ξ_i, $i=1,2,\ldots,2T$. The process to estimate the objective function is summarized as

Approximation method

Step 1. Generate K points $\hat{\boldsymbol{\zeta}}_s^k=\left(\hat{\zeta}_{s,1}^k,\hat{\zeta}_{s,2}^k,\ldots,\hat{\zeta}_{s,2T}^k\right)$ from the support Ξ of $\boldsymbol{\xi}$ for $k=1,2,\ldots,K$;
Step 2. Calculate $g_s(\hat{\zeta}_s^k)$ for $k=1,2,\ldots,K$;
Step 3. Set $\nu_k=\nu_{s,1}\left(\hat{\zeta}_{s,1}^k\right)\wedge\nu_{s,2}\left(\hat{\zeta}_{s,2}^k\right)\wedge\ldots\wedge\nu_{s,2T}\left(\hat{\zeta}_{s,2T}^k\right)$ and $Q_k=Q\left(\boldsymbol{x},\zeta_s^k\right)$ for $k=1,2,\ldots,K$;
Step 4. Rearrange the subscript k of ν_k and $Q(\boldsymbol{x},\zeta_s^k)$ such that $Q_1\le Q_2\le\cdots\le Q_K$ for $k=1,2,\ldots,K$;
Step 5. Calculate w_k according to (8) for $k=1,2,\ldots,K$;
Step 6. Return $\mathcal{Q}(\boldsymbol{x})$ via the estimation formula (7).

In what follows, we refer to the sequence $\{\boldsymbol{\zeta}_s\}$ of discrete fuzzy vector as the discretization of the fuzzy vector $\boldsymbol{\xi}$.

The convergence of AA is ensured by the following theorem. As a consequence, the original objective function $E_{\boldsymbol{\xi}}[Q(\mathbf{x},\boldsymbol{\xi}(\gamma))]$ can be estimated by the approximating objective function $E_{\boldsymbol{\xi}}[Q(\mathbf{x},\boldsymbol{\zeta}_s(\gamma))]$ provided that s is sufficiently large.

Theorem 3. *Consider fuzzy production planning problem (2). Suppose the fuzzy variables coefficient $\boldsymbol{\xi}$ is a continuous and bounded fuzzy vector and the sequence $\{\boldsymbol{\zeta}_s\}$ of fuzzy vectors is the discretization of $\boldsymbol{\xi}$, then for any given feasible decision $\boldsymbol{x}$, we have*

$$\lim_{s\to\infty}E_{\boldsymbol{\xi}}[Q(\boldsymbol{x},\boldsymbol{\zeta}_s(\gamma))]=E_{\boldsymbol{\xi}}[Q(\boldsymbol{x},\boldsymbol{\xi}(\gamma))].$$

Proof. For any given feasible solution $\boldsymbol{x}$, the suppositions of the theorem satisfy the conditions of [26, Theorem 3]. Thus the theorem is valid. The proof of the theorem is complete. □

We now provide an example to help our understanding the result of Theorem 3.

Example 2. Suppose $T = 2$, $\eta_1 = \eta_2 = 1$, $\boldsymbol{\xi} = (\xi_1, \xi_2)$ is the fuzzy vector defined in Example 1, and $\{\boldsymbol{\zeta}_s\}$ is the discretization of $\boldsymbol{\xi}$.

By

$$Q(\mathbf{x}, \boldsymbol{\xi}(\gamma)) = \sum_{t=1}^{T} \eta_t I_t^+,$$

we have

$$Q(\mathbf{x}, \boldsymbol{\xi}(\gamma)) = \xi_1 + \xi_2,$$

and

$$Q(\mathbf{x}, \boldsymbol{\zeta}(\gamma)) = \zeta_{s,1} + \zeta_{s,2}.$$

Show that

$$\lim_{s\to\infty} E_{\boldsymbol{\xi}}[Q(\mathbf{x}, \boldsymbol{\zeta}_s(\gamma))] = E_{\boldsymbol{\xi}}[Q(\mathbf{x}, \boldsymbol{\xi}(\gamma))].$$

From [23], the expected value of a normal fuzzy variable with the parameters (a, σ) is a. Therefore, by the linearity of the expected value operator [29], we have

$$E[\xi_1 + \xi_2] = E[\xi_1] + E[\xi_2] = 9.$$

On the other hand, from Example 1, the fuzzy variable $\zeta_{s,1}$ has the possibility distribution (9). As a consequence, we can obtain the weights

$$\omega_{k_1} = \begin{cases} \dfrac{1}{2}\left(\exp\left(-\left(\dfrac{k_1+1}{s} - 3\right)^2 \Big/ 0.5^2\right) - \exp\left(-\left(\dfrac{k_1}{s} - 3\right)^2 \Big/ 0.5^2\right)\right), & \text{if } 2s \le k_1 < 3s, \\ \dfrac{1}{2}\left(\exp\left(-\left(\dfrac{k_1}{s} - 3\right)^2 \Big/ 0.5^2\right) - \exp\left(-\left(\dfrac{k_1+1}{s} - 3\right)^2 \Big/ 0.5^2\right)\right), & \text{if } 3s \le k_1 \le 4s \end{cases}$$

by applying (8). It follows from (7) that the expected value of $\zeta_{s,1}$ is

$$E[\zeta_{s,1}] = \sum_{k_1=2s}^{4s} w_{k_1}\frac{k_1}{s} = \sum_{k_1=2s}^{3s-1} w_{k_1}\frac{k_1}{s} + \sum_{k_1=3s}^{4s-1} w_{k_1}\frac{k_1}{s} = \frac{3}{2} + \frac{3s-1}{2s}.$$

Similarly, according to the possibility distribution (11) of the fuzzy variable $\zeta_{s,2}$, we deduce

$$\omega_{k_2} = \begin{cases} \dfrac{1}{2}\left(\exp\left(-\left(\dfrac{k_2+1}{s} - 6\right)^2 \Big/ 0.5^2\right) - \exp\left(-\left(\dfrac{k_2}{s} - 6\right)^2 \Big/ 0.5^2\right)\right), & \text{if } 5s \le k_2 < 6s, \\ \dfrac{1}{2}\left(\exp\left(-\left(\dfrac{k_2}{s} - 6\right)^2 \Big/ 0.5^2\right) - \exp\left(-\left(\dfrac{k_2+1}{s} - 6\right)^2 \Big/ 0.5^2\right)\right), & \text{if } 6s \le k_2 \le 7s, \end{cases}$$

and

$$E[\zeta_{s,2}] = \sum_{k_2=5s}^{7s} w_{k_2}\frac{k_2}{s} = \sum_{k_2=5s}^{6s-1} w_{k_2}\frac{k_2}{s} + \sum_{k_2=6s}^{7s-1} w_{k_2}\frac{k_2}{s} = 3 + \frac{6s-1}{2s}.$$

By the independence of fuzzy variables (see [29]), we can obtain

$$E[\zeta_{s,1} + \zeta_{s,2}] = E[\zeta_{s,1}] + E[\zeta_{s,2}]$$

which yields

$$\lim_{s\to\infty} E_{\boldsymbol{\zeta}}[\zeta_{s,1} + \zeta_{s,2}] = 9 = E_{\boldsymbol{\xi}}[\xi_1 + \xi_2].$$

4.3. Particle swarm optimization

PSO algorithm, originally developed in [27], is a method for optimization on metaphor of social behavior of flocks of birds and/or schools of fish. Compared to other evolutionary algorithms, PSO has a faster convergence rate and many fewer parameters to adjust, which makes it particularly easy to implement. Recently the PSO algorithm has attracted much attention and been successfully applied in the fields of evolutionary computing, unconstrained continuous optimization problems and many others [28]. As for constrained optimization problems, Dong et al. [30] proposed a PSO algorithm embedded with a constraint fitness priority-based ranking method.

PSO is based on an n-dimensional of pop_size particles, each of which indicates a possible solution of the problem space. Each particle has its own best position (pbest) which represents the personal smallest objective value so far at time t. The global best particle (gbest) represents the best particle found so far at time t in the colony.

As a consequence, the new velocity of the ith particle is updated by the following formula

$$V_i(t+1) = \omega V_i(t) + c_1 r_1(P_i(t) - X_i(t)) + c_2 r_2(P_g(t) - X_i(t)) \tag{13}$$

while the new position of the ith particle is renewed by

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{14}$$

where $i = 1, 2, \ldots, pop\_size$; ω is called the inertia coefficient; c_1 and c_2 are learning rates and usually $c_1 = c_2 = 2$, r_1 and r_2 are are two independent random numbers generated randomly in the unit interval [0, 1]. The solution process of PSO combined AA is summarized as follows.

Algorithm 1 (*PSO Algorithm*).

Step 1. Initialize pop_size particles with random positions and velocities, then compute their objective values by AA.
Step 2. Set pbest of each particle and its objective value equal to its current position and objective value, and set gbest and its objective value equal to the position and objective value of the best initial particle;
Step 3. Renew the velocity and position of each particle according to formulas (13) and (14), respectively.
Step 4. Calculate the objective values for all particles by AA.
Step 5. For each particle, compare the current objective value with that of its pbest. If the current objective value is smaller than that of pbest, then renew pbest and its objective value with the current position and objective value.
Step 6. Find the best particle of the current swarm with the smallest objective value. If the objective value is smaller than that of gbest, then renew gbest and its objective value with the position and objective value of the current best particle.
Step 7. Repeat the third to six steps for a given number of cycles.
Step 8. Return the gbest and its objective value as the optimal solution and the optimal value.

4.4. Hybrid PSO algorithm

So far, we have designed the PSO algorithm combined with AA. During the solution process of PSO, we employ AA to compute the objective values of all particles. Thus it is a time-consuming process. To speed up the solution process, we desire to replace the objective function $\mathcal{Q}(\mathbf{x})$ by an NN since a trained NN has the ability to approximate functions. In this paper, we employ the fast BP algorithm to train a feedforward NN to approximate the objective function $\mathcal{Q}(\mathbf{x})$. Usually, an NN with two hidden layers is better in generation than the NN with one hidden layer. But in most applications, an NN with one hidden layer is enough to be a universal approximator for any integrable functions. Thus, in this paper, we only consider the NN with input layer, one hidden layer and output layer connected in a feedforward way, in which there are n_1 input neurons in the input layer representing the input values of decision variables, p neurons in the hidden layer and 1 neuron in the output layer representing the value of the objective function. Let $\{(x_i, y_i) \mid i = 1, 2, \ldots, n\}$ be a set of input–output data generated by AA. The training process is to find the best weight vector $\boldsymbol{w}_i$ that minimizes the following error functions

$$Err(w_i) = \frac{1}{2}\sum_{i=1}^{n} |F(x_i, w_i) - y_i|, \quad i = 1, 2, \ldots, n.$$

In the following, we incorporate AA, neural network (NN) and particle swarm optimization (PSO) algorithm to produce a hybrid PSO algorithm for solving the fuzzy production planning problem. In our proposed hybrid PSO algorithm, the technique of AA is used to generate a set of input–output data for the expected value objective. Using the generated data set, an NN is trained to approximate the expected value function. After NN is well trained, it is embedded into a PSO algorithm to produce a hybrid algorithm to search for the optimal solution. Thus, during the solution process of the hybrid PSO algorithm, the objective value of all particles can be computed by the trained NN instead of AA. It is well-known that a trained NN has high speed of operations, thus much time can be saved by the designed hybrid PSO algorithm. The fact will be demonstrated in the next section via one numerical example. We now summarize the process of the hybrid PSO algorithm in the following.

Algorithm 2 (*A Hybrid PSO Algorithm*).

Step 1. Generate a set of input–output data for the expected value function

$$\mathcal{Q} : \mathbf{x} \to E_{\xi}[Q(\mathbf{x}, \xi(\gamma))]$$

by the proposed AA;
Step 2. Train an NN to approximate the expected value function $\mathcal{Q}(\mathbf{x}, \boldsymbol{\xi}(\gamma))$ by the generated data;

Step 3. Initialize pop_size particles with random positions and velocities, and evaluate the objective values for all particles by the trained NN;
Step 4. Set pbest of each particle and its objective value equal to its current position and objective value, and set gbest and its objective value equal to the position and objective value of the best initial particle;
Step 5. Renew the velocity and position of each particle according to formulas (13) and (14), respectively;
Step 6. Calculate the objective values for all particles by the trained NN;
Step 7. For each particle, compare the current objective value with that of its pbest. If the current objective value is smaller than that of pbest, then renew pbest and its objective value with the current position and objective value;
Step 8. Find the best particle of the current swarm with the smallest objective value. If the objective value is smaller than that of gbest, then renew gbest and its objective value with the position and objective value of the current best particle;
Step 9. Repeat the fifth to eighth steps for a given number of cycles;
Step 10. Return the gbest and its objective value as the optimal solution and the optimal value.

5. One numerical example

In order to compare the effectiveness of the proposed two algorithms, we consider the following fuzzy production planning problem with $K = T = 6, I_0 = 0$.

$$
\begin{array}{l}
\min \quad E\left[\sum_{t=1}^{6} \tau_t I_t^+\right] + E\left[\sum_{i=1}^{6}\sum_{t=1}^{6} \eta_{i,t} x_{i,t}\right] \\
\text{subject to:} \\
\quad \left(\sum_{i=1}^{6} x_{i,1} - 54\right)^2 + 9\ln 0.20 \ge 0, \\
\quad \left(\sum_{i=1}^{6}\sum_{j=1}^{2} x_{i,j} - 54\right)^2 + 9\ln 0.16 \ge 0, \\
\quad \left(\sum_{i=1}^{6}\sum_{j=1}^{3} x_{i,j} - 54\right)^2 + 9\ln 0.12 \ge 0, \\
\quad \left(\sum_{i=1}^{6}\sum_{j=1}^{4} x_{i,j} - 54\right)^2 + 9\ln 0.14 \ge 0, \\
\quad \left(\sum_{i=1}^{6}\sum_{j=1}^{5} x_{i,j} - 54\right)^2 + 9\ln 0.20 \ge 0, \\
\quad \left(\sum_{i=1}^{6}\sum_{j=1}^{6} x_{i,j} - 54\right)^2 + 9\ln 0.16 \ge 0, \\
\quad x_{i,t} \ge 0, \quad i = 1, \ldots, 6, t = 1, \ldots, 6 \\
\quad I_t^+ = \max\{0, I_t\}, \quad t = 1, \ldots, 6,
\end{array}
\tag{15}
$$

where $I_t = I_{t-1} + \sum_{i=1}^{N} x_{i,t} - \xi_t, t = 1, \ldots, 6$.
The required data set for this manufacturing system is collected in Table 1, while the possibility distributions of fuzzy costs in this production planning problem is provided in Table 1. The demands $\xi_t, t = 1, \ldots 6$, are normal fuzzy variables with the following possibility distributions

$$
\begin{array}{ll}
\mu_{\xi_1} = \exp\left(-\left(\frac{r-10}{0.5}\right)^2\right), & \mu_{\xi_2} = \exp\left(-\left(\frac{r-9}{0.5}\right)^2\right), \\
\mu_{\xi_3} = \exp\left(-\left(\frac{r-11}{0.5}\right)^2\right), & \mu_{\xi_4} = \exp\left(-\left(\frac{r-9}{0.5}\right)^2\right), \\
\mu_{\xi_5} = \exp\left(-\left(\frac{r-8}{0.5}\right)^2\right), & \mu_{\xi_6} = \exp\left(-\left(\frac{r-7}{0.5}\right)^2\right).
\end{array}
$$

Also, the fuzzy variables involved in this problem are assumed to be mutually independent.

We first solve the fuzzy production planning problem via PSO algorithm. During the solution process, for each particle $\boldsymbol{x}$, we generate 3000 sample points via AA to estimate the objective value of the particle

$$\mathcal{Q} : \mathbf{x} \to E_\xi[Q(\mathbf{x}, \boldsymbol{\xi}(\gamma))].$$

Table 1
The data set for production planning problem.

Production profit matrix (η_{it})						
Periods	1	2	3	4	5	6
Product source						
1	(4, 7, 10)	(5, 9, 11)	(5, 7, 9)	(6, 8, 10)	(3, 7, 10)	(4, 7, 11)
2	(7, 9, 11)	(5, 7, 9)	(8, 10, 12)	(4, 7, 10)	(3, 6, 9)	(5, 7, 10)
3	(3, 7, 11)	(4, 6, 8)	(5, 7, 9)	(2, 4, 6)	(8, 10, 12)	(7, 8, 12)
4	(8, 10, 12)	(7, 9, 11)	(5, 8, 9)	(6, 9, 12)	(3, 5, 8)	(5, 7, 12)
5	(5, 8, 13)	(5, 7, 10)	(3, 6, 9)	(3, 5, 8)	(5, 10, 12)	(5, 7, 12)
6	(2, 6, 9)	(5, 8, 10)	(4, 8, 12)	(3, 5, 10)	(2, 7, 12)	(5, 7, 12)
Inventory cost matrix (τ_t)						
Periods	1	2	3	4	5	6
	(2, 3, 4)	(1, 2, 3)	(0.5, 1.5, 2.5)	(1, 3, 4)	(2, 3, 5)	(3, 4, 6)
Service level constraints (α_t)						
Periods	1	2	3	4	5	6
	0.90	0.92	0.94	0.93	0.90	0.92

Table 2
Comparison solution of PSO algorithm.

pop_size	gen	Optimal solution	Objective value	Error %
30	500	(4.0676, 2.4533, 0.2652, 9.9335, 0.0715, 0.0000, 10.0000, 5.7894, 10.0000, 9.9330, 10.0000, 9.5209, 5.8743, 9.9420, 2.6099, 9.9635, 6.0703, 2.7580, 0.0000, 0.5301, 10.0000, 1.2660, 10.0000, 9.8286, 0.0000, 10.0000, 4.1072, 10.0000, 0.6867, 0.0000, 5.1727, 10.0000, 8.5253, 7.0652, 7.5758, 9.4378)	1517.1689	0.00
30	400	(1.1584, 8.4299, 0.2301, 0.0486, 9.3566, 0.0000, 7.8773, 0.0000, 9.9937, 8.1679, 9.5109, 10.0000, 8.5018, 10.0000, 3.4490, 8.4545, 9.8393, 10.0000, 7.5881, 0.0233, 9.9628, 0.0000, 10.0000, 3.2161, 10.0000, 10.0000, 0.4239, 10.0000, 0.0000, 0.0000, 9.8405, 10.0000, 6.6536, 10.0000, 10.0000, 2.6889)	1540.8826	1.56
35	500	(0.1150, 0.7426, 9.8341, 9.9650, 8.8009, 0.0000, 10.0000, 10.0000, 9.9360, 9.5159, 9.3707, 0.0000, 0.0000, 0.0000, 0.5777, 10.0000, 9.6013, 0.0000, 10.0000, 7.3661, 10.0000, 0.0000, 8.9821, 10.0000, 10.0000, 10.0000, 0.0000, 10.0000, 0.1158, 0.0000, 0.0000, 6.5134, 8.3454, 8.6061, 0.0183, 3.3957)	1541.8240	1.63
35	400	(5.1444, 0.0000, 2.6437, 9.1986, 2.9621, 0.0000, 1.0340, 4.0296, 10.0000, 0.1035, 7.8349, 0.0000, 8.1515, 0.0000, 0.0000, 9.9328, 5.9444, 0.0000, 10.0000, 10.0000, 10.0000, 3.1658, 10.0000, 7.6949, 10.0000, 10.0000, 0.2242, 10.0000, 0.0000, 1.4703, 10.0000, 10.0000, 2.1131, 8.4846, 10.0000, 1.7175)	1560.9036	2.88
30	450	(0.0000, 10.0000, 0.1717, 4.7251, 0.0000, 0.0000, 9.8030, 6.9559, 10.0000, 10.0000, 9.9276, 9.7510, 5.3057, 6.3123, 0.9181, 8.4900, 9.0324, 4.1473, 0.0000, 0.0448, 10.0000, 0.0000, 10.0000, 6.8061, 8.6616, 10.0000, 2.6795, 8.2463, 0.0574, 4.1344, 0.0000, 9.9563, 10.0000, 1.7344, 10.0000, 9.9487)	1528.5236	0.75
35	450	(2.1864, 0.0000, 3.3191, 10.0000, 0.0000, 0.0000, 3.2782, 4.2353, 10.0000, 3.8728, 7.7924, 0.0000, 9.1090, 0.1199, 0.0000, 7.1773, 3.0637, 0.0000, 10.0000, 10.0000, 10.0000, 0.4120, 0.0000, 0.0000, 10.0000, 10.0000, 1.4752, 6.6218, 0.0000, 0.7522, 4.1970, 9.1249, 1.1816, 8.3232, 6.2453, 2.1219)	1546.9537	1.96

To identify the parameters' influence on the solution quality, a numerical study is made to compare the solutions obtained by running the approximation-based PSO with careful variation of parameters. The computational results are reported in Table 2, where the parameter 'relative error' in the last column is defined as $((optimal\ value - objective\ value)/optimal\ value \times 100\%)$ with the the optimal value being the least one of the objective values in the fourth column. It can be seen from Table 2 that the relative errors do not exceed 3% when various parameters of PSO are selected, which implies the approximation-based algorithm is robust to parameters settings. However, during the solution process, we are required to employ AA to compute the objective value of all particles, which results in the solution process being slow. To speed up the solution process, in the following we use hybrid PSO algorithm to solve the above problem.

We first generate a set of 3000 input–output data. Then we train an NN via the input–output data to approximate the expected value function $\mathcal{Q}(\mathbf{x})$. After the NN is well-trained, it is embedded into a PSO to produce a hybrid PSO algorithm to search for the optimal solution. In view of parameters' influence on solution quality, we compare solutions by careful variations of parameters in PSO. The computational results of the hybrid PSO algorithm are collected in Table 3, in which the parameter 'relative error' is defined the same as above. From Table 3, we can see that the relative errors do not exceed 3%

Table 3
Comparison solution of hybrid PSO algorithm.

pop_size	gen	Optimal solution	Objective value	Error %
30	500	(8.0119, 6.9352, 7.9598, 2.5985, 0.8854, 3.1683, 2.6895, 7.0059, 1.6315, 1.6901, 4.6548, 2.9025, 7.6225, 3.6019, 0.6304, 3.2533, 4.7925, 8.4013, 4.7769, 0.5215, 7.3680, 4.2480, 4.1300, 3.6324 2.8037, 3.8554, 0.9156, 0.0000, 3.5250, 1.7946, 6.1434, 8.7538, 4.9033, 7.7157, 0.0000, 0.4046)	1542.0952	0.00
30	400	(8.0811, 6.8988, 7.8636, 2.5642, 1.0585, 3.3366, 2.7983, 6.9857, 1.6558, 1.8412, 4.7840, 2.9188, 7.6681, 3.7572, 0.8226, 3.3179, 4.9008, 8.3056, 4.7081, 0.6084, 7.3531, 4.2409, 4.2395, 3.7297, 2.8851, 3.9523, 1.1597, 0.0006, 3.6729, 1.8908, 6.1722, 8.7550, 4.9115, 7.6900, 0.0000, 0.4046)	1561.1645	1.23
35	500	(10.0000, 10.0000, 7.7584, 1.1754, 0.0000, 3.0792, 4.9571, 6.4032, 0.0000, 1.7794, 6.3969, 0.4374, 10.0000, 1.7821, 0.1518, 1.4530, 7.2165, 7.1393, 4.5673, 0.0675, 9.6136, 4.1914, 5.5912, 1.3628, 0.7878, 3.0970, 2.4084, 1.8469, 2.0453, 0.0000, 4.4282, 10.0000, 6.2173, 8.4129, 0.0000, 0.1728)	1564.2169	1.43
35	400	(10.0000, 10.0000, 7.7814, 1.2911, 0.0000, 3.2456,) 4.9635, 6.3200, 0.0000, 1.8891, 6.4977, 0.5691, 10.0000, 1.9941, 0.2492, 1.7819, 7.3642, 7.1693, 4.6878, 0.2222, 9.3965, 4.1660, 5.5401, 1.5492, 0.9301, 1.9302, 2.3173, 0.0000, 3.1560, 2.5011, 4.4477, 9.9999, 6.1238, 8.5291, 0.0000, 0.1728)	1586.2874	2.87
30	450	(8.0449, 6.9110, 7.9143, 2.5978, 0.9860, 3.2417, 2.7698, 7.0007, 1.6770, 1.7634, 4.6769, 2.9121, 7.6412, 3.6613, 0.7166, 3.2903, 4.8392, 8.3659, 4.7666, 0.5598, 7.3725, 4.2402, 4.1894, 3.6730, 2.8511, 3.8914, 1.0251, 0.0000, 3.6051, 1.8311, 6.1622, 8.7540, 4.9103, 7.6974, 0.0000, 0.4046)	1551.9369	0.64
35	450	(10.0000, 10.0000, 7.8186, 1.2413, 0.0000, 3.1375, 5.0053, 6.3736, 0.0000, 1.8598, 6.4313, 0.4801, 4.6343, 1.8712, 0.2150, 1.5970, 7.2898, 7.1827, 5.5655, 0.1262, 9.5761, 4.1491, 5.5680, 1.4301, 0.8400, 1.8800, 2.1639, 0.0000, 3.1521, 2.4500, 4.4424, 9.9984, 6.1773, 8.4141, 0.0000, 0.1728)	1575.2225	2.15

Table 4
Comparison of PSO algorithm and hybrid PSO algorithm.

pop_size	gen	Objective1 (hybrid PSO)	Objective2 (PSO)	Deviation (%)	Time difference (min)
30	500	1542.0952	1517.1689	1.61	20.10
30	400	1561.1645	1540.8826	1.30	18.50
35	500	1564.2169	1541.8240	1.43	19.60
35	400	1586.2874	1560.9036	1.60	21.50
30	450	1551.9369	1528.5236	1.50	19.60
35	450	1575.2225	1546.9537	1.79	20.80

when various parameters of PSO are selected, which implies that the hybrid PSO algorithm is also robust to the parameters settings.

Finally, we compare the computational results about PSO and hybrid PSO algorithms. For this purpose, we define two indices, 'deviation' and 'time difference'. The 'deviation' is defined as $(objective1 - objective2)/objective1 \times 100\%$, in which objective1 is the objective value obtained via hybrid PSO algorithm, and objective2 is the objective value obtained via PSO algorithm; while the 'time difference' is defined as the difference between the solution time consumed by PSO algorithm and that consumed by hybrid PSO algorithm. The comparison of the two algorithms is provided in Table 4, from which we can see the *deviation* about objective values does not exceed 1.8%, but the *time difference* is about 20 min, which depend on problem size and the number of sample points discretized via AA. Therefore, much time can be saved if we adopt hybrid PSO algorithm to solve the proposed production planning problem. From this viewpoint, we can conclude that the hybrid PSO algorithm is more effective than PSO algorithm.

6. Conclusions

In this paper, we have presented a new class of fuzzy production planning problem. When demands are independent normal fuzzy variables, we have transformed the credibility constraint into its crisp equivalent form. Since the possibility distribution of fuzzy variables coefficients has an infinite support in this fuzzy production planning model, the fuzzy production planning problem is inherently an infinite-dimensional optimization one, and we cannot solve it via conventional optimization algorithms. To avoid this difficulty, this paper designed two algorithms to solve the proposed production planning problem. The first is an approximation-based PSO algorithm, and the second is the hybrid PSO algorithm combining the AA, NN, and PSO algorithm. One numerical example was provided to compare the effectiveness of the two algorithms. The computational results demonstrated that both algorithms are robust to parameters' settings, but the hybrid PSO algorithm can save much time compared with the PSO algorithm. From this viewpoint, we concluded that the designed hybrid PSO algorithm is more effective than PSO one when we employ them to solve the proposed production planning problem.

Acknowledgments

The authors would like to thank the reviewers for their valuable comments that have been incorporated into the new version of this paper. This work was supported by the Program for One Hundred Excellent and Innovative Talents in Colleges and Universities of Hebei Province, the Natural Science Foundation of Hebei Province A2008000563, and the National Natural Science Foundation of China Grant No.70571021.

References

[1] J. Galbraith, Designing Complex Organizations, Addison-Wesley, Reading, MA, 1973.
[2] C. Ho, Evaluating the impact of operating environments on MRP system nervousness, International Journal of Production Research 27 (1989) 1115–1135.
[3] D.C. Whybark, J.G. Williams, Material requirements planning under uncertainty, Decision Science 7 (1976) 595–606.
[4] A. Bũchel, Stochastic material requirements planning for optional parts, International Journal of Production Research 21 (1983) 511–527.
[5] H. Gfrerer, G. Zäpfel, Hierarchical model for production planning in the case of uncertain demand, European Journal of Operational Research 86 (1995) 142–161.
[6] D. Kira, M. Kusy, I. Rakita, A stochastic linear programming approach to hierarchical production planning, Journal of the OR Society 48 (1997) 207–211.
[7] R.C. Wang, H.H. Fang, Aggregate production planning with multiple objectives in a fuzzy environment, European Journal of Operational Research 133 (2001) 521–536.
[8] S.D. Thompson, W.J. Davis, An integrated approach for modeling uncertainty in aggregate production planning, IEEE Transactions on Systems, Man and Cybernetics 20 (1990) 1000–1012.
[9] M. Gen, Y. Tsujimura, K. Ida, Method for solving multi objective aggregate production planning problem with fuzzy parameters, Computers & Industrial Engineering 23 (1992) 117–120.
[10] D. Petrovic, R. Roy, R. Petrovic, Supply chain modelling using fuzzy sets, International Journal of Production Economics 59 (1999) 443–453.
[11] A. Gupta, C.D. Maranas, Managing demand uncertainty in supply chain planning, Computers & Chemical Engineering 27 (2003) 1216–1227.
[12] G.R. Bitran, H.H. Yanasee, Deterministic approximations to stochastic production problems, Operations Research 32 (1984) 999–1018.
[13] S.C.H. Leung, S.O.S. Tsang, W.L. Ng, Yue Wu, A robust optimization model for multi-site production planning problem in an uncertain environment, European Journal of Operational Research 181 (2007) 224–238.
[14] G. Zäpfel, Production planning in the case of uncertain individual demand Extension for an MRP II concept, International Journal of Production Economics 119 (1996) 153–164.
[15] I. Yildrim, B. Tan, F. Karaesmen, A multiperiod fuzzy production planning and sourcing problem with service level constraints, OR Spectrum 27 (2005) 471–489.
[16] P. Kelly, G. Clendenen, P. Dardeau, Economic lot scheduling heuristic for random demand, International Journal of Production Economics 35 (1994) 337–342.
[17] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[18] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3–28.
[19] D. Dubois, H. Prade, Possibility Theory, Plenum Press, New York, 1988.
[20] S. Nahmias, Fuzzy variables, Fuzzy Sets and Systems 1 (1978) 97–110.
[21] H. Tanaka, P. Guo, H.J. Zimmermann, Possibility distribution of fuzzy decision variables obtained from possibilistic linear programming problems, Fuzzy Sets and Systems 113 (2000) 323–332.
[22] L.H. Shih, Cement transportation planning via fuzzy linear programming, International Journal of Production Economics 58 (1999) 277–287.
[23] B. Liu, Y.K. Liu, Expected value of fuzzy variable and fuzzy expected value models, IEEE Transactions on Fuzzy Systems 10 (2002) 445–450.
[24] B. Liu, Uncertain Theory: An Introduction to its Axiomatic Foundations, Springer-Verlag, Berlin, 2004.
[25] B. Liu, A survey of credibility theory, Fuzzy Optimization and Decision 5 (2006) 387–408.
[26] Y.K. Liu, Convergent results about the use of fuzzy simulation in fuzzy optimization problems, IEEE Transactions on Fuzzy Systems 14 (2006) 295–304.
[27] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, 1995, pp. 1942–1948.
[28] J. Kennedy, R.C. Eberhart, Y. Shi, Swarm Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, 2001.
[29] Y.K. Liu, J.W. Gao, The independence of fuzzy variables with applications to fuzzy random optimization, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 15 (2007) 1–20.
[30] Y. Dong, J. Tang, B. Xu, D. Wang, An application of swarm optimization to nonlinear programming, Computers & Mathematics with Applications 49 (2005) 1655–1666.