ISSN 1746-7659, England, UK
Journal of Information and Computer Science
Vol. 1, No. 4, 2006, pp. 235-244.

A Hybrid Intelligent Algorithm for Fuzzy Dynamic Inventory Problem

Jinwu Gao[1]*, Xueqin Feng[2]
[1] School of Information, Renmin University of China, Beijing 100872, China
[2] College of Economics, Hebei University, Baoding 071002, Hebei, China

(*Received Dec. 20, 2005, Accepted April 1, 2006*)

Abstract. In this paper, a fuzzy inventory problem with multiple commodities is casted into a dynamic programming model with continuous state space and decision space. In order to solve the dynamic programming model, genetic algorithms are used to get samples of the optimal cost functions, and then neural networks are trained to approximate the optimal cost function on a randomly generated sample set, which may bypass "the curse of dimensionality". A hybrid intelligent algorithm is thus produced to get the optimal cost functions functions that represented by neural networks. Lastly, a numerical example is given for illustrating purpose

Keywords: Fuzzy variable, inventory, dynamic programming, neural network, genetic algorithm

1 Introduction

In the past three decades, fuzzy inventory systems have received more and more attention with the development of fuzzy set theory. In literature, EOQ models with fuzzy parameters were discussed by many researchers (e.g. Chen *et al.* [5] Park [24], Roy and Maiti [25] [26], and Yao and Lee [32]. Fuzzy multi-stage inventory problems were also considered by some researchers (e.g., Kacprzyk and Staniewski [14], Roy and Maiti [27], Liu [19]). As far as I know, high dimensional dynamic inventory system with varying fuzzy demands is a new and challenging work.

This paper investigates the high dimensional dynamic inventory system with varying fuzzy demands. When the decision criterion is to minimize the fuzzy expected value of the total cost incurred over the horizon, this problem is formulated as a dynamic programming model with continuous state and decision space. Traditional method for solving this model is to discretize the continuous state components uniformly so that the optimal cost functions that characterize the solutions need only be solved over a finite number values of the state vector (grid points). But the use of a full tensor-product grid may lead to "the curse of dimensionality", which limit the practical applications of dynamic programming to low dimensional problems.

In order to deal with the curse of dimensionality, we use a neuro-dynamic programming approach whose main theme is the use neural networks to approximate the optimal cost functions and then to guide decision-making. It outgoes the traditional methods in the following two aspects: (i) randomly generated samples versus uniform grid points for tensor product; and (ii) compact representation versus grid points with interpolation representation of the optimal cost function. And this methodology has significant potential as a general approach to approximately solve a wide variety of complex Markov decision problems. For more expositions to neuro-dynamic programming with its successful applications, the reader may consult the book by Bertsekas and Tsitsiklis [2] and the literature [3][6][23][28].

Corresponding author. Tel: +86-10-82500690.
E-mail address: jgao@orsc.edu.cn (J. Gao).

This paper is arranged as follows. Section 2 recalls the definition of the expected value operator of fuzzy variables. Section 3 describes and formulates a multi-commodity inventory system with varying fuzzy demands as problem as a dynamic programming model with continuous state and solution spaces. Then Section 4 discusses how to approximate the Q-factors, how to compute the optimal cost for a given state, and how to approximate the optimal cost functions, thus producing a hybrid intelligent algorithm for solving the fuzzy dynamic programming model. Lastly, a numerical algorithm is presented for illustrating the effectiveness of the hybrid intelligent algorithm.

2 An Introduction to Credibility Theory

Since its introduction in 1965 by Zadeh [33], fuzzy set theory has been well developed and applied in a wide variety of real problems. Possibility theory was proposed by Zadeh [34] in 1978, and developed by many researchers such as Dubois and Prade [8][9]. Although possibility measure has been widely used, it has no self-duality property. However, a self-dual measure like probability is absolutely needed in both theory and practice. The the expected value of a fuzzy variable has been defined in many ways. Dubois and Prade [7] defined the expected value operator as an interval that applicable to only upper-semi-continuous fuzzy numbers. Heilpern [12] defined the expected value operator via a random set that is applicable to continuous fuzzy numbers, Yager [30][31] defined a expected value operator that is applicable to discrete fuzzy variables. That is, an general expected value operator that applicable to both continuous fuzzy variables and discrete fuzzy variables is also absolutely needed in both theory and practice.

Let ξ be a fuzzy variable with membership function μ. In order to deal with fuzzy event, Liu and Liu [17] gave the concept of credibility measure by

$$\mathrm{Cr}\{\xi \in B\} = \frac{1}{2}\left(\sup_{x \in B} \mu(x) + 1 - \sup_{x \in B^c} \mu(x)\right)$$

for any set B of real numbers. Conversely, if ξ is a fuzzy variable, then its membership function is derived from the credibility measure by

$$\mu(x) = (2\mathrm{Cr}\{\xi = x\}) \wedge 1,\ x \in R.$$

It is obvious that the credibility measure is self dual, i.e., $\mathrm{Cr}\{A\} + \mathrm{Cr}\{A^c\} = 1$ for any $A \in \mathcal{P}(\Theta)$.

Remark 1 *(Gao and Liu [10]) A fuzzy event may fail even though its possibility achieves* 1*, and hold even though its necessity is* 0*. However, the fuzzy event must hold if its credibility is* 1*, and fail if its credibility is* 0*.*

Based on the credibility measure, we have the expected value operator as follows.

Definition 1 (Liu and Liu [17]) *Let ξ be a fuzzy variable. The expected value of ξ is defined as*

$$\boldsymbol{E}[\xi] = \int_0^{\infty} \mathrm{Cr}\{\xi \geq r\}\mathrm{d}r - \int_{-\infty}^{0} \mathrm{Cr}\{\xi \leq r\}\mathrm{d}r, \tag{1}$$

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

Let ξ and η be independent fuzzy variables. Then for any real numbers a and b, we have $\boldsymbol{E}[a\xi + b\eta] = a\boldsymbol{E}[\xi] + b\boldsymbol{E}[\eta]$.

Example 1 *The expected value of a triangular fuzzy variable (r_1, r_2, r_3) is*

$$E[\xi] = \frac{1}{4}(r_1 + 2r_2 + r_3).$$

Example 2 *The definition of expected value operator is not only applicable to continuous case but also discrete case. Assume that ξ is a discrete fuzzy variable whose membership function is given by*

$$\mu(x) = \begin{cases} \mu_1, & \text{if } x = a_1 \\ \mu_2, & \text{if } x = a_2 \\ \cdots & \\ \mu_m, & \text{if } x = a_m. \end{cases}$$

Without loss of generality, we also assume that $a_1 \le a_2 \le \cdots \le a_m$. Definition 1 implies that the expected value of ξ is

$$E[\xi] = \sum_{i=1}^{m} w_i a_i \tag{2}$$

where the weights $w_i, i = 1, 2, \cdots, m$ are given by

$$\begin{aligned} w_1 &= \frac{1}{2}\left(\mu_1 + \max_{1\le j\le m} \mu_j - \max_{1<j\le m} \mu_j\right), \\ w_i &= \frac{1}{2}\left(\max_{1\le j\le i} \mu_j - \max_{1\le j<i} \mu_j + \max_{i\le j\le m} \mu_j - \max_{i<j\le m} \mu_j\right), \quad 2 \le i \le m-1 \\ w_m &= \frac{1}{2}\left(\max_{1\le j\le m} \mu_j - \max_{1\le j<m} \mu_j + \mu_m\right). \end{aligned}$$

It is easy to verify that all $w_i \ge 0$ and

$$\sum_{i=1}^{m} w_i = \max_{1\le j\le m} \mu_j = 1$$

since any fuzzy variables defined on a possibility space are normalized.

For more detailed expositions on the credibility theory, the reader may consult the book [21][22].

3 Problem Description and Formulation

Multi-item dynamic inventory system with stochastic demands has been discussed by many researchers such as Johnson [13] and Veinott [29]. Here, we discuss a multi-item dynamic inventory system with varying fuzzy demands with the assumption that the system is reviewed periodically and decisions are made at the beginning of each stage. First of all, we give some notations as follows.

N : problem horizon which is divided into N stages;

θ : discount rate;

$\boldsymbol{x}$: state vector;

$\boldsymbol{d}$: order quantity;

$\boldsymbol{\xi}_n$: fuzzy demands vector at stage n;

$c_n(\boldsymbol{d})$: ordering cost function at stage n;

$l_n(\boldsymbol{x})$: inventory cost function at stage n.

Let $\widehat{\boldsymbol{x}} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$ be a sequence of state vectors, and $\widehat{\boldsymbol{d}} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_N)$ be the corresponding decision sequence. That is, the inventory level at stage n is $\boldsymbol{x}_n$, and if we choose a decision (ordering quantity) $\boldsymbol{d}_n$, then we move to stage $n+1$ with a fuzzy state $x_{n+1} = \boldsymbol{x}_n + \boldsymbol{d}_n - \boldsymbol{\xi}_n$. As a result, incurred the immediate costs including a linear ordering cost $\boldsymbol{c}_n \cdot \boldsymbol{d}_n$ and an fuzzy inventory cost $l_n(\boldsymbol{x}_n + \boldsymbol{d}_n - \boldsymbol{\xi}_n)$. Simultaneously, the future decisions as well as future costs, which is fuzzy due to the fuzzy state $\boldsymbol{x}_{n+1}$, are also affected. So we must take into account both the immediate and future costs. Then the objective function at stage n is as follows,

$$J_n(\widehat{\boldsymbol{x}}; \widehat{\boldsymbol{d}}) = \sum_{i=n}^{N} \theta^{i-n} \boldsymbol{E}[c_i(\boldsymbol{d}_i) + l_i(\boldsymbol{x}_i + \boldsymbol{d}_i - \boldsymbol{\xi}_i)] \tag{3}$$

with the state transition equation

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{d}_n - \boldsymbol{\xi}_n. \tag{4}$$

By Bellman's principle of optimality, we can formulate the inventory system as the following fuzzy dynamic programming model,

$$\begin{cases} f_N(x) = \min\limits_{\boldsymbol{d} \geq 0} \{c_N(\boldsymbol{d}) + \boldsymbol{E}\left[l_N(\boldsymbol{x} + \boldsymbol{d} - \boldsymbol{\xi}_N)\right]\} \\ f_n(\boldsymbol{x}) = \min\limits_{\boldsymbol{d} \geq 0} \{c_n(\boldsymbol{d}) + \boldsymbol{E}\left[l_n(\boldsymbol{x} + \boldsymbol{d} - \boldsymbol{\xi}_n)\right] + \theta \boldsymbol{E}\left[f_{n+1}(\boldsymbol{x} + \boldsymbol{d} - \boldsymbol{\xi}_n)\right]\} \\ \qquad n = 1, 2, \cdots, N-1. \end{cases} \tag{5}$$

Let

$$F_n(\boldsymbol{y}) = \boldsymbol{c}_n \cdot \boldsymbol{y} + \boldsymbol{E}[l_n(\boldsymbol{y} - \boldsymbol{\xi}_n)] + \theta \boldsymbol{E}[f_{n+1}(\boldsymbol{y} - \boldsymbol{\xi}_n)], \tag{6}$$

for $n = 1, 2, \cdots, N$, and $f_{N+1}(\boldsymbol{y}) = 0$. Then we have

$$f_n(\boldsymbol{x}) = \min_{\boldsymbol{y} \geq \boldsymbol{x}} F_n(\boldsymbol{y}) - \boldsymbol{c}_n \cdot \boldsymbol{x}. \tag{7}$$

Now, we assume that $\boldsymbol{\xi}_{i,n-1}$ are triangular fuzzy number $(a_{i,n-1}, b_{i,n-1}, c_{i,n-1})$ with $c_{i,n} \geq c_{i,n-1} - a_{i,n-1}$ for all i and n. Then we may restrict $\boldsymbol{x}_{i,n-1} + \boldsymbol{d}_{i,n-1}$ to the interval $[a_{n-1}, c_{n-1}]$, otherwise, it would cause more inventory cost or shortage cost. That is, we have

$$\boldsymbol{y}_{i,n-1} = \boldsymbol{x}_{i,n-1} + \boldsymbol{d}_{i,n-1} \in [a_{i,n-1}, c_{i,n-1}]$$

and

$$\boldsymbol{x}_{i,n} = \boldsymbol{x}_{i,n-1} + \boldsymbol{d}_{i,n-1} - \boldsymbol{\xi}_{i,n-1} \in [a_{i,n-1} - c_{i,n-1}, c_{i,n-1} - a_{i,n-1}].$$

In brief, we can restrict the state $\boldsymbol{x}$ to a hypercube Π_n which is determined by the former stage $n-1$, and $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{d}$ to a hypercube Ξ_n determined by the former and current stage. In view of this, we need only to solve the model

$$\begin{cases} f_n(\boldsymbol{x}) = \min\limits_{\boldsymbol{y} \geq \boldsymbol{x}} F_n(y) - \boldsymbol{c}_n \cdot \boldsymbol{x} \\ \boldsymbol{x} \in \Pi_n, \boldsymbol{y} \in \Xi_n. \end{cases} \tag{8}$$

4 Hybrid intelligent algorithm

For practical use of the fuzzy dynamic programming model, we design a hybrid intelligent algorithm with reasonable computational cost by using neural network and genetic algorithm.

4.1 Fuzzy Simulation

Fuzzy simulation technique was proposed by Liu and Iwamura [16][17], and the reader may consult the book by Liu [21][22]. We know, the function $F_n(\boldsymbol{y})$ involves uncertain functions like $\boldsymbol{E}[g(\boldsymbol{y}-\boldsymbol{\xi})]$. Due to complexity, we design a fuzzy simulation procedure for computing the uncertain functions.

Fuzzy Simulation for Cr $\{g(\boldsymbol{y}-\boldsymbol{\xi}_n)\le 0\}$

Step 1. Randomly generate u_{ink} from the ε-level set of $\boldsymbol{\xi}_{in}$, $i=1,2,\cdots,m$, respectively, where $k=1,2,\cdots,M$ and ε is a sufficiently small positive number.

Step 2. Set $\nu_k=\min_i\{\mu_{\xi_{in}}(u_{ink})\}$.

Step 3. Return L via the following estimation formula

$$L=\frac{1}{2}\left(\max_{1\le k\le N}\left\{\nu_k\mid g(\boldsymbol{y}-\boldsymbol{u}_{nk})\le 0\right\}+\min_{1\le k\le N}\left\{1-\nu_k\mid g(\boldsymbol{y}-\boldsymbol{u}_{nk})>0\right\}\right)$$

where $\boldsymbol{u}_{nk}=(u_{1nk},u_{2nk},\cdots,u_{mnk})$.

Fuzzy Simulation for Expected Value:

Step 1. Set $e=0$.

Step 2. Randomly generate u_{ink} from the ε-level set of $\boldsymbol{\xi}_{in}$, $i=1,2,\cdots,m$, respectively, where $k=1,2,\cdots,M$ and ε is a sufficiently small positive number.

Step 3. Set $\nu_k=\min_i\{\mu_{\xi_{in}}(u_{ink})\}$.

Step 4. Randomly generate b from $[b_1,b_2]$.

Step 5. If $b\ge 0$, then $e\leftarrow e+\mathrm{Cr}\{g(\boldsymbol{y}-\boldsymbol{\xi}_n)\le b\}$.

Step 6. If $b<0$, then $e\leftarrow e-\mathrm{Cr}\{g(\boldsymbol{y}-\boldsymbol{\xi}_n)\ge b\}$.

Step 7. Repeat the fourth to sixth steps for N times.

Step 8. Return $\boldsymbol{E}\left[g(\boldsymbol{y}-\boldsymbol{\xi}_n)\right]=b_1\vee 0+b_2\wedge 0+e\cdot(b_2-b_1)/N$.

4.2 Approximating $F_n(\boldsymbol{y})$ by Neural Networks

Neural network is well-known as a universal approximator whose input-output mapping is matched to an unknown nonlinear mapping. In literature [15], Leshno *et al.* gave a result that multi-layer feedforward network with a non-polynomial activation can approximate any nonlinear continuous function arbitrarily well over a closed bounded set. So do the optimal cost function provided that it is continuous over a closed bounded set. Inspired by this, we first use fuzzy simulation to generate samples and then use the backpropagation algorithm to train neural networks for approximating functions $F_n(\boldsymbol{y})$.

A feedforward neural networks is essentially a nonlinear mapping from the input space to the output space. Assume that the mapping is characterized by $U(\boldsymbol{y},\boldsymbol{w})$ where $\boldsymbol{w}$ denotes the network weights. A training process on a set of input-output data $\left\{(\boldsymbol{y}^{(k)},z^{(k)})\mid k=1,2,\cdots,M\right\}$ is to find the best weight vector that minimizes the following error function

$$Sum\_Squared\_Err(\boldsymbol{w})=\frac{1}{2}\sum_{k=1}^{M}|U(\boldsymbol{y}^{(k)},\boldsymbol{w})-z^{(k)}|^2. \tag{9}$$

In this paper, the popular backpropagation algorithm is employed as the learning algorithm, and the average error

$$Average\_Err(\boldsymbol{w}) = \frac{1}{M}\sum_{k=1}^{M} |U(\boldsymbol{y}^{(k)}, \boldsymbol{w}) - z^{(k)}|. \tag{10}$$

is also used to demonstrate the accuracy of the trained neural networks. For detailed discussion on uncertain function approximation, the reader may consult Chapter 3 in the book [21] by Liu.

4.3 Compute $f_n(\boldsymbol{x})$ for Given $\boldsymbol{x}$ by Genetic Algorithm

For each given state $\boldsymbol{x}$, in order compute the optimal cost with respect to it, we must solve the following optimization problem

$$f_n(\boldsymbol{x}) = \min_{\boldsymbol{y}\geq\boldsymbol{x}} F_n(y) - \boldsymbol{c}_n \cdot \boldsymbol{x}, \boldsymbol{x} \in \Pi_n, \boldsymbol{y} \in \Xi_n.$$

Now, we give a genetic algorithm procedure for solving the above optimization problem.

Genetic Algorithm Procedure for Optimal Cost:

Step 1. Initialize pop_size chromosomes randomly.

Step 2. Update the chromosomes by crossover and mutation operations.

Step 3. Calculate the objective values for all chromosomes.

Step 4. Compute the fitness of each chromosome according to the objective values.

Step 5. Select the chromosomes by spinning the roulette wheel.

Step 6. Repeat the second to fifth steps for a given number of cycles.

Step 7. Report the best chromosome as the optimal cost for the given state $\boldsymbol{x}$.

We note that, in the genetic algorithm procedure, the trained network representing $F(\boldsymbol{y})$ substitutes the work of simulation. Thus much reduce the computation cost in the computing procedure.

4.4 Approximating $f_n(\boldsymbol{x})$ by Neural Networks

In general, the function $f_n(\boldsymbol{x})$ is very complex, and it is impossible for us to get the analytical properties of it. However, we can employ neural network to approximate the continuous function over a bounded hypercube. Firstly, we generate a set of samples by employing the genetic algorithm procedure. And then, we can also train a feed-forward neural network on the set of samples to approximate the function $f_n(\boldsymbol{x})$. The procedure is also omitted here.

4.5 Hybrid Intelligent Algorithm

In order to solve the Bellman's equation (7), the recurrence relation should be performed N times. We only take stage n as an example to illustrate how we get the neural networks that match to functions $F_n(\boldsymbol{y})$ and $f_n(\boldsymbol{x})$.

Firstly, generate a set of points uniformly from the hypercube Π_n, then for each $\boldsymbol{y}$ in the set, calculate $F_n(\boldsymbol{y})$ by stochastic simulation (here a neural network trained to match to the optimal function $f_{n+1}(x)$ is used, i.e., we get the sample set of the function $F_n(\boldsymbol{y})$. Secondly, use these samples as training data to train a neural network to approximate the complex function $F_n(\boldsymbol{y})$. Thirdly, generate a set of points uniformly from the hypercube Ξ_n, then for each $\boldsymbol{x}$ in the set, we embed the trained neural network into genetic algorithm to get the minimum discounted expected value $f_n(\boldsymbol{x})$ of all the cost incurred over the horizon. i.e., we get the sample set of the function $f_n(\boldsymbol{x})$. lastly, we use these samples as training data to train a neural network to approximate the complex optimal function $f_n(\boldsymbol{x})$.

The procedure of hybrid intelligent algorithm may be written as follows:

Hybrid intelligent algorithm:

Step 1. Initialize a neural network represent $f_{N+1}(\boldsymbol{x}) = 0$.

step 2. Set $n \leftarrow N$.

Step 3. Generate a set of points uniformly from the hypercube Π_n, then for each $\boldsymbol{y}$ in the set, use stochastic simulations to calculate the value of $F_n(\boldsymbol{y})$.

Step 4. Train a neural network to approximate the uncertain function $F_n(\boldsymbol{y})$ according to the generated training data.

Step 5. Generate a set of points uniformly from the hypercube Ξ_n, then embed the trained neural network to genetic algorithm to get the value of $f_n(\boldsymbol{x})$ for each $\boldsymbol{x}$ in the set.

Step 6. Train neural networks to approximate the optimal cost function $f_n(\boldsymbol{y})$ according to the generated training data.

Step 7. Report the neural network that matches to the optimal function $f_n(\boldsymbol{x})$.

Step 8. Set $n \leftarrow (n-1)$, repeat the third to seventh steps until $n = 0$.

5 A Numerical Example

The computer code for the HIA has been written in C language. In order to illustrate its effectiveness, we provide one numerical example performed on a personal computer.

Example. Consider a dynamic inventory system with 6 commodities, and the problem horizon 4 stages. The ordering cost function at all stage is the same and given as follows

$$c(\boldsymbol{d}) = \sum_{i=1}^{m} s_i d_i,$$

where $s1_i$ is randomly generated from interval $[4, 6]$ and given in Table 1. The inventory cost function at all stage is the same and given as follows

$$l(\boldsymbol{x}) = \sum_{i=1}^{m} l_i(x_i)$$

where

$$l_i(x_i) = \begin{cases} h_i\sqrt{x_i}, & x_i \geq 0 \\ p_i x_i^2, & x_i < 0 \end{cases}$$

where h_i and p_i are randomly generated from interval $[3, 7]$ and $[6, 10]$, respectively and given in Table 1. The fuzzy demands at all stages are given in Table 2.

After a run of the hybrid intelligent algorithm, we get the optimal cost function $f_n(\boldsymbol{x})$, $n = 1, 2, 3, 4$ represented by neural networks.

Table 1: parameters s_i, h_i and p_i

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
s_i	4.0025	5.1271	4.3866	5.6174	5.1700
h_i	4.4011	6.5838	6.2913	5.9864	3.6964
p_i	8.8420	8.0541	7.2159	6.0599	6.3656

Table 2: fuzzy demands ξ_{in}

	$n = 1$	$n = 2$	$n = 3$	$n = 4$
$i = 1$	$(1.473, 11.658, 29.885)$	$(3.521, 10.570, 26.076)$	$(8.622, 12.096, 27.796)$	$(0.562, 10.087, 29.187)$
$i = 2$	$(4.456, 11.190, 20.046)$	$(7.833, 18.026, 25.198)$	$(8.436, 19.967, 29.996)$	$(2.758, 12.728, 25.879)$
$i = 3$	$(0.089, 13.778, 25.316)$	$(3.019, 18.759, 27.266)$	$(6.114, 13.924, 22.662)$	$(6.911, 18.376, 27.264)$
$i = 4$	$(5.711, 16.017, 26.071)$	$(9.559, 19.257, 25.393)$	$(2.972, 18.401, 20.237)$	$(4.849, 12.053, 27.437)$
$i = 5$	$(1.662, 16.630, 24.507)$	$(1.423, 14.620, 22.353)$	$(3.758, 10.926, 26.772)$	$(4.684, 14.579, 29.491)$

6 Conclusion

This paper contributes to the area of inventory by presenting a new fuzzy dynamic programming model, and a hybrid intelligent algorithm which may bypass the curse of dimensionality .

Acknowledgments

This work was supported by National Natural Science Foundation of China Grant No.70601034, and Research Foundation of Renmin University of China.

References

[1] Bellman R.E., *Dynamic Programming*, Princeton University Press, New Jersey, 1957.

[2] Bertsekas D.P. and Tsitsiklis J.N., *Neuro-Dynamic Programming*, Athena Scientific, Bellmont, MA, 1996.

[3] Bertsekas D.P., Homer M.L., Logan D.A., Patek S.D. and Sandell N.R., Missile defense and interceptor allocation by neuro-dynamic programming *IEEE Transactions on Systems, Man and Cybernetics–Part A: Systems and Humans*, vol.30, no.1, pp.42-51, 2000.

[4] Campos, L., and González, A., A subjective approach for ranking fuzzy numbers, *Fuzzy Sets and Systems*, Vol. 29, 145-153, 1989.

[5] Chen S.H., Wang C.V. and Ramer A., Backorder fuzzy inventory model under function principle, *Information Science*, Vol. 95, pp.71–79, 1996.

[6] Crites R.H. and Barto A.G., Improving elevator performance using reinforcement learning, *Advances in Neural Information Processing Systems 8*, Touretzky D.S., Mozer M.E. and Hasselmo M.E., eds., MIT Press, Cambridge, MA, 1993.

[7] Dubois, D., and Prade, H., The mean value of a fuzzy number, *Fuzzy Sets and Systems*, Vol. 24, 279-300, 1987.

[8] Dubois, D., and Prade, H., *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum, New York, 1988.

[9] Dubois, D., and Prade, H., Fuzzy numbers: An overview, in *Analysis of Fuzzy Information*, Vol. 2, 3-39, Bezdek, J.C. (Ed.), CRC Press, Boca Raton, 1988.

[10] Gao J. and Liu B., New Primitive Chance Measures of Fuzzy Random Event, *International Journal of Fuzzy Systems*, Vol.3, No.4, 527-531, 2001.

[11] González, A., A study of the ranking function approach through mean values, *Fuzzy Sets and Systems*, Vol. 35, 29-41, 1990.

[12] Heilpern, S., The expected value of a fuzzy number, *Fuzzy Sets and Systems*, Vol. 47, 81-86, 1992.

[13] Johnson E.L., Optimality and computation of (σ, S) policies in the multi-item infinite horizon inventory problem, *Management Science*, Vol.13, No.7, pp. 475–491, 1967.

[14] Kacprzyk J. and Staniewski P., Long-term inventory policy-making through fuzzy decision-making models, *Fuzzy Sets & Systems*, vol.8, no.2, pp.117-32, 1982.

[15] Leshno M., Ya V. and Schochen S., Multilayer feedforward networks with a nonpolynomial activation can approximate any functions, *Neural Networks*, vol.6, pp.861-867, 1993.

[16] Liu B. and Iwamura K., Chance constrained programming with fuzzy parameters, *Fuzzy Sets and Systems*, Vol.94, No.2, 227-237, 1998.

[17] Liu B., and Liu Y.-K., Expected value of fuzzy variable and fuzzy expected value models, *IEEE Transactions on Fuzzy Systems*, Vol.10, No.4, 445-450, 2002.

[18] Liu Y.-K., and Liu B., Expected value operator of random fuzzy variable and random fuzzy expected value models, *Technical Report,* 2001. Vol.11, No.2, 195-215, 2003.

[19] Liu B., Fuzzy criterion models for inventory systems with partial backorders, *Annals of Operations Research*, Vol. 87, pp. 117–126, 1999.

[20] Liu B. and Esogbue A.O., *Decision Criteria and Optimal Inventory Processes,* Kluwer Academic Publishers, Boston, 1999.

[21] Liu B., *Theory and Practice of Uncertain Programming*, Physica-Verlag, Heidelberg, 2002.

[22] Liu B., *Uncertainty Theory: An Introduction to its Axiomatic Foundations*, Springer-Verlag, Berlin, 2004.

[23] Marbach P., Mihatsch O., and Tsitsiklis J.N., Call admission control and routing in integrated service networks using neuro-dynamic programming, *IEEE Journal on Selected Areas in Communications*, vol.18, no.2, pp.197-208, 2000.

[24] Park K.S., Fuzzy-set theoretic interpretation of economic order quantity, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-17, No. 6, pp. 1082–1084, 1987.

[25] Roy TK. and Maiti M., A fuzzy inventory model with constraint, *Opsearch*, vol.32, no.4, pp.287-98, 1995.

[26] Roy T.K. and Maiti M., Multi-objective inventory models of deteriorating items with some constraints in a fuzzy environment, *Computers & Operations Research*, vol.25, no.12, pp.1085-95, 1998.

[27] Roy T.K. and Maiti M., A multi-period inventory model with fuzzy cost and fuzzy demand, *Advances in Modeling & Analysis B,* vol.39, no.2, pp.25-38, 1998.

[28] Secomandi N., Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands, *Computer & Operations Research*, vol.27, pp.1201-1225, 2000

[29] Veinott A.F.Jr., Optimal policy for a multi-product, dynamic, nonstationary inventory problem, *Management Science*, Vol.12, No.3, pp.206-222, 1965.

[30] Yager, R.R., A procedure for ordering fuzzy subsets of the unit interval, *Information Sciences*, Vol. 24, 143-161, 1981.

[31] Yager, R.R., On the evaluation of uncertain courses of action, *Fuzzy Optimization and Decision Making*, Vol. 1, 13-41, 2002.

[32] Yao J.-S. and Lee H.-M., Fuzzy inventory with or without backoder for fuzzy order quantity with trapezoid fuzzy numbers , *Fuzzy Sets and Systems*, Vol. 105, pp. 311–337, 1999.

[33] Zadeh, L.A., Fuzzy sets, *Information and Control*, Vol. 8, 338-353, 1965.

[34] Zadeh, L.A., Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets and Systems*, Vol. 1, 3-28, 1978.