First Joint Congress on Fuzzy and Intelligent Systems
Ferdowsi University of Mashhad, Iran
29-31 Aug 2007

Intelligent Systems Scientific Society of Iran

A NEURAL NETWORK MODEL FOR SOLVING STOCHASTIC FUZZY MULTIOBJECTIVE LINEAR FRACTIONAL PROGRAMS

S. EFFATI [1], H. SADOGHI[2], Z. SABERI [1]

Abstract. The paper deals with stochastic fuzzy multiobjectives linear fractional programs. It is transformed to its equivalent deterministic crisp multiobjective linear program by using a modified possibility programming approach. Then is converted to a neural network model. Our linear neural network is able to generated optimal solutions. We solve neural network model with one of numerical method.Finally, simple numerical examples are provided for the sake of illustration.

1. Introduction.

The chance-constrained approach and the possibility programming technique, that has been stated by Negi and Lee [1] and modified by Iskander [2], are utilized to transform the suggested program to its equivalent deterministic-crisp multiobjective linear program in the case of exceedance possibility or the case of strict exceedance possibility[3]. In this paper we use neural network for solving the stochastic fuzzy multiobjective linear fractional program . Recently, several new dynamic solvers using artificial neural network models have been developed .See e.g.Tank and Hopfield (1985), Kennedy and Chua (1987), Rodriguez-Vazquez et al(1990), Wu et al (1996), Xia et al(2002), Effati(2006).

In the present paper, one neural network model for a stochastic fuzzy multiobjective linear fractional program is converged to real solutions.

2. Problem Formulation

In general, consider a stochastic fuzzy multiobjective linear fractional programming problem of the following form:

$$\text{(2.1)} \qquad \text{Maximize} \qquad \frac{c_{r1}x_1 + \ldots + c_{rn}x_n + c_{r0}}{d_{r1}x_1 \ldots + d_{rn}x_n + d_{r0}}, \; r = 1, \ldots, p,$$

Key words and phrases. Possibility programming,Chance-constrained,Stochastic fuzzy multiobjective linear fractional programs, Neural networks.

subject to:

$$\sum_{j=1}^{n} a_{ij} xj \leq b_i, \quad i = 1, \ldots, m, \tag{2.2}$$

$$x_j \geq \ 0 \ j = 1, \ldots, n \tag{2.3}$$

Where $x_j, j = 1, \ldots, n$ are nonnegative decision variables, c_{rj} and d_{rj}, $j = 1 \ldots, n$ are fuzzy coefficients, while c_{r0} and d_{r0} are two fuzzy scalars, for the r th linear fractional objective function,$r = 1, \ldots, p$, and p is the number of the distinct fuzzy linear fractional objective functions, b_i, $i = 1, \ldots, m$ are independent random variables with known distribution functions, while a_{ij} represents the fuzzy coefficient of the$j$ th decision variable in the i th stochastic constraint. Thus, by incorporating predetermined tolerance measures δ_i, $0 \leq \delta_i \leq 1$, $i = 1, \ldots, m$, and by utilizing the chance-constrained approach, the set of stochastic fuzzy constraints (2.2) can be transformed to their deterministic fuzzy equivalents as follows.

$$Pr(\sum_{j=1}^{n} a_{ij} xj \leq b_i) \geq \delta_i, \quad i = 1, \ldots, m, \tag{2.4}$$

then,

$$\sum_{j=1}^{n} a_{ij} xj \leq F_i^{-1}(\beta_i), \quad i = 1 \ldots, m, \tag{2.5}$$

where $\beta_i = 1 - \delta_i$ and $F_i^{-1}(.)$ is the inverse distribution function of the random variable b_i, i = 1,. . . ,m;

that is equivalent to the model (2.1)-(2.3) can be presented as:

$$\text{Maximize} \qquad \sum_{j=1}^{n} c_{rj} yj + c_{r0} t, \quad r = 1, \ldots, p, \tag{2.6}$$

subject to:

$$\sum_{j=1}^{n} a_{ij} \ yj - F_i^{-1}(\beta_i) t \leq \ 0 \ \imath = 1 \ldots, m, \tag{2.7}$$

$$\sum_{j=1}^{n} d_{rj} \ yj + d_{r0} t = 1, \qquad r = 1 \ldots, p, \tag{2.8}$$

$$t, y_j \geq \ 0, \qquad \jmath = 1 \ldots, n, \tag{2.9}$$

where $y_j = tx_j, j = 1 \ldots, n$, and $t = 1/(d_{r1}x_l + \ldots + d_{rn}x_n + d_{r0})$.
In the next section, the crisp equivalent of this model is presented.

2.1. If $\tilde{c}$,$\tilde{d}$, and$\tilde{a}$ Are Considered Trapezoidal Fuzzy Numbers.

let $\tilde{c}_{rj} = (\underline{c}_{rj}, c_{rjl}, c_{rj2}, \bar{c}_{rj}), \tilde{d}_{rj} = (\underline{d}_{rj}, d_{rjl}, d_{rj2}, \bar{d}_{rj})$, and $\tilde{a}_{ij} = (\underline{a}_{rj}, a_{ijl}, a_{ij2}, \bar{a}_{ij},$ where $\tilde{c}_{rj}$ and $\tilde{d}_{rj}$represent the model's coefficients and scalars, i.e., for $j = 0, 1, \ldots, n$ Then, the deterministic-crisp multi objective linear programming model can be formulated whether in the case of exceedance possibility .

consider following problem[3]:

(2.10)

$$\text{Maximize} \qquad \sum_{j=1}^{n}((1-\alpha)\bar{c_{rj}} + \alpha c_{rj2})y_j + ((1-\alpha)\bar{c_{r0}} + \alpha c_{r02})t, \quad r = 1, \ldots, p,$$

subject to:

$$\sum_{j=1}^{n}((1-\alpha)\underline{a}_{ij} + \alpha a_{ij1})y_j - F_i^{-1}(\beta_i)t \leq\ 0, \;\; i = 1\ldots, m, \tag{2.11}$$

$$\sum_{j=1}^{n}((1-\alpha)\underline{d}_{rj} + \alpha d_{rj1})y_j + ((1-\alpha)\underline{d}_{r0} + \alpha d_{r01})t \leq\ 1, \;\; r = 1, \ldots, p, \tag{2.12}$$

$$\sum_{j=1}^{n}((1-\alpha)\bar{d}_{rj} + \alpha d_{rj2})y_j + ((1-\alpha)\bar{d}_{r0} + \alpha d_{r02})t \geq\ 1, \;\; r = 1, \ldots, p, \tag{2.13}$$

$$t, y_j \geq\ 0, \qquad j = 1, \ldots, n, \tag{2.14}$$

3. Neural network model

Let us consider the following parametric programming problem[4]:

$$\begin{aligned} &\text{Minimize} \quad f(x,\alpha), \\ &\text{subject to: } h_i(x,\alpha) \leq 0,, \qquad i = 1 \cdots, m, \\ &\qquad\qquad \alpha \in [0,1], \end{aligned} \tag{3.1}$$

where x$\in \mathbb{R}^n, f(x,\alpha)$ and $h_i(x,\alpha)(i = 1, \ldots, m)$ are convex functions with respect to the first argument. It is also assumed that$f, h_i(i = 1, \ldots, m)$ are twice continuously differentiable, and for each $\alpha \in [0,1]$ problem (3.1) has feasible solution. In general, if $f(x,\alpha)$ is nonlinear and if the penalty method is applied to solve

(3.1), then we can obtain an unconstrained optimization problem:

$$\min_x P(x,\alpha) = f(x,\alpha) + \frac{k}{2}\sum_{i=1}^{m}(h_i^+(x,\alpha))^2 \tag{3.2}$$

where k is a positive number and
$h_i^+(x,\alpha) = \max_x\{0, h_i(x,\alpha)\}, (i = 1,\dots,m), \alpha \in [0,1]$is fixed.
Thus, the necessary condition for optimality of (3.2) for each $\alpha \in [0,1]$ is:

$$\frac{\partial P(x,\alpha)}{\partial x} = \frac{\partial f(x,\alpha)}{\partial x} + k\sum_{i=1}^{m} h_i^+(x,\alpha)\ \frac{\partial h_i(x,\alpha)}{\partial x} = 0, \tag{3.3}$$

We define the following neural network model:

$$x'(t,\alpha) = -\frac{\partial f(x(t),\alpha)}{\partial x} - k\sum_{i=1}^{m} h_i^+(x(t),\alpha)\ \frac{\partial h_i(x(t),\alpha)}{\partial x} \tag{3.4}$$

Proposition 1. If for any k (3.2) has an optimal solution, and if for system (3.4) we can find a state variable $x(t,\alpha)$ such that the neural network (3.4) is asymptotically stable at $x^*(\alpha)$, then the optimal solution to (3.2) will be the equilibrium state of (3.4).
Proof. The necessary condition for optimality of (3.2) is

$$\frac{\partial P(x,\alpha)}{\partial x} = \frac{\partial f(x,\alpha)}{\partial x} + k\sum_{i=1}^{m} h_i^+(x,\alpha)\ \frac{\partial h_i(x,\alpha)}{\partial x} = 0 \tag{3.5}$$

This is equivalent to

$$-\frac{\partial f(x,\alpha)}{\partial x} - k\sum_{i=1}^{m} h_i^+(x,\alpha)\ \frac{\partial h_i(x,\alpha)}{\partial x} = 0, \tag{3.6}$$

Eq. (3.4) is asymptotically stable for all $\alpha \in [0,1]$, then the equilibrium state $x^*(\alpha)$ satisfies (3.6). Thus, the optimal solution to (3.2) is the equilibrium state of (3.4). Under the combined action of both $h_i^+(x,\alpha)$ and $\frac{\partial h_i(x,\alpha)}{\partial x}$, the neural network (3.4) reach to its equilibrium. Thus, the mechanism of the optimization neural networks can be in general explained with parametric differential equations. Because of its specific roles, we refer to $-k\sum_{i=1}^{m} h_i{}^+(x,\alpha)\ \frac{\partial h_i(x,\alpha)}{\partial x}$ as the penalty method law. We show that under the penalty method,$P(x,\alpha)$ of (3.2) is also a Lyapunov function of (3.4).
Proposition 2. Under the penalty method, $P(x,\alpha)$of (3.2) is a Lyapunov function of system(3.4).

Proof. Taking the derivative of $P(x,\alpha)$ with respect to time t, we have:

$$\frac{\partial P(x(t),\alpha)}{\partial x} = \frac{\partial f^T(x(t),\alpha)}{\partial x} x'(t,\alpha) + k \sum_{i=1}^{m} h_i^+(x(t),\alpha) \frac{\partial h_i^T(x(t),\alpha)}{\partial x} x'(t,\alpha) = (x')^T(-x') = -x'^T x' \leq 0$$

From Proposition 2, it can be seen that under the penalty method, system (3.4) evolves along a direction such that the Lyapunov function $P(x,\alpha)$ decreases the most.

4. Numerical method

The Euler method is used for solving our neural network. We obtain state variable $x(t,\alpha)$ for the neural network (3.4) in r points $\alpha_j \in [0,1]$where $\alpha_j = \frac{j}{r}(j = 0,\ldots,r)$.

The following Matlab code describes the our neural network.

x_0: initial point;
n: Number of loop;
$dx = 0$;
$dt = 1/n$;
k: Suitable positive number;
$\alpha = 0 : \frac{j}{r} : 1$;
for $j = 1 : r$
for $i = 1 : n$
$h = -\frac{\partial f(x,\alpha_j)}{\partial x} - k \sum_{i=1}^{m} h_i^+(x,\alpha_j) \frac{\partial h_i(x,\alpha_j)}{\partial x}$
$dx = dth$;
$dx = \max(x + dx, 0) - x$;
$x = x + dx$;
end;
$X(j) = x$; $X(j)$ is optimal solution for α_j.
end.

Example 1.Consider the following stochastic fuzzy multiobjective linear fractional programming problems:

$$\text{Maximize } \frac{(4,7,10,12)x_1+(8,10,14,15)x_2+(2.5,4,7.5,11.5)x_3+(2,3,4,6)}{(10,14,20,22)x_1+(20,23.5,27.5,29)x_2+(18,20,25,28)x_3+(5,10,18,20)}$$

$$\text{Maximize } \frac{(20,24,28)x_1+(18,25,30)x_2+(14,19,25)x_3+(1,6,10)}{(14,16,19,23)x_1+(18,21,25,27)x_2+(15,20,25,30)x_3+(10,15,20,25)}$$

subject to

$$(10,17,19,25)x_1+(14,16,22,24)x_2+(20,25,27,30)x_3 \le b_1,$$
$$(0.03,0.07,0.09)x_1+(0.05,0.08,0.1)x_2+(0.02,0.06,0.07)x_3 \le b_2,$$
$$(4,6,10,13)x_1+(0,5,10,15)x_2+(8,11,14,20)x_3 \le b_3,$$
$$x_1,x_2,x_3 \ge 0. \tag{4.1}$$

Where b_l, b_2, and b_3 are independent random variables, with b_l normally distributed random variable having mean 50 and variance 36, b_2 an upper truncated exponential random variable whose values do not exceed 12 and its probability density function $f(b_2)$ is given by $f(b_2) = e^{-b_2}/(1-e^{-12})$, while b_3 is a uniformly distributed random variable on the interval [30, 40]. The decision maker's tolerance measures are $\delta_1 = 0.8, \delta_2 = 0.4$, and $\delta_3 = 0.5$. Thus, $F_1^{-1}(0.2) = 44.96, F_2^{-1}(0.6) = 0.9163, and F_3^{-1}(0.5) = 35$.

Therefore, the equivalent deterministic-crisp multiobjective linear programming model can be stated whether in the case of exceedance possibility.

THE CASE OF EXCEEDANCE POSSIBILITY.

According to this model , the equivalent model can be presented as:

$$\text{Maximize } (12-2\alpha)y_1+(15-\alpha)y_2+(11.5-4\alpha)y_3+(6-2\alpha)t,$$
$$\text{Maximize } (28-4\alpha)y_1+(30-5\alpha)y_2+(25-6\alpha)y_3+(10-4\alpha)t,$$

subject to

$$(10+7\alpha)y_1+(14+2\alpha)y_2+(20+5\alpha)y_3-44.96t \le 0,$$
$$(0.03+0.04\alpha)y_1+(0.05+0.03\alpha)y_2+(0.02+0.04\alpha)y_3-0.9163t \le 0,$$
$$(4+2\alpha)y_1+5\alpha y_2+(8+3\alpha)y_3-35t \le 0,$$
$$(10+4\alpha)y_1+(20+3.5\alpha)y_2+(18+2\alpha)y_3+(5+5\alpha)t \le 0,$$
$$(14+2\alpha)y_1+(18+3\alpha)y_2+(15+5\alpha)y_3+(10+5\alpha)t \le 0,$$
$$(22-2\alpha)y_1+(29-1.5\alpha)y_2+(28-3\alpha)y_3+(20-2\alpha)t \ge 0,$$
$$(23-4\alpha)y_1+(27-2\alpha)y_2+(30-5\alpha)y_3+(25-5\alpha)t \ge 0,$$
$$t, y_1, y_2, y_3 \ge 0$$

Table1. Results of solving by using global criterion method, with equal weights.

α	x_1	x_2	x_3	t	z
0.3	3.8	0	0	0.0152	78.1773
0.5	3.313	0	0	0.016	68.1603
0.8	2.8754	0	0	0.017	56.1937

Table2. Results of solving by using neural network model.

α	x_1	x_2	x_3	t	z
0.3	3.7876	0	0	0.0167	79.4441
0.5	3.s976	0	0	0.0175	69.3558
0.8	2.9427	0	0	0.0183	57.3908

References

[1] D.S. Negi and E.S. Lee, *Possibility programming by the comparison of fuzzy numbers,* Computers Math. Applic. 25 (9), 43-50, (1993).

[2] M.G. Iskander, *Comparison of fuzzy numbers using possibility programming: Comments and new concepts,* Computers Math. Applic. 43 (6/7), 833-840, (2002).

[3] M.G.Iskander, *A possibility programming approach for stochastic fuzzy multiobjective linear fractional programs*Computers and Mathematics with Applications 48, 1603-1609, (2004) .

[4] S. Effati and M.Jafarzadeh, *A new nonlinear neural network for solving a class of constrained parametric optimization problems*, Applied Mathematics And Computation. (2006)

[1] Department of Mathematics, Tarbiat Moallem University, Sabzevar, Iran.
E-mail address: effati@sttu.ac.ir z.saberi2007@yahoo.com

[2] Engineering Department, Tarbiat Moallem University, Sabzevar, Iran.
E-mail address: Sagoghi@sttu.ac.ir