6 Fuzzy Monte Carlo Method

6.1 Introduction

This chapter introduces our fuzzy Monte Carlo method. We will be working with a very simple linear programming problem. The crisp linear program is presented in the next section. Then we fuzzify the linear program in the third section. We make some of the parameters in the problem triangular fuzzy numbers and allow all the variables to be triangular shaped fuzzy numbers. We will need to decide on a definition of $\leq$ between fuzzy numbers and we will use Kerre's method (Section 2.6.2 of Chapter 2) first and then Chen's method (Section 2.6.3 of Chapter 2) second. This chapter, and Chapters 7 and 8, are based on ([5],[6]), see also ([3],[4]).

Fuzzy linear programming has become a very large area of research. Put "fuzzy linear programming" into your search engine and obtain over $17,000$ web sites to visit. Obviously we can not search all of these sites. A few recent references to this topic are the papers ([10]-[16],[18],[20],[22]-[25],[27]) and books (or articles in these books) ([1],[2],[7]-[9],[17],[19],[21],[26]).

6.2 Crisp Linear Program

Consider the optimization problem

$$\max Z = (2x_1 + 3x_2), \tag{6.1}$$

subject to

$$x_1 + 2x_2 \leq 6, \tag{6.2}$$

$$2x_1 + x_2 \leq 6, \tag{6.3}$$

$$0 \leq x_1, x_2 \leq M, \tag{6.4}$$

J.J. Buckley et al.: Monte Carlo Meth. in Fuzzy Optimization, STUDFUZZ 222, pp. 57–65, 2008.
springerlink.com

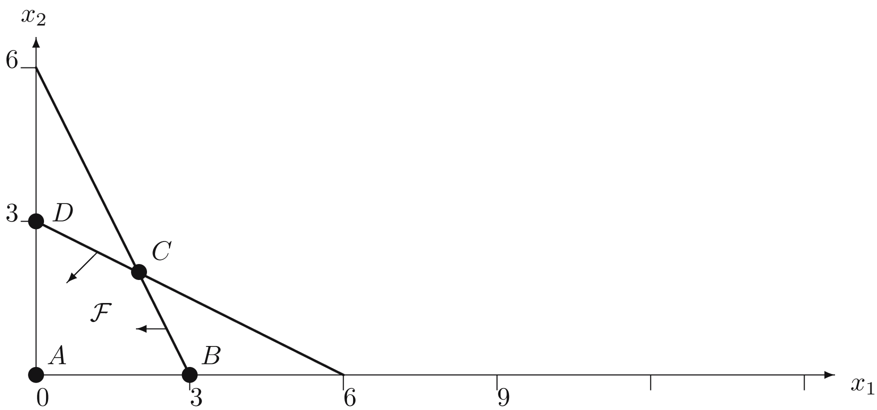

Fig. 6.1. Linear Programming Problem

Table 6.1. Solution to the Linear Program

Vertex	Coordinates	Objective Function
A	$(0,0)$	$Z = 0.0$
B	$(3,0)$	$Z = 6$
C	$(2,2)$	$Z = 10$
D	$(0,3)$	$Z = 9$

for some positive constant M. The positive constant M is for later on in the chapter and we will not get to use it in this section. Figure 6.1 shows the constraints and the feasible set $\mathcal{F}$. We know that the optimal solution will be at a vertex point of the feasible set. The values of the objective function at these vertex points is shown in Table 6.1 and we see that $\max Z = 10$ at $x_1 = x_2 = 2$.

6.3 Fuzzy Linear Program

Now we allow the parameters in the objective function and the constants on the right side of the inequalities to be fuzzy. The fuzzy linear program is

$$\max \overline{Z} = (\overline{C}_1\overline{X}_1 + \overline{C}_2\overline{X}_2), \tag{6.5}$$

subject to

$$\overline{X}_1 + 2\overline{X}_2 \leq \overline{B}_1, \tag{6.6}$$

$$2\overline{X}_1 + \overline{X}_2 \leq \overline{B}_2, \tag{6.7}$$

$$0 \leq \overline{X}_1, \overline{X}_2 \leq M, \tag{6.8}$$

where $\overline{C}_1 = (1/2/3)$, $\overline{C}_2 = (2/3/4)$, $\overline{B}_1 = (5/6/7)$, $\overline{B}_2 = (5/6/7)$ and $\overline{X}_1 \approx (x_{11}/x_{12}/x_{13})$, $\overline{X}_2 \approx (x_{21}/x_{22}/x_{23})$. The fuzzy parameters are all triangular fuzzy numbers but the variables will be Bézier (quadratic) fuzzy numbers (QBGFNs in Chapter 4).

Now we will look at two cases for evaluating $\leq$ between fuzzy numbers. Both of these methods are needed in the next two chapters, but they will not be used after Chapter 8. The first is Kerre's method from Section 2.6.2 in Chapter 2.

6.3.1 Kerre's Method

We will randomly generate, from Chapter 4, vectors $\overline{V} = (\overline{X}_1, \overline{X}_2)$ and first check to see if they satisfy equations (6.6) and (6.7) using Kerre's $\leq$. If these equations are satisfied, then $\overline{V}$ is feasible and we evaluate $\overline{Z} = \overline{C}_1\overline{X}_1 + \overline{C}_2\overline{X}_2$. Let the previous best (max) value of $\overline{Z}$ be $\overline{Z}^*$ and the current value of $\overline{Z} = \overline{Z}_0$ from the recent feasible $\overline{V}$. If $\overline{Z}^* < \overline{Z}_0$, then set $\overline{Z}^*$ to be $\overline{Z}_0$, otherwise discard $\overline{Z}_0$ and generate the next random $\overline{V}$. We are looking for an optimal solution and not all the $\overline{V}$ that produce the best $\overline{Z}$ value.

Next we need to determine intervals $I_i = [0, M_i]$, $M_i > 0$, $i = 1, 2$, for the $\overline{X}_i$, $i = 1, 2$, respectively. A good selection of these intervals will make the fuzzy Monte Carlo process more efficient. If an interval is too big, then too many $\overline{V}$ will be rejected as not being feasible. If an interval is too small we can miss the optimal solution. There is no natural upper bound on x_{13} (x_{23}) so that $\overline{V}$ is feasible. See Figure 6.2. Also see Figure 2.7 in Chapter 2. Here $\overline{E} \approx (e_1/e_2/e_3)$ represents $\overline{X}_1 + 2\overline{X}_2$ or $2\overline{X}_1 + \overline{X}_2$ and let $\overline{B} = (0.5/1.5/2.5)$. We changed $\overline{B}$ from (5/6/7) to this value for this figure. Then $e_3 = x_{13} + 2x_{23}$ or $2x_{13} + x_{23}$. In Figure 6.2 $\overline{E} \approx (0/1/5)$. We see that $d(\overline{E}, \overline{max})$ is the area of regions A_1 and A_2 and $d(\overline{B}, \overline{max})$ is the area of A_3. Since $area(A_3) < area(A_1) + area(A_2)$ we get $\overline{E} < \overline{B}$ and $\overline{V}$ is feasible even as e_3 grows larger and larger. In practical problems there is going to be an upper bound for the variables which will produce an upper bound for e_3. Management will decide on practical upper bounds for the x_i giving the upper bounds for the M_i in the intervals I_i. Sometimes the optimization problem will dictate the upper bounds, but in this case we get them for experts familiar with the problem. Let us assume that $I_i = [0, 5]$, $i = 1, 2$, which implies that $\overline{E} < 15$.

So we now randomly generate a sequence $\overline{V}_k = (\overline{X}_{1k}, \overline{X}_{2k})$ with $\overline{X}_{ik} \in [0, 5]$ all i and all k. Using our Sobol quasi-random number generator we produce sequences of random vectors $v_{1k} = (x_{1k1}, ..., x_{1k5})$, $v_{2k} = (x_{2k1}, ..., x_{2k5})$, $k = 1, 2, 3, ...$ The sequence v_{1k} is used to get the sequence of quadratic fuzzy numbers $\overline{X}_{1k}$, recall that we only require vectors of length five for these fuzzy numbers (see Chapter 4), and the other sequence v_{2k} constructs the sequence of quadratic fuzzy numbers $\overline{X}_{2k}$, $k = 1, 2, 3, ...$ However, because we use vectors $\overline{V} = (\overline{X}_1, \overline{X}_2)$ we choose our stream of quasi-random numbers generated 10 at a time to get the two pairs of 5. If $\overline{V}_k$ is feasible we compute

$$\overline{Z}_k = \overline{C}_1\overline{X}_{1k} + \overline{C}_2\overline{X}_{2k}, \tag{6.9}$$

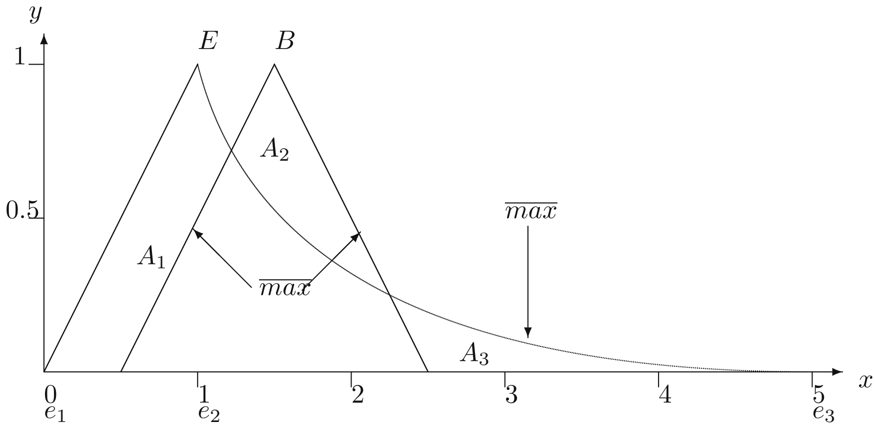

Fig. 6.2. Finding the Intervals for the $\overline{X}_i$ Using Kerre's Method

for $k = 1, 2, 3, ..., N$, where N is the predetermined total number of iterations. We want to find a k value, and hence a $\overline{V}_k$, to solve

$$max\{\overline{Z}_k | k = 1, 2, 3, ..., N\}. \tag{6.10}$$

With $N = 100,000$ pairs of QBGFNs (v_{1k}, v_{2k}), the results of the fuzzy Monte Carlo method are shown in Table 6.2, and Figures 6.3 & 6.4. All the fuzzy numbers in Table 6.2 are triangular shaped fuzzy numbers. $\overline{X}_1$ and $\overline{X}_2$ are QBGFNs. The notation we use for these fuzzy numbers was explained in Section 4.3.2. We define a QBGFN as (a, b, c, d, e) where: (1) the support is the interval $[a, c]$; (2) the vertex is at $x = b$; (3) the three numbers a, d, b define the quadratic function for the left side of the fuzzy number; and (4) the three numbers b, e, c specify the quadratic function for the right side of the fuzzy number. Since $max\overline{Z}$ is not necessarily, or likely to be a QBGFN, we only give the support and core for $max\overline{Z}$. Our approximate solution to this fuzzy linear program are the fuzzy numbers determined by this Monte Carlo program. In the following three chapters we may have another solution, using an evolutionary algorithm, to compare to our fuzzy Monte Carlo solution.

Table 6.2. Monte Carlo Solution to the Fuzzy Linear Program, Kerre's Method, QBGFNs, N=100,000

type random	$max\overline{Z}$	$\overline{X}_i$
Sobol	$\approx (2.70/8.67/33.74)$	$\overline{X}_1 = (1.28, 1.49, 4.97, -0.42, 3.61)$
		$\overline{X}_2 = (0.71, 1.90, 4.70, -0.32, 2.69)$

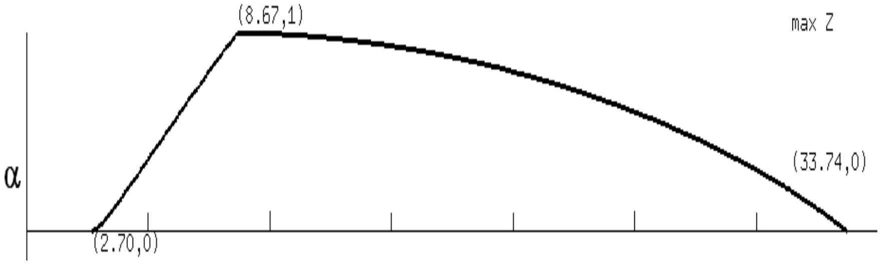

Fig. 6.3. Monte Carlo Solution to the Fuzzy Linear Program, $max\overline{Z}$, Kerre's Method, QBGFNs, N=100,000

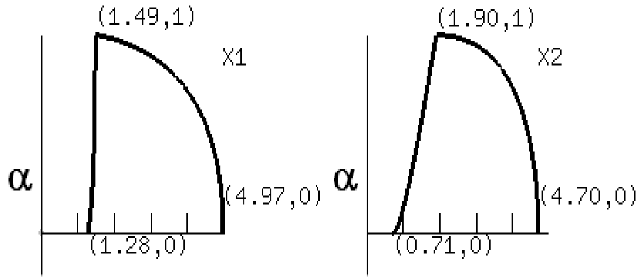

Fig. 6.4. Monte Carlo Solution to the Fuzzy Linear Program, $\overline{X}_1$ & $\overline{X}_2$ Kerre's Method, QBGFNs, N=100,000

6.3.2 Chen's Method

We randomly generate vectors $\overline{V} = (\overline{X}_1, \overline{X}_2)$ and check to see if they satisfy equations (6.6) and (6.7) using Chen's $\leq$. If these equations are satisfied, then $\overline{V}$ is feasible and we evaluate $\overline{Z}$. Let the previous best (max) value of $\overline{Z}$ be $\overline{Z}^*$ and the current value of $\overline{Z} = \overline{Z}_0$ from the recent feasible $\overline{V}$. If $\overline{Z}^* < \overline{Z}_0$ using Chen's $<$, then set $\overline{Z}^*$ to be $\overline{Z}_0$, otherwise discard $\overline{Z}_0$ and generate the next random $\overline{V}$.

Now we need to determine intervals $I_i = [0, M_i]$, $M_i > 0$, $i = 1, 2$, for the $\overline{X}_i$, $i = 1, 2$, respectively. There is no natural upper bound on x_{13} (x_{23}) so that $\overline{V}$ is feasible. See Figure 6.5. Also see Figure 2.8 in Chapter 2. Here $\overline{E} \approx (e_1/e_2/e_3)$ represents $\overline{X}_1 + 2\overline{X}_2$ or $2\overline{X}_1 + \overline{X}_2$ and let $\overline{B} = (0.5/1.5/2.5)$. We changed $\overline{B}$ from (5/6/7) to this value for this figure. Then $e_3 = x_{13} + 2x_{23}$ or $2x_{13} + x_{23}$. In Figure 6.5 $\overline{E} \approx (0/1/5)$. Consulting Figures 2.8 and 6.5 we see that the y coordinate at: (1) L_E is 0.8; (2) L_B is 0.7; (3) R_E is 0.3; and (4) R_B is 0.4. So, from equation (2.53) in Chapter 2 we compute

$$\mu_T(\overline{E}) = 0.5(0.3 + (1 - 0.8)) = 0.25, \tag{6.11}$$

and

$$\mu_T(\overline{B}) = 0.5(0.4 + (1 - 0.7)) = 0.35, \tag{6.12}$$

and

$$\mu_T(\overline{E}) < \mu_T(\overline{B}), \tag{6.13}$$

implying that $\overline{E} < \overline{B}$ and $\overline{V}$ is feasible even as e_3 grows larger and larger. In practical problems there is going to be an upper bound for the variables which will produce an upper bound for e_3. Management will decide on practical upper bounds for the x_i giving the upper bounds for the M_i in the intervals I_i. Sometimes the optimization problem will dictate the upper bounds, but in this case we get them for experts familiar with the problem. Let us assume that $I_i = [0, 5]$, $i = 1, 2$, so that $\overline{E} < 15$.

So we now randomly generate a sequence $\overline{V}_k = (\overline{X}_{1k}, \overline{X}_{2k})$ with $\overline{X}_{ik} \in [0, 5]$ all i and all k. Using our Sobol quasi-random number generator we produce sequences of random vectors $v_{1k} = (x_{1k1}, ..., x_{1k5})$, $v_{2k} = (x_{2k1}, ..., x_{2k5})$, $k = 1, 2, 3, ...$ The sequence v_{1k} is used to get the sequence of quadratic fuzzy numbers $\overline{X}_{1k}$ and the other sequence v_{2k} constructs the sequence of quadratic fuzzy numbers $\overline{X}_{2k}$, $k = 1, 2, 3, ...$ However, because we use vectors $\overline{V} = (\overline{X}_1, \overline{X}_2)$ we choose our stream of Sobol quasi-random numbers generated 10 at a time to get the two pairs of 5. If $\overline{V}_k$ is feasible we compute

$$\overline{Z}_k = \overline{C}_1 \overline{X}_{1k} + \overline{C}_2 \overline{X}_{2k}, \tag{6.14}$$

for $k = 1, 2, 3, ..., N$, where N is the predetermined total number of iterations. We want to find a k value, and hence a $\overline{V}_k$, to solve

$$max\{\overline{Z}_k | k = 1, 2, 3, ..., N\}. \tag{6.15}$$

With $N = 100,000$ the results of the fuzzy Monte Carlo method are shown in Table 6.3, and Figures 6.6 & 6.7. All the fuzzy numbers in Table 6.3 are triangular shaped fuzzy numbers. $\overline{X}_1$ and $\overline{X}_2$ are QBGFNs. The notation we use for these fuzzy numbers was explained above and in Section 4.3.2. Since $max\overline{Z}$ is not necessarily or likely to be a QBGFN, we only give the support

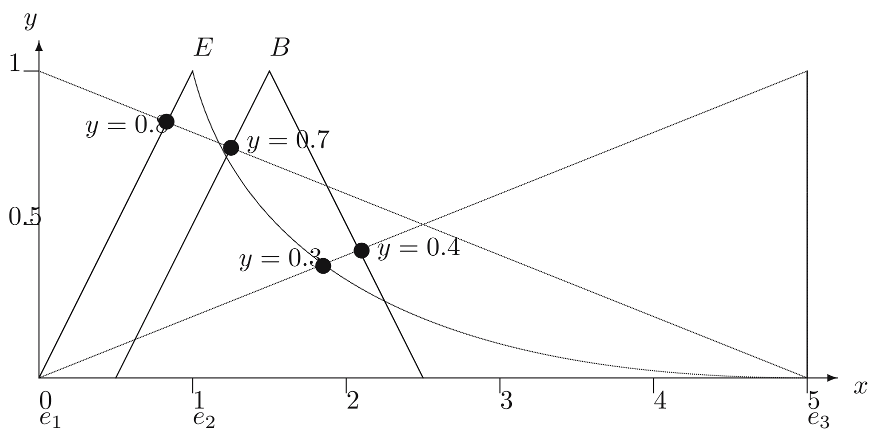

Fig. 6.5. Finding the Intervals for the $\overline{X}_i$ Using Chen's Method

Table 6.3. Monte Carlo Solution to the Fuzzy Linear Program, Chen's Method, QBGFNs, N=100,000

type random	$max\overline{Z}$	$\overline{X}_i$
Sobol	$\approx (1.37/5.07/34.84)$	$\overline{X}_1 = (0.08, 0.50, 4.99, -0.96, 4.69)$ $\overline{X}_2 = (0.64, 1.36, 4.97, 0.51, 2.72)$

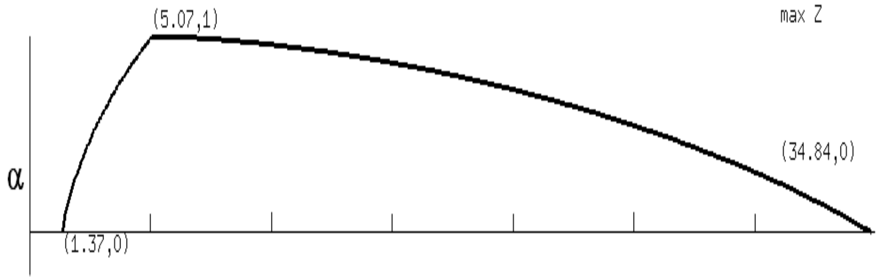

Fig. 6.6. Monte Carlo Solution to the Fuzzy Linear Program, $max\overline{Z}$, Chen's Method, QBGFNs, N=100,000

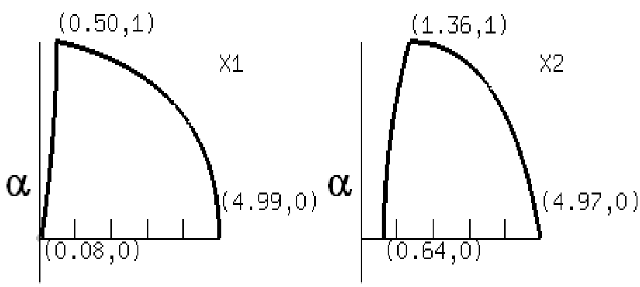

Fig. 6.7. Monte Carlo Solution to the Fuzzy Linear Program, $\overline{X}_1$ & $\overline{X}_2$ Chen's Method, QBGFNs, N=100,000

and core for $max\overline{Z}$. Our approximate solution to this fuzzy linear program are the fuzzy numbers determined by this Monte Carlo program. In the following three chapters we may have another solution, using an evolutionary algorithm, to compare to our fuzzy Monte Carlo solution.

6.3.3 Comparison of Solutions

All of these software efforts were performed on Windows-based PCs. For these fuzzy Monte Carlo optimizations, several computers were used, all Dell Optiplex

GX270's, 3.0GHz, 1GB RAM. The Kerre/Sobol running time was 2:27:50; the Chen/Sobol finished in 2:31:10.

The Kerre comparison method found $2,864$ feasible sets in a stream of $100,000$ $(\overline{X}_{1k},\overline{X}_{2k})$; 15 of them had triggered new maximums. The Chen comparison method found $17,047$ feasible sets in the same stream of $100,000$ $(\overline{X}_{1k},\overline{X}_{2k})$; 12 of them had triggered new maximums. The 341^{st} $(\overline{X}_{1k},\overline{X}_{2k})$ was a new maximum by both methods; the 4^{th} new maximum under Kerre, the 6^{th} new maximum under Chen. The last maximum by the Kerre method was found at the $15,251^{st}$ $(\overline{X}_{1k},\overline{X}_{2k})$ (it was the $2,864^{th}$, the last, feasible set). The last maximum by the Chen method was found at the $48,798^{th}$ $(\overline{X}_{1k},\overline{X}_{2k})$ (the $8,322^{nd}$ feasible set).

For our Monte Carlo solution using Sobol quasi-random numbers, we compare Kerre's method results and Chen's method results to find that the $max\overline{Z}$ from Kerre's method is greater than the $max\overline{Z}$ from Chen's method solution (regardless which of Buckley's method, Kerre's method, or Chen's method is used to compare the maximums). It appears that Kerre's method produced a solution closest to the crisp solution $x_1 = x_2 = 2$, $maxZ = 10$.

References

1. Aliev, R.A., Fazlollahi, B.: Soft Computing and its Applications to Business and Economics. Springer, Heidelberg (2004)
2. Bector, C.R., Chandra, S.: Fuzzy Mathematical Programming and Fuzzy Matrix Games. Springer, Heidelberg (2005)
3. Buckley, J.J.: Joint Solution to Fuzzy Linear Programming. Fuzzy Sets and Systems 72, 215–220 (1995)
4. Buckley, J.J., Feuring, T., Hayashi, Y.: Neural Net Solution to Fuzzy Linear Programming. Fuzzy Sets and Systems 106, 99–111 (1999)
5. Buckley, J.J., Feuring, T.: Evolutionary Algorithm Solution to Fuzzy Problems: Fuzzy Linear Programming. Fuzzy Sets and Systems 109, 35–53 (2000)
6. Buckley, J.J., Eslami, E., Feuring, T.: Fuzzy Mathematics in Economics and Engineering. Physica-Verlag, Heidelberg (2002)
7. Carlsson, C., Fuller, R.: Fuzzy Reasoning in Decision Making and Optimization. Physica-Verlag, Heidelberg (2002)
8. Ehrgott, M., Gandibleux, X.: Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys. Kluwer Academic Press, Norwell, Mass. (2002)
9. Gen, M., Cheng, R.: Genetic Algorithms and Engineering Optimization. Wiley Interscience, N.Y. (1999)
10. Inuiguchi, M., Ramik, J., Tanino, T., Vlach, M.: Satisficing Solutions and Duality in Interval and Fuzzy Linear Programming. Fuzzy Sets and Systems 135, 151–177 (2003)
11. Jamison, K.D., Lodwick, W.A.: Fuzzy Linear Programming Using a Penalty Method. Fuzzy Sets and Systems 119, 97–110 (2001)
12. Li, D., Cheng, C.: Fuzzy Multiobjective Programming Methods for Fuzzy Constrained Matrix Games with Fuzzy Numbers. Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems 10, 385–400 (2002)
13. Li, D.-F., Yang, J.-B.: Fuzzy Linear Programming Technique for Multiattribute Group Decision Making in Fuzzy Environments. Information Sciences 158, 263–275 (2004)

14. Liu, X.: Measuring the Satisfaction of Constraints in Fuzzy Linear Programming. Fuzzy Sets and Systems 122, 263–275 (2001)
15. Maeda, T.: Fuzzy Linear Programming Problems as Bi-Criteria Optimization Problems. Applied Mathematics and Computation 120, 109–121 (2001)
16. Nishizaki, I., Sakawa, M.: Solutions Based on Fuzzy Goals in Fuzzy Linear Programming Games. Fuzzy Sets and Systems 115, 105–119 (2000)
17. Nishizaki, I., Sakawa, M.: Fuzzy and Multiobjective Games for Conflict Resolution. Physica-Verlag, Heidelberg (2001)
18. Rommelfanger, H.: Fuzzy Linear Programming and Applications. European J. Operational Research 92, 512–527 (1996)
19. Sakawa, M.: Genetic Algorithms and Fuzzy Multiobjective Optimization. Kluwer Academic Press, Norwell, Mass. (2002)
20. Tang, J., Wang, D., Fung, R.Y.K., Yung, K.-L.: Understanding of Fuzzy Optimization: Theories and Methods. J. System Science and Complexity 17, 117–136 (2004)
21. Trzaskalik, T., Michnik, J. (eds.): Multiple Objective and Goal Programming. Physica-Verlag, Heidelberg (2002)
22. Van Hop, N.: Solving Fuzzy (Stochastic) Linear Programming Problems Using Superiority and Inferiority Measures. Information Sciences 177, 1977–1991 (2007)
23. Vasant, P.M.: Application of Fuzzy Linear Programming in Production Planning. Fuzzy Optimization and Decision Making 2, 229–241 (2003)
24. Wang, R.-C., Fang, H.-H.: Aggregate Production Planning with Multiple Objectives in a Fuzzy Environment. European J. Operational Research 133, 521–536 (2001)
25. Wu, H.-C.: Duality Theory in Fuzzy Linear Programming Problems with Fuzzy Coefficients. Fuzzy Optimization and Decision Making 2, 61–73 (2003)
26. Yoshida, Y. (ed.): Dynamical Aspects in Fuzzy Decision Making. Physica-Verlag, Heidelberg (2001)
27. Zhang, G., Wu, Y.-H., Remias, M., Lu, J.: Formulation of Fuzzy Linear Programming Problems as Four-Objective Constrained Optimization Problems. Applied Mathematics and Computation 139, 383–399 (2003)