1 Introduction

1.1 Introduction

The objective of this book is to introduce Monte Carlo methods to find good approximate solutions to fuzzy optimization problems. Many crisp (nonfuzzy) optimization problems have algorithms to determine solutions. This is not true for fuzzy optimization. There are other things to discuss in fuzzy optimization, which we will do later on in the book, like $\leq$ and $<$ between fuzzy numbers since there will probably be fuzzy constraints, and how do we evaluate $max/min\overline{Z}$ for $\overline{Z}$ the fuzzy value of the objective function.

This book is divided into four parts: (1) Part I is the Introduction containing Chapters 1-5; (2) Part II, Chapters 6-16, has the applications of our Monte Carlo method to obtain approximate solutions to fuzzy optimization problems; (3) Part III, comprising Chapters 17-27, outlines our "unfinished business" which are fuzzy optimization problems for which we have not yet applied our Monte Carlo method to produce approximate solutions; and (4) Part IV is our summary, conclusions and future research.

1.1.1 Part I

First we need to be familiar with fuzzy sets. All you need to know about fuzzy sets for this book comprises Chapter 2. For a beginning introduction to fuzzy sets and fuzzy logic see [2]. Three other items related to fuzzy sets, needed in this book, are also in Chapter 2: (1) in Section 2.5 we discuss how we have dealt in the past with determining $max/min(\overline{Z})$ for $\overline{Z}$ a fuzzy set representing the value of an objective function in a fuzzy optimization problem; (2) in Section 2.6 we present the three methods we will use in this book to determine which of the following possibilities $\overline{M} < \overline{N}$, $\overline{M} > \overline{N}$ or $\overline{M} \approx \overline{N}$ is true, for two fuzzy numbers $\overline{M}$, $\overline{N}$; and (3) in Section 2.7 we investigate dominated, and undominated, fuzzy vectors.

Chapter 3 introduces the random number generators that we will be using in the rest of the book. We will need sequences of crisp random vectors $v =$

J.J. Buckley et al.: Monte Carlo Meth. in Fuzzy Optimization, STUDFUZZ 222, pp. 3–7, 2008.
springerlink.com

$(x_1, ..., x_n)$ with $x_i \in [0, 1]$, $n \geq 3$. If we use a pseudo-random number generator to produce the v it is well known (see Chapter 3) that when we plot these points in $[0, 1]^n$ we can get empty regions and clustering. We need a method of getting sequences of v that will uniformly fill $[0, 1]^n$. Such a method already exists. They are called quasi-random number generators, introduced and discussed in Chapter 3.

Next we need to randomly generate sequences of fuzzy numbers and sequences of fuzzy vectors. We usually use triangular fuzzy numbers (TFNs) and quadratic fuzzy numbers (Section 2.2.1). The quadratic fuzzy numbers we use are called quadratic Bézier generated fuzzy numbers (QBGFNs) which are defined in Chapter 4. We show in this chapter how we use the sequences $v = (x_1, ..., x_n)$, from a quasi-random number generator, to get sequences of TFNs/QBGFNs and sequences of vectors of TFNs/QBGFNs. We use these results to show that the three methods of evaluating $\leq$ and $<$ between fuzzy numbers given in Sections 2.6.1 - 2.6.3 quite often give the same results.

Chapter 5 is about testing our sequences of fuzzy numbers/vectors for randomness. There are many tests for randomness for sequences of crisp numbers, but most are not applicable to fuzzy numbers. However, the run test can be extended to fuzzy numbers and our results are presented in this chapter. We have one randomness test for sequences of fuzzy vectors. But we do argue that our method of producing sequences of fuzzy vectors will uniformly fill the search space in a fuzzy optimization problem. That is exactly what is needed for a sequence of fuzzy vectors used to get an approximate solution to a fuzzy optimization problem.

1.1.2 Part II

Part II contains our applications of Monte Carlo methods to generating approximate solutions to fuzzy optimization problems. Fuzzy linear programming is in Chapters 6-9. Getting solutions, or approximate solutions, to fuzzy equations is the topic of Chapter 10. Applications to fuzzy regression is the theme for Chapters 11-14. The last two applications are to fuzzy game theory (Chapter 15) and to fuzzy queuing theory (Chapter 16).

1.1.3 Part III

The chapters here describe more fuzzy optimization problems that do not have algorithms that give an exact fuzzy solution. Therefore they are candidates for an approximate Monte Carlo solution. In each case we first outline the problem and then discuss how we might use Monte Carlo to generate approximate solutions, but we do not do it. We leave it for future research by the authors or interested readers.

1.1.4 Part IV

This consists of one short chapter containing a summary, our suggestions for future research and our conclusions.

1.2 Notation

It is difficult, in a book with a lot of mathematics, to achieve a uniform notation without having to introduce many new specialized symbols. Our basic notation is presented in Chapter 2. What we have done is to have a uniform notation within each chapter. What this means is that we may use the letters "a" and "b" to represent a closed interval $[a, b]$ in one chapter but they could stand for something else in another chapter. We will have the following uniform notation throughout the book: (1) we place a "bar" over a letter to denote a fuzzy set ($\overline{A}$, $\overline{B}$, etc.), and all our fuzzy sets will be fuzzy subsets of the real numbers; and (2) an α-cut of a fuzzy set (Chapter 2) is always denoted by "α".

We use the abbreviations: FN for fuzzy number; TFN for triangular fuzzy number; TrFN for trapezoidal fuzzy number, and QBGFN for quadratic Bézier generated fuzzy number (defined in Chapter 4). All fuzzy arithmetic is performed using α-cuts and interval arithmetic and not by using the extension principle (Chapter 2).We also use TBC to mean "to be completed" in Part III.

The term "crisp" means not fuzzy. A crisp set is a regular set and a crisp number is a real number. There is a potential problem with the symbol "$\leq$". It sometimes means "fuzzy subset" as $\overline{A} \leq \overline{B}$ stands for $\overline{A}$ is a fuzzy subset of $\overline{B}$ (defined in Chapter 2). However, also in Chapter 2, $\overline{A} \leq \overline{B}$ means that fuzzy set $\overline{A}$ is less than or equal to fuzzy set $\overline{B}$. The meaning of the symbol "$\leq$" should be clear from its use.

1.3 Previous Research

Mathematica has added random fuzzy numbers [11]. It can create "random" trapezoidal, Gaussian and triangular fuzzy numbers. They are represented by thin vertical bars similar to a histogram. We would need the functional expressions for the sides of the fuzzy numbers and it is not clear how we could get that information from Mathematica. The web site does not tell the user how these "random" fuzzy numbers are generated.

Most authors ([1],[5],[6],[8]-[10]) have used the following method to define random fuzzy numbers. Consider LR fuzzy numbers which we write as $(a, b, c)_{LR}$. Let m, l and r be three real-valued random variables with $l > 0, r > 0$. Then a random LR-fuzzy number is $(m, l, r)_{LR}$. The functions L and R are called the left and right membership functions, m is where the membership value equals one (vertex point) and $l(r) \geq 0$ is the left (right) spread of the fuzzy number. So, once you pick and fix L and R, the randomness is in the end points of the $\alpha = 0$ cut and the vertex point of the fuzzy number. We think the randomness should also be in the shape of the fuzzy number. That is, we should also be able to randomly change L and R.

Next, the paper [4] has another way to construct random fuzzy numbers. Let $F_i(x)$, $i = 1, 2, 3$, be probability distribution functions. Randomly choose $y \in [0, 1]$, then a random triangular fuzzy number has base $[F_1^{-1}(y), F_3^{-1}(y)]$ and vertex point at $F_2^{-1}(y)$. We have assumed that $F_1^{-1}(y) < F_2^{-1}(y) < F_3^{-1}(y)$

for y in $[0, 1]$. This just randomly produces a triangular fuzzy number but it always has the same shape (a triangle). [12] uses Gaussian fuzzy numbers to generate random fuzzy numbers but employs crisp Monte Carlo methods. The book [3] has one very short section on fuzzy Monte Carlo simulation

Finally, [7] generates a random triangular fuzzy number as $(m - 6/m/m + 2)$ for m uniform on $[1, 3]$ and a random trapezoidal fuzzy number as $(m - 4/m - 2, m + 4/m + 6)$ for m a standard normal random variable. Again the shape is always the same, straight line segments for the sides of the fuzzy number.

To complete a search for random fuzzy numbers/vectors we suggest putting the following items into your search engine: fuzzy random numbers, random fuzzy numbers, fuzzy Monte Carlo,... The authors believe that there are no other publications that cover in our detail generating random fuzzy numbers/vectors to give approximate solutions to fuzzy optimization problems. Our random fuzzy numbers will have random base, random vertex point and also (limited) random shape. We believe this gives a better picture of random fuzzy numbers for fuzzy Monte Carlo methods.

Beginning in Chapter 6 many different topics are covered until Chapter 27. Most chapters have a list of references at the end of the chapter. We give only a few key references to the topic were the interested reader may find other references. This book will not do a complete literature search on each area for you, but we do hope that we have given you a good start.

1.4 MATLAB/C++ Programs

Computer programs were written in MATLAB or C++. Only one MATLAB program is given in the book and it is at the end of Chapter 14. However, any of these programs can be obtained from the authors.

References

1. Bandemer, H., Gebhardt, A.: Bayesian Fuzzy Kriging, `www-stat.uni-klu.ac.at/~agebhard/fuba/`
2. Buckley, J.J., Eslami, E.: An Introduction to Fuzzy Logic and Fuzzy Sets. Springer, Heidelberg (2002)
3. El-Hawary, M.E. (ed.): Electric Power Applications of Fuzzy Systems. IEEE Power Engineering Series. Wiley-IEEE Press (1998)
4. Ferson, S., Ginzburg, L.: Hybrid Arithmetic. In: Proceedings ISUMA-NAFIPS, pp. 619–623 (1995)
5. Hong, D.H., Ahn, C.H.: Equivalent Conditions for Laws of Large Numbers for T-Related L-R Fuzzy Numbers. Fuzzy Sets and Systems 136, 387–395 (2003)
6. Jimenez-Gamero, M.D., Pino-Mejias, R., Rojas-Medar, M.A.: A Bootstrap Test for the Expectation of Fuzzy Random Variables. Comput. Statist. Data Anal. (2004)
7. Korner, R.: An Asymptotic $\alpha-$test for the Expectation of Random Fuzzy Variables. J. Statistical Planning and Inference 83, 331–346 (2000)
8. Liu, Y.-K., Chen, Y.-J.: Multicriteria Optimization Problem in Fuzzy Random Decision Systems. In: Proc. Third Int. Conf. Machine Learning and Cybernetics, Shanghai, China (2004)

9. Nather, W.: Linear Statistical Inference for Fuzzy Data. In: 3^{rd} Int. Symposium Uncertainty Modeling and Analysis (ISUMA-NAFIPS), pp. 71–74 (1995)
10. Nather, W., Korner, R.: Linear Regression with Random Fuzzy Numbers. In: Ayyub, B.M., Gupta, M.M. (eds.) Uncertainty Analysis in Engineering and the Sciences, Kluwer, Boston (1998)
11. `www.wolfram.com/products/mathematica`
12. Zonouz, S.A., Miremadi, S.G.: A Fuzzy-Monte Carlo Simulation Approach for Fault Tree Analysis. In: IEEE 52^{nd} Annual Reliability and Maintainability Symposium (RAMS), Newport Beach, CA, pp. 428–433 (January 2006)