28 Summary, Conclusions, Future Research

28.1 Summary

In this book we hoped to convince the reader that Monte Carlo methods can be useful in generating approximate solutions to fuzzy optimization problems. In a Monte Carlo procedure we randomly produce $N$ possible (feasible) solutions to an optimization problem, subject to some criteria we keep only the “best” feasible solutions, and as $N$ grows larger and larger we converge to an optimal solution. Having a feasible solution usually means it satisfies the constraints to the problem and these constraints usually involve equalities and inequalities. Monte Carlo methods are known to be very inefficient and are seldom used in crisp optimization problems since these problems usually have their own efficient solution algorithms. However, fuzzy optimization problems usually do not have their own efficient solution algorithms so Monte Carlo methods become more important in fuzzy optimization.

Monte Carlo methods in fuzzy optimization require us to randomly generate $N$ feasible solutions. These feasible solutions will be fuzzy numbers or fuzzy vectors. The fuzzy numbers we use are triangular fuzzy numbers, trapezoidal fuzzy numbers or quadratic fuzzy numbers (coded QBGFNs in the book). A quadratic fuzzy number is a triangular shaped fuzzy number whose sides are described by quadratic functions. Fuzzy vectors are vectors composed of fuzzy numbers. We need to produce random sequences of fuzzy numbers and random sequences of fuzzy vectors. How we do this is explained in detail in Chapter 4. We use streams of crisp quasi-random numbers discussed in Chapter 3 to produce sequences of random fuzzy numbers/vectors. Randomness tests on our sequences of random fuzzy numbers/vectors is presented in Chapter 5.

Next we need to check to see if our random fuzzy number/vector is feasible. When the constraints involve $<$, or $\leq$, between fuzzy numbers we use three methods to evaluate inequalities between fuzzy numbers: (1) Buckley’ Method in Section 2.6.1; (2) Kerre’s Method in Section 2.6.2; and (3) Chen’s Method in Section 2.6.3. Of course, our computer programs may be altered to employ any method of evaluating inequalities between fuzzy numbers. Once we have a
sequence of random feasible solutions we need to pick the "best" ones. Most of the fuzzy optimization problems we study in the applications chapters (Chapters 6 - 16) involve finding the maximum/minimum of a fuzzy function subject to fuzzy constraints. We use the same methods (Buckley, Kerre, Chen) to find the maximum/minimum of a collection of fuzzy numbers. A key property that a Monte Carlo method must have, to be able to converge on an optimal solution, is that it uniformly fill the search space. We argue that our procedure has this property in Section 5.3.

A major problem in fuzzy Monte Carlo is to decide on intervals \([a_i, b_i], 1 \leq i \leq m\), for our random fuzzy numbers \(X_i, i = 1, 2, 3, \ldots, m\). That is, randomly generate \(X_i \in [a_i, b_i]\) all \(i\). If an interval is too small we can miss a good solution. If the intervals are too big we can produce many infeasible candidates. This problem is discussed starting in Chapter 6 and you can use the key phrase “intervals for Monte Carlo” in the Index.

We applied our fuzzy Monte Carlo method to fully fuzzified linear programming (Chapters 7-9), solving fuzzy equations (Chapter 10), to fuzzy regression (Chapters 11-14), fuzzy game theory (Chapter 15) and fuzzy queuing theory (Chapter 16). In some cases these fuzzy optimization problems had approximate optimal solutions from previous publications. In all cases, except one situation as explained in Chapter 11, our Monte Carlo method obtained a better approximate solution.

The computer time can be quite long. In one case it was 68 hours for \(N = 100,000\) in Chapter 15. So we suggested doing it in parallel. Use 10 computers each for \(N = 100,000\) to get a run of 1,000,000.

There are many fuzzy optimization problems we have not yet applied our fuzzy Monte Carlo method to calculate an approximate solution and some of these are outlined in Chapters 17-27.

### 28.2 Future Research

Future research could be involved with continuing to use our fuzzy Monte Carlo method on fuzzy linear programming problems and on fuzzy regression problems. But more importantly it would involve attacking those fuzzy optimization problems discussed in Chapters 17-27. Of course, there are fuzzy optimization problems not presented in this book that may also be (approximately) solved by Monte Carlo.

We encourage others to review and extend the work which we have begun. In preparation for Chapters 3-10 of this book we have developed seven programs (Table 28.1). Each is written in Visual C++; some use the OpenGL graphics library. We offer the source and data files freely to those wanting to do further research. Contact Leonard Jowers (LJJowers@uab.edu) for more information. A number of MATLAB programs were created for Chapters 11-16. One is contained in Chapter 14. Others may be obtained by contacting James J.Buckley (buckley@math.uab.edu).