16 Fuzzy Queuing Models

16.1 Introduction

This chapter is based on, and expanded from, Chapters 11, 12 and 14 of [4] which is about using fuzzy probabilities and fuzzy sets in web site planning. So the queuing network considered in this chapter is within a web site. For other papers/chapters in books, on this topic of fuzzy queuing theory, we refer the reader to ([2],[3],[6],[7],[10]) and the references in these papers/books. In the next section we discuss the crisp queuing optimization problem and then we fuzzify the optimization problem in the third section. In the fourth section we present our fuzzy Monte Carlo method and how we will generate sequences of random fuzzy vectors. Our fuzzy Monte Carlo solution to the fuzzy queuing optimization problem is the fifth section and the last section has a summary and our conclusions. All the fuzzy numbers used in this chapter, except fuzzy profit starting in Section 16.3, will be non-negative. We will program our fuzzy Monte Carlo method in MATLAB [8]. This chapter is also based on [1].

16.2 Queuing Model

We will model the queuing system using the arrival rate λ and the service rate μ for any server. This is a common method used in queuing theory ([9],[11]). The system has c parallel and identical servers, system capacity M (in the servers and in the queue, $c \leq M$) and an infinite calling source. If the system is full, new arrivals are turned away and lost from the system. The λ rate will be state independent, which means that λ does not depend on how many customers are in the system. But if there are n customers in the system, then the rate of departure from the whole system is $\mu_n = n\mu$, for $0 \leq n < c$ and $\mu_n = c\mu$ for $c \leq n \leq M$.

A basic assumption is that we are in steady-state, all transient behavior has died down and can be neglected, and the time interval δ is sufficiently small so that the probability of two or more events occurring during δ is zero. If we are in state n, or there are n customers in the system with $0 < n < M$, we can

J.J. Buckley et al.: Monte Carlo Meth. in Fuzzy Optimization, STUDFUZZ 222, pp. 175–184, 2008.
springerlink.com

have only two events occurring: (1) one customer arrives and we have $n+1$ in the system; or (2) one customer finishes service and leaves and we have $n-1$ left in the system. Usually for steady-state we assume that $\lambda \leq \mu$ when we have infinite capacity. However, since we have finite system capacity we do not need to assume that $\lambda \leq \mu$. If we are in state zero ($n=0$), we can only go to $n=1$ and we can get to state $n=0$ from $n=1$ when a customer leaves service. We can get to state $n=M$ only from $n=M-1$ with an arrival and we can leave state $n=M$ to state $M-1$ when a customer leaves service.

The first objective is to compute the steady-state probabilities w_i, $0 \leq i \leq M$, from which we may determine various measures of system performance.

Using a transition rate diagram, the expected rate of flow into state n is

$$\lambda w_{n-1} + \mu_{n+1} w_{n+1}, \tag{16.1}$$

and the expected rate of flow out of state n is

$$\lambda w_n + \mu_n w_n, \tag{16.2}$$

for $0 < n < M$. We set these two equal to get the balance equation

$$\lambda w_{n-1} + \mu_{n+1} w_{n+1} = \lambda w_n + \mu_n w_n, \tag{16.3}$$

for $0 < n < M$. The balance equation for state $n=0$ is simply $\lambda w_0 = \mu_1 w_1$ and for $n=M$ it is $\lambda w_{M-1} = \mu_M w_M$. We solve these balance equations for w_i, $1 \leq i \leq M$, functions of w_0 and then use the fact that the sum of all the w_i must equal one to obtain a formula for w_0. The final result is that $w_i = F_i(\lambda, \mu, c, M)$, $0 \leq i \leq M$. That is, the steady-state probabilities are function of λ, μ, c and M ([9],[11]).

Now we can determine measures of system performance such as U=server utilization and N=expected number of customers in the system. All we will need in this chapter is

$$N = \sum_{k=0}^{M} k w_k. \tag{16.4}$$

We next wish to consider the optimal queuing network maximizing profit with variables λ, μ, c and M.

There are many types of servers each with associated service rate $\mu \in [0, 10]$. Let $C = K_1 \mu$ be the cost, in \$ per unit time, of operating a server having corresponding service rate μ for some constant $K_1 > 0$. Determining the cost of a server per unit time is a difficult number to estimate so we later assign a fuzzy number to its value.

There is a cost involved in maintaining the queue, or those in the system but not yet in a server. Let Q be the cost, in \$ per unit time, of having one space available in the queue. So the queue cost is $Q(M-c)$. Q is also difficult to estimate exactly so we will later model it as fuzzy.

There will be certain fixed costs associated with maintaining the web site which do not depend on the decision variables. Since they do not depend on M,c,λ and μ they can be omitted for the model.

We will assume that we can affect the arrival rate λ through advertising. Let $A = K_2\lambda$ be the advertising level in \$ per unit time that is expected to produce arrival rate λ for constant $K_2 > 0$. This cost A will be hard to know precisely so it too will become fuzzy. The web site pays for these advertisements but other advertisers will pay the web site, to place their ads, depending on the number of customers in the queue who can see their (pop-up) adds.

Revenue from advertisers is assumed to be proportional to the average number of customers in the system N. If T is the revenue, in \$ per unit time, per customer in the system, then total revenue per unit time is TN.

Profit per unit time, to be maximized, is

$$Profit = TN - [K_2\lambda + Q(M-c) + K_1\mu\ c], \tag{16.5}$$

We will next model all the cost/income parameters as fuzzy numbers, and if any are known exactly, then we would use their exact values. The variables are $c = 1, 2, .., 10$ and $M = c, c+1, ..., 30$, $\lambda \in [0, 10]$ and $\mu \in [0, 10]$.

We could consider a budget constraint, only so much money available per unit time, but we will not do this here. The above profit equation gives us only one goal, maximize profit. We could add other goals [4] such as maximize server utilization, minimize number of lost customers due to system capacity M, etc.

There are many other costs associated with the system, such as startup costs and operating costs ([9], Chapter 5), which we have not incorporated into the model. Many of these costs are independent from our variables, so can be classified as fixed costs in our model, and hence omitted.

16.3 Fuzzy Queuing Model

The arrival rate would need to be estimated and we will use a fuzzy estimator $\overline{\lambda}$ ([4], Chapter 3). Assume that we gather data to estimate the arrival rate. Then we can construct $(1-\gamma)100\%$ confidence intervals for λ. If we place these confidence intervals one on top of another, $0.001 \leq \gamma \leq 1$, we obtain a fuzzy number $\overline{\lambda}$. We can easily change the MATLAB program to use a crisp value for λ but in this chapter we will use a fuzzy $\overline{\lambda}$. Also, the service rate has to be estimated so we have a fuzzy estimator $\overline{\mu}$ ([4], Chapter 3). We get $\overline{\mu}$ as described above from the $(1-\gamma)100\%$ confidence intervals, $0.001 \leq \gamma \leq 1$. The fuzzy numbers obtained from the confidence interval method will be triangular shaped fuzzy numbers but in this chapter we will use triangular fuzzy numbers for $\overline{\lambda}$ and $\overline{\mu}$. The server cost automatically becomes fuzzy $\overline{C} = K_1\overline{\mu}$ for crisp constant $K_1 > 0$. Also, the advertising cost is fuzzy $\overline{A} = K_2\overline{\lambda}$ for crisp $K_2 > 0$. The fuzzy queue cost is $\overline{Q} > 0$ and the fuzzy revenue from advertisers is $\overline{T} > 0$. All these crisp and fuzzy constants are given in Table 16.1. The fuzzy constants are all triangular fuzzy numbers. The constants K_1, K_2, $\overline{Q}$ and $\overline{T}$ might be obtained from expert opinion (Section 3.4 in [4]).

The fuzzy optimization problem is to find integer $c \in [1, 10]$, integer $M \in [1, 30]$ with $c \leq M$ and $\overline{\lambda}$, $\overline{\mu} \in [0, 10]$ to maximize fuzzy profit

$$\overline{\Pi} = \overline{T}\ \overline{N} - [K_2\overline{\lambda} + \overline{Q}(M-c) + K_1\overline{\mu}c]. \tag{16.6}$$

Table 16.1. Crisp/Fuzzy Parameters in the Fuzzy Optimization Problem

Constant	$\alpha = 0$ Cut
K_1	0.04
K_2	0.03
$\overline{Q}$	(0.04/0.07/0.10)
$\overline{T}$	(3.15/3.45/3.80)

Given the values of the variables the next thing to do is to get the fuzzy steady-state probabilities $\overline{w}_k$, $0 \le k \le M$. We will first discuss computing the crisp steady-state probabilities and then fuzzify them using the extension principle. Let $\rho = \lambda/\mu$ and [11]

$$w_k = F_k(\lambda, \mu, c, M) = \frac{\rho^k}{k!} w_0, \ 1 \le k \le c, \tag{16.7}$$

and

$$w_k = F_k(\lambda, \mu, c, M) = \frac{\rho^k}{c!c^{k-c}} w_0, \ c \le k \le M. \tag{16.8}$$

Now $w_0 = F_0(\lambda, \mu, c, M)$ where

$$w_0 = [\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c(1-(\rho/c)^{M-c+1})}{c!(1-\rho/c)}]^{-1}, \ \rho/c \ne 1, \tag{16.9}$$

and

$$w_0 = [\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!}(M-c+1)]^{-1}, \ \rho/c = 1. \tag{16.10}$$

The test $\rho/c \ne 1$ and $\rho/c = 1$ in equations (16.9) and (16.10) will be difficult to do when λ and μ are fuzzy numbers. So we combine both of these equations into one equation eliminating the two tests on ρ/c. Let $s = M - c + 1$ and do the division in equation (16.9) producing

$$w_0 = [\sum_{n=0}^{c-1} \frac{\rho^n}{n!} + \frac{\rho^c}{c!c^{s-1}} P(c, s, \rho)]^{-1}, \tag{16.11}$$

where

$$P(c, s, \rho) = c^{s-1} + c^{s-2}\rho + c^{s-3}\rho^2 + \ldots + \rho^{s-1}. \tag{16.12}$$

We will use equations (16.11) and (16.12) to determine the steady-state probability w_0. Then

$$\overline{w}_k = F_k(\overline{\lambda}, \overline{\mu}, c, M), \tag{16.13}$$

for $0 \le k \le M$, evaluated using the extension principle. Let $\overline{w}_k[\alpha] = [w_{k1}(\alpha), w_{k2}(\alpha)]$, $k = 0, \ldots, M$, $0 \le \alpha \le 1$. Then we know how to get the α-cuts of the $\overline{w}_k$ as [5]

$$w_{k1}(\alpha) = min\{F_k(\lambda, \mu, c, M) \mid \lambda \in \overline{\lambda}[\alpha], \mu \in \overline{\mu}[\alpha]\}, \tag{16.14}$$

and

$$w_{k2}(\alpha) = max\{F_k(\lambda, \mu, c, M) \mid \lambda \in \overline{\lambda}[\alpha], \mu \in \overline{\mu}[\alpha]\}, \tag{16.15}$$

for all k and α. Let us change equations (16.14) and (16.15) to F_k a function of ρ, c and M. Let $\overline{\lambda}[\alpha] = [\lambda_1(\alpha), \lambda_2(\alpha)]$ and $\overline{\mu}[\alpha] = [\mu_1(\alpha), \mu_2(\alpha)]$. Then $\overline{\rho} = \overline{\lambda}/\overline{\mu}$ so that $\overline{\rho}[\alpha] = [\lambda_1(\alpha)/\mu_2(\alpha), \lambda_2(\alpha)/\mu_1(\alpha)]$. Then

$$w_{k1}(\alpha) = min\{F_k(\rho, c, M) \mid \rho \in \overline{\rho}[\alpha]\}, \tag{16.16}$$

and

$$w_{k2}(\alpha) = max\{F_k(\rho, c, M) \mid \rho \in \overline{\rho}[\alpha]\}, \tag{16.17}$$

for all k and α. We solve these optimization problems, equations (16.16) and (16.17), using the Optimization Toolbox in MATLAB [8].

Next we find $\overline{N}$ as

$$\overline{N}[\alpha] = \{\sum_{k=0}^{M} k w_k | \ \mathbf{S}\}, \tag{16.18}$$

all $\alpha \in [0,1]$, where $\mathbf{S}$ is the statement "$w_k \in \overline{w}_k[\alpha], 0 \leq k \leq M, w_0 + ... + w_M = 1$". If $\overline{N}[\alpha] = [n_1(\alpha), n_2(\alpha)]$ then

$$n_1^*(\alpha) = min\{\sum_{k=0}^{M} k w_k | \ \mathbf{S}\}, \tag{16.19}$$

and

$$n_2^*(\alpha) = max\{\sum_{k=0}^{M} k w_k | \ \mathbf{S}\}. \tag{16.20}$$

Then since $n_i^*(\alpha)$, $i = 1, 2$, could exceed M we set

$$n_1(\alpha) = min\{n_1^*(\alpha), M\}, \tag{16.21}$$

and

$$n_2(\alpha) = min\{n_2^*(\alpha), M\}, \tag{16.22}$$

Equations (16.19) and (16.20) are linear programming problems which can be solved using the Optimization Toolbox in MATLAB.

Now we can compute the fuzzy profit $\overline{\Pi}$. We will use α-cuts and interval arithmetic because in this case it produces the same result as the extension principle. Let $\overline{\Pi}[\alpha] = [\pi_1(\alpha), \pi_2(\alpha)]$. Also let $\overline{T}[\alpha] = [t_1(\alpha), t_2(\alpha)]$, $\overline{N}[\alpha] = [n_1(\alpha), n_2(\alpha)]$, $\overline{\lambda}[\alpha] = [\lambda_1(\alpha), \lambda_2(\alpha)]$, $\overline{Q}[\alpha] = [q_1(\alpha), q_2(\alpha)]$, $\overline{\mu}[\alpha] = [\mu_1(\alpha), \mu_2(\alpha)]$. Then

$$\overline{\Pi}[\alpha] = [t_1(\alpha)n_1(\alpha), t_2(\alpha)n_2(\alpha)] - [s, t], \tag{16.23}$$

where

$$[s, t] = [K_2\lambda_1(\alpha) + q_1(\alpha)(M - c) + K_1\mu_1(\alpha)c, K_2\lambda_2(\alpha) + q_2(\alpha)(M - c) + K_1\mu_2(\alpha)c], \tag{16.24}$$

and

$$\overline{\Pi}[\alpha] = [t_1(\alpha)n_1(\alpha) - t, t_2(\alpha)n_2(\alpha) - s]. \tag{16.25}$$

16.4 Fuzzy Monte Carlo Method

We plan to produce (approximate) solutions to the fuzzy optimization problem in equation (16.6) using our fuzzy Monte Carlo method. We will randomly generate fuzzy vectors

$$\overline{V_k} = (\overline{\lambda}_k, \overline{\mu}_k, c_k, M_k), \tag{16.26}$$

with $\overline{\lambda}_k, \overline{\mu}_k \in [0, 10]$, integer c_k in $[1, 10]$ and integer M_k in $[c_k, 100]$, for $k = 1, 2, \dots, P$. We evaluate fuzzy profit $\overline{\Pi}_k$ for each $\overline{V}_k$ and find the $\overline{V}_k$ to maximize $\overline{\Pi}_k$. With $P = 100,000$ we should get a good estimate of maximum fuzzy profit. So we need to do two things: (1) describe how to get random sequences of the vectors $\overline{V}_k$; and (2) how we will determine the maximum of the set $\Lambda = \{\overline{\Pi}_k \mid k = 1, 2, \dots, P\}$. We first consider $\overline{V}_k$ and how to produce the fuzzy values for λ and μ and then separately generate c and M. Then we discuss finding the maximum of Λ.

16.4.1 Random Sequence $\overline{V}_k$

To obtain random sequences $\overline{V}_{k1} = (\overline{\lambda}_k, \overline{\mu}_k)$, $k = 1, 2, ..., P$, where the $\overline{\lambda}_k$ and $\overline{\mu}_k$ are triangular fuzzy numbers, we first randomly generate crisp vectors $v_k = (x_{k1}, ..., x_{k6})$, using our Sobol quasi-random number generator (Chapter 3), with all the x_{ki} in $[0, 1]$, $k = 1, 2, .., P$. We choose the first three numbers in v_k and order them from smallest to largest. Assume that $x_{k3} < x_{k1} < x_{k2}$. Then the first triangular fuzzy number $\overline{\lambda}_k = (x_{k3}/x_{k1}/x_{k2})$. Continue with the next three numbers in v_k making $\overline{\mu}_k$. However the $\overline{\lambda}_k$ and $\overline{\mu}_k$ we want need to be in $[0, 10]$. Since $\overline{\lambda}_k$ and $\overline{\mu}_k$ start out in $[0, 1]$ we may easily map them into $[0, 10]$ by using $10\overline{\lambda}_k$ ($10\overline{\mu}_k$) for fuzzy λ (μ).

Next we need to randomly get the sequence of integers $c_k \in [1, 10]$. Randomly generate $\nu \in [0, 1]$ and then define $\pi = 0.5 + 10\nu$ making $\pi \in [0.5, 10.5]$. Then round π off to the nearest integer producing c. We round 0.5 to one and 10.5 to 10. Use a pseudo-random number generator for ν. Finally randomly produce $\xi \in [0, 1]$ and then define $\sigma = (c - 0.5) + (31 - c)\xi$ putting $\sigma \in [c - 0.5, 30.5]$. Round σ off to the nearest integer giving M. Also use a pseudo-random number generator for ξ.

16.4.2 Maximum of Fuzzy Profit

Given a finite set of fuzzy numbers $\overline{\Pi}_1, ..., \overline{\Pi}_P$ we want to order them from smallest to largest. For a finite set of real numbers there is no problem in ordering them from smallest to largest. However, in the fuzzy case there is no universally accepted way to do this. There are probably more than 50 methods proposed in the literature of defining $\overline{U} \leq \overline{V}$, for two fuzzy numbers $\overline{U}$ and $\overline{V}$.

Here we will use only one procedure for ordering fuzzy numbers which is Buckley's Method in Section 2.6.1. We will now use $\eta = 0.9$ in Buckley's Method to help reduce the number of fuzzy profits that could be considered approximately equal for the maximum fuzzy profit. But note that different definitions of $\leq$

between fuzzy numbers can give different orderings and therefore different final answers to the fuzzy optimization problem.

Now apply this to $\overline{\Pi}_1, \ldots, \overline{\Pi}_P$. These are all triangular shaped fuzzy numbers. We want H_K the set of undominated fuzzy profits. We then present H_K, together with the corresponding values for $\overline{\lambda}$,$\overline{\mu}$, c and M, to management for their decision. It is usually better to present multiple optimal solutions then one unique optimal solution. Managers are decision makers and if you give then one optimal solution they essentially have almost no decision: accept it or reject it. However, with multiple solutions the manager can study them, bringing in new information etc., to make their final decision. However, with $P = 100,000$ H_K could be too large, like $100 - 200$ fuzzy sets. We need to restrict the size of H_K and in this chapter we decide that the maximum size of H_K will be three fuzzy sets. Whenever we have more than three fuzzy sets to be in H_K we pick the three with largest vertex points. Therefore, the maximum of Λ will be H_K.

Now we need to incorporate this into the iterations in our fuzzy Monte Carlo method. Suppose at some point in the iterations $H_K = \{\overline{\Pi}_a, \overline{\Pi}_b, \overline{\Pi}_c\}$. The next iteration produces fuzzy profit $\overline{\Pi}_0$. We then compare $\overline{\Pi}_0$ to $\overline{\Pi}_i$, for $i = a, b, c$. There are nine possible outcomes. For example $\overline{\Pi}_0 \approx \overline{\Pi}_c$, $\overline{\Pi}_0 > \overline{\Pi}_b$ and $\overline{\Pi}_0 \approx \overline{\Pi}_a$ is one possible result. Then $H_K = \{\overline{\Pi}_0, \overline{\Pi}_a, \overline{\Pi}_c\}$. Any fuzzy set which is dominated can not be in H_K and all the fuzzy sets in H_K must be equivalent ($\approx$).

16.5 Fuzzy Monte Carlo Solution

We will describe our fuzzy Monte Carlo program. The program is written in MATLAB. We first generate $\overline{V}_k$, $k = 1, \ldots, P$. We compute the fuzzy numbers using the α-cuts $\alpha = 0.00, 0.30, 0.60, 0.90, 1.00$. We have our first file

$$\mathcal{F}_1 = \{(\overline{\lambda}[\alpha], \overline{\mu}[\alpha], c_k, M_k) \mid \alpha = 0, 0.30, 0.60, 0.90, 1;\ k = 1, \ldots, P\}. \quad (16.27)$$

Using this file we determine the fuzzy steady-state probabilities $\overline{w}_{jk}$, $j = 0, \ldots, M$, $k = 1, \ldots, P$, from equations (16.16) and (16.17), using the Optimization Toolbox. This makes our second file

$$\mathcal{F}_2 = \{(\overline{w}_{0k}[\alpha], \ldots, \overline{w}_{M,k}[\alpha]) \mid \alpha = 0, 0.30, 0.60, 0.90, 1;\ k = 1, \ldots, P\ \}. \quad (16.28)$$

Using file $\mathcal{F}_2$ we determine $\overline{N}_k[\alpha]$ from equations (16.19)-(16.22) using the Optimization Toolbox. This produces file

$$\mathcal{F}_3 = \{\overline{N}_k[\alpha] \mid \alpha = 0, 0.30, 0.60, 0.90, 1;\ k = 1, \ldots, P\ \}. \quad (16.29)$$

Now we are ready to find fuzzy profit $\overline{\Pi}_k[\alpha]$, $\alpha = 0, 0.30, 0.60, 0.90, 1$, $k = 1, \ldots, P$, from equations (16.23)-(16.25).

The results of the Monte Carlo method, after $P = 100,000$ iterations, gave us the three (approximate) optimal solutions shown in Table 16.2. The graph of the three maximum fuzzy profits in the final H_K are shown in Figure 16.1. For

Table 16.2. Optimal Numerical Results from the Fuzzy Monte Carlo Method

Solution	$\overline{\lambda}$	$\overline{\mu}$	c	M	$\overline{\Pi}$
1	(4.81/5.85/6.59)	(2.40/3.28/4.00)	1	30	$\approx$ (16.83/96.75/112.54)
2	(4.52/7.16/7.46)	(3.25/4.18/4.18)	1	30	$\approx$ (−0.62/96.25/108.77)
3	(7.96/9.72/9.78)	(1.91/2.33/2.36)	2	29	$\approx$ (83.57/94.51/103.18)

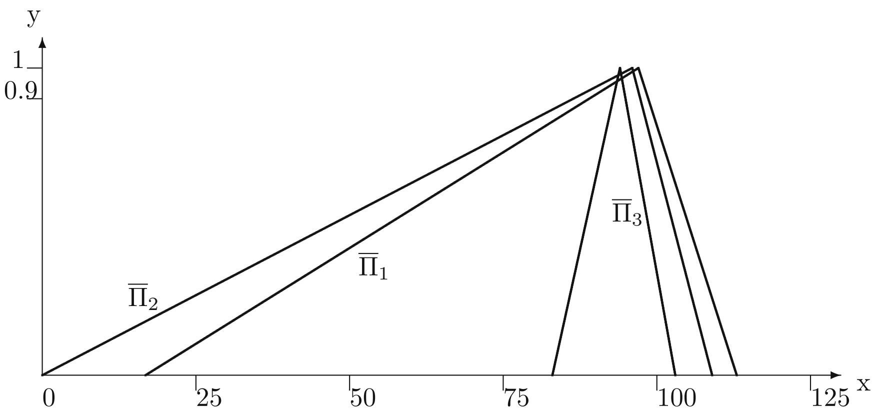

Fig. 16.1. Optimal Fuzzy Profits from the Fuzzy Monte Carlo Method

simplicity the graphs in Figure 16.1 are triangular fuzzy numbers, using only their base and vertex, where they are really triangular shaped fuzzy numbers.

Let us now compare the results above for our fuzzy Monte Carlo method to those in Chapter 14 of [4]. Both basically consider the same problem of maximizing (almost the same) fuzzy profit with variables fuzzy arrival rate $\overline{\lambda}$, fuzzy service rate $\overline{\mu}$, number of servers c and system capacity M. Because of the computational burden of calculating fuzzy profit in [4] the author only considered 16 cases. For example, $\overline{\lambda} = (4/5/6)$, $\overline{\mu} = (5/6/7)$, $c = 2$, $M = 10$ was one of the cases. The author of [4], at that time, could not look at $P = 100,000$ random cases. In [4] $H_K = \{\overline{\Pi}_a, \overline{\Pi}_b, \overline{\Pi}_c\}$ where $\overline{\Pi}_a \approx (-0.8/0.7/4.6)$, $\overline{\Pi}_b \approx (-0.5/1.9/6.3)$ and $\overline{\Pi}_c \approx (-0.8/0.9/4.3)$. The graphs of these fuzzy profits, as triangular fuzzy numbers, are shown in Figure 16.2 for comparison to Figure 16.1. Now let us briefly summarize the model in Chapter 14 of [4] to compare to the model in this chapter.

The fuzzy profit function in [4] was

$$\overline{\Pi} = \overline{TN} - [\overline{A}_u + \overline{Q}(M - c) + \overline{C}_v c]. \tag{16.30}$$

In [4] the author considered only two fuzzy arrival rates $\overline{\lambda}_i$ and only two fuzzy service rates $\overline{\mu}_i$, $i = 1, 2$, in their 16 cases. Then $\overline{A}_u = \overline{A}_i$ when $\overline{\lambda} = \overline{\lambda}_i$, $i = 1, 2$. In this chapter we used $K_2\overline{\lambda}$ since we could not use $P = 100,000$ different constants for that many different fuzzy arrival rates. Also, $\overline{C}_v = \overline{C}_i$ when $\overline{\mu} = \overline{\mu}_i$, $i = 1, 2$.

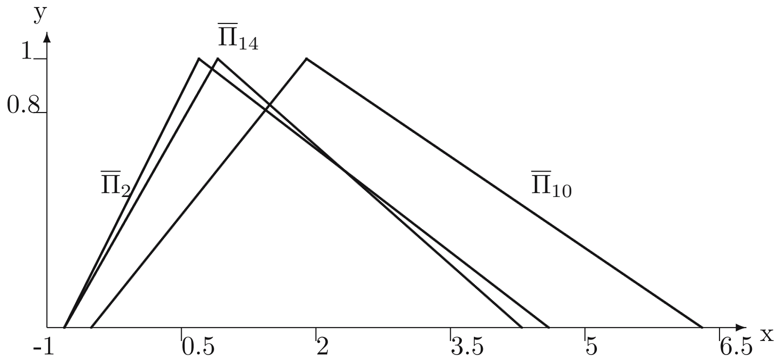

Fig. 16.2. Optimal Fuzzy Profits from Chapter 14 of [4]

In this chapter we used $K_1\overline{\mu}c$ since we could not use $P = 100,000$ different constants for that many different fuzzy service rates. The $\overline{Q}(M-c)$ is the same in both models. In this chapter our $\overline{T}$ is greater then the fuzzy T value used in [4] because the $\overline{\Pi} \approx (\pi_1/\pi_2/\pi_3)$ in [4] had too many results with negative π_2 values. Finally, in [4] they used $\eta = 0.8$ in the comparison of fuzzy numbers and in this chapter we used $\eta = 0.9$ to help in reducing the size of the set H_K.

We see from Figures 16.1 and 16.2 that the results are similar: three fuzzy sets clustered together approximating maximum fuzzy profit. However, those in Figure 16.1 would present a better approximation since they are the result of 100,000 random choices for the variables. The fuzzy profits in Figure 16.1 will all lie to the right of those in Figure 16.2 because $\overline{T}$ used in this chapter is approximately three plus the $\overline{T}$ used in Chapter 14 of [4].

16.6 Summary and Conclusions

In this chapter we introduced our new fuzzy Monte Carlo procedure. The basic requirement of any fuzzy Monte Carlo method is to be able to randomly produce fuzzy/crisp vectors to uniformly fill the search space. We suggested using a quasi-random number generator to make these random fuzzy/crisp vectors. Theoretically, given enough iterations of the fuzzy Monte Carlo technique, it will produce a very good approximate solutions to the fuzzy optimization problem.

We applied our fuzzy Monte Carlo method to a fuzzy optimization problem from fuzzy queuing theory. For all our Monte Carlo calculations, we used a Dell Optiplex GX 250 with a dual core and a 64-bit pentium D 2.8 GHz processor running on Windows XP. The computer time for 100,000 iterations was approximately 52.5 hours. To construct file $\mathcal{F}_2$ (equation (16.29)) using the Optimization Toolbox the computer time was approximately 47 hours and then to finish the program it was approximately 5.5 hours.

It would be nice to try 1,000,000 iterations, but the computing time on one office PC would be too excessive. However, if we could run the fuzzy Monte Carlo program 100,000 iterations simultaneously on ten separate machines, computing over the weekend, and then combine the results, we could go to 1,000,000 iterations. The MATLAB program is available from the authors.

References

1. Abdalla, A., Buckley, J.J.: Monte Carlo Methods in Fuzzy Queuing Theory (under review)
2. Buckley, J.J.: Elementary Queuing Theory Based on Possibility Theory. Fuzzy Sets and Systems 37, 43–52 (1990)
3. Buckley, J.J.: Fuzzy Probabilities: New Approach and Applications. Springer, Heidelberg (2003)
4. Buckley, J.J.: Fuzzy Probabilities and Fuzzy Sets for Web Planning. Springer, Heidelberg (2004)
5. Buckley, J.J., Qu, Y.: On Using α-cuts to Evaluate Fuzzy Equations. Fuzzy Sets and Systems 38, 309–312 (1990)
6. Buckley, J.J., Eslami, E., Feuring, T.: Fuzzy Mathematics in Economics and Engineering. Physica-Verlag, Heidelberg (2002)
7. Buckley, J.J., Feuring, T., Hayashi, Y.: Fuzzy Queuing Theory Revisited. Int. J. Uncertainty, Fuzziness and Knowledge Based Systems 9, 527–538 (2001)
8. MATLAB, The MathWorks, `http://www.mathworks.com`
9. Menasce, D.A., Almeida, V.A.F.: Capacity Planning for Web Performance. Prentice Hall, Upper Saddle River, N.J. (1998)
10. Pardo, M.J., de la Fuente, D.: Optimizing a Priority-Discipline Queueing Model Using Fuzzy Set Theory. Computers & Math. with Applications 54, 267–281 (2007)
11. Taha, H.A.: Operations Research, 5th edn. Macmillan, N.Y. (1992)