CHAPTER - III

SIMULATION

TECHNIQUES IN QUEUING MODELS
CHAPTER III

SIMULATION TECHNIQUES IN QUEUING MODELS

3.1 Introduction

The computer simulation is a method that demonstrates dynamically the structure and the behaviors of a system with computer in order to evaluate and predict the effect of the behaviors of some system and provide information for decision. It is an effective way to solve complicated practical problems. The queuing system is the most typical problem in the discrete event system, computer system, communication system and transportation system are all typical tangible or intangible queuing system. As a result of the widely used [57, 64-65] queuing system, the queuing character, the queuing regulation, the service organization become more and more complex.

The computer simulation is a quite effective way for solving the queuing problem and analyzing the performances of the queuing system, which construct a real system model with computer program, and attain the performances and the characters changing with time through computation. With the computer simulation, the cost of the development of the system can be reduced, and the safety of the experiment and the debugging.

In the queuing model, data units are often considered as customers and CPUs, transmission lines, channels and terminals are thought as queue. This is just a queuing model. Because input, computation, transmission, storage and output are discrete in time field, it is also called discrete queuing model. In real life, waiting in queue is a common phenomenon, which makes people inconvenient.
In this chapter, the payment of mini type supermarket is taken as an example to discuss the computer simulation for the single-channel and multi-channel queuing system.

3.2 Some Concepts In The Queuing System

The queuing problem is a problem about a balance between average waiting time and idle time of server, i.e. how to queue to be both good for entity and server. The queuing theory is a science to solve the problem mentioned as above, and it is also named random serve theory since the arrival time of entities and the time of serve acceptance are usually random variable obeying some probability distribution. In the simulation of the queuing system, there are some conceptions usually being used as below:

**Entity Arrival Mode:**

The entity is limited or limitless, and the arrival of the entity is in individual or in batch. Entity arrival mode is often described with arrival interval. The random arrival mode applied in the system appears very complex, and different probability distributions have to be adopted for different systems. Index distribution, normal distribution, Poisson distribution and etc are quite common. Poisson distribution arrival is provided as below:

In \( (t, t+s) \), the probability of entity number \( k \) is

\[
P\{N(t+s)-N(t)=k\} = \frac{e^{-\lambda s} (\lambda s)^k}{k!}
\]

In the formula, \( N(t) \) is the number of entity arrival in \( (0,t) \), \( t \geq 0, s \geq 0, k=0,1,2,\lambda \) is the arrival velocity.

If the entity arrival satisfies steady Poisson distribution, arrival interval will obey Index distribution and density function is:
f(t) = \lambda e^{-\lambda t} = \frac{1}{\beta} e^{-\frac{t}{\beta}}, t \geq 0, \beta = \frac{1}{\lambda} \text{ is the average value of arrival interval.}

**Service Mode**

Its character is that its server may be single or multiple, and service time distribution is nothing about time or something about time, and server's service time is certain or random. Random service time is described with probability distribution, for instance Normal distribution,

\[ f(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < \mu < \infty, \sigma > 0 \]

In the equation above, t is the time of server for each custom, which is obeying Normal distribution, and the average value is \( \mu \), the variance is \( \sigma^2 \).

**The Queuing Rule and the Criteria of the Queuing System:**

There are some queuing rules such as FCFS, Random served, priority served and LCFS, etc. With studying the performances of the queuing system, some criteria usually used is as below:

Steady-state means delaying time \( d \):

\[ d = \lim_{n \to \infty} \sum_{i=1}^{n} \frac{D_i}{n} \]

In the equation, \( D_i \) means NO. I entity's delaying time, i.e. waiting time in the queue; \( n \) is the number of the accepted entities; \( d \) is the mean time of waiting time \( f \) the \( n \) entities. The staying time of the entity in the system \( w \):

\[ d = \lim_{n \to \infty} \sum_{i=1}^{n} \frac{D_i}{n} = \lim_{n \to \infty} \sum_{i=1}^{n} \frac{(D_i + S_i)}{n} \]
In the equation as above, \( W_t \) is the staying time of No. 1 entity in the system, and equals to the sum of waiting time in the queue \( D_t \) and accepting service time \( S_t \).

Steady-state means step-length \( Q \):

\[
Q = \lim_{T \to \infty} \int_0^T \frac{Q(t)dt}{T}
\]

In the formula, \( Q(t) \) is the length of the queue at \( t \), and \( T \) is the simulate time of the system. Steady-State entity mean number \( L \):

\[
Q = \lim_{T \to \infty} \int_0^T \frac{L(t)dt}{T} = \lim_{T \to \infty} \int_0^T \frac{Q(t)+S(t)dt}{T}
\]

In the formula, \( L(t) \) is the number of the entity in the system at \( t \), and equals to \( Q(t) \) and \( S(t) \).

3.3 Single Channel M/M/1 model

The Construction of Model

In the mini-type super-market, there is a cash desk, and customers reaches the desk in random. Provided that the customer reaches as the cashier is idle, the customer will pay off immediately and leave. If the cashier is busy as the customer reaches, the customer will have to wait in the line, say, no person leaves without waiting. Once the customer enters in the queue, he will receive service according to FCFS rule. The customer departs after receiving once service. The interval of customers arriving desk obeys negative index distribution with average value equaling to 5, and service time of each customer complies with normal distribution with average value being 1.6 and standard deviation being 0.6. Time calculates at minute, and service time must be positive.
The Simulation of Model

1. **The creation of random number.** It is desired to describe random factors in the objective process in nearly all of the simulation process like arrival process and service process in actual system. Random number comes from collectivity in random. In this model, there are the interval of customers' arrival and the service time of each customer. The former obeys negative index distribution with average number being 2.5, and the latter obeys normal distribution with average value being 1.6 and standard deviation being 0.6. The symmetrical-distribution random number $U(0, 1)$ must create ahead of the creation of specific-distribution random number.

**a.** Creating the algorithm of obeying negative index distribution with Inversion of The density function of negative index distribution:

$$F(x) = \lambda e^{-\lambda x}, x \geq 0, E(X) = \frac{1}{\lambda}$$

Its distribution function:

$$F(x) = \int_0^x \lambda e^{-\lambda x} \, dt = 1 - e^{-\lambda x}, x \geq 0, i.e.$$  

$$R = F(X) = 1 - e^{-\lambda x}$$

Through inversion transform, we can obtain:

$$x = -\frac{1}{\lambda} \ln(1-R) \text{ Let } u=1-R$$

Thus $u$ is a random number in $(0, 1)$($R$ obeys symmetrical distribution in $(0,1)$) Therefore

$$x = -\frac{1}{\lambda} \ln u, x$$

is a random number obeying negative index distribution. With this method, as long as producing a random number obeying even
distribution, we could get a random number \( x \) obeying the negative index number distribution with parameter \( \lambda \)

**b. Creating the algorithm of obeying normal distribution with discarding-selecting method**

1. Create two random number \( R_1 \) and \( R_2 \) in \((0,1)\)
2. With \( R_1 \), create a random number in \([a,b] \) \( f(x)=a+(b-a)R_1 \), and calculate the value of \( f(x) \)
3. when \( R_2 \leq f(x)/M \), select \( x \) as, and \( M \) is the max value of \( f(x) \)

2. Simulating the model with event step-length method

Event step-length method takes the time of event as increment, and simulates the behaviors of the system according to the process of time until the scheduled time ends. In some degree, the process of the simulation is considered as a series of event point. In terms of the sequence of event, the system arranges the sequence of event executing with a table called “event table”. Event table has three properties, ID, event style and time.

The simulation algorithm given below:

**Step1** Initiation (clear simulation clock, configuring the state of the system, clear accumulation statistic, creating initiating event table)

**Step2** Do

{Find most recent event from event table

Simulation clock increase

If (Customers arrive)

Add the time of next customer arriving into event table

If (cashier is busy/ S=1)

LQ++ //queue increment

Else

S=1 //cashier turn to busy

Add the time of the customer finishing into event table
Else
    Delete the customer from event table //service ends
If (the queue is not empty/ LQ>0
    LQ-- //the length of queue decrease
    Add the time of the customer finishing into event table
Else
    S=0 // cashier is idle
} While (event table is not null)

Output result

4. Ascertaining the times of repeating simulation for reaching the precision $\beta$.

A Repeat simulation $R_0 (R_0 \geq 2)$ times independently, and set $R=R_0$
B Compute $\bar{Y}(R), S^2(R)$ and absolute procession $\theta(R, \alpha) \bar{Y}(R), S^2(R)$ are sample average value and variance after $R$ times running, and $\theta(R, \alpha)$ is half length of the Confidence Interval of $R$ times running under significance degree.
If $\theta(R, \alpha) \leq \beta$, C.I($\alpha, \beta$)=\{\bar{Y}(R)-\theta(R, \alpha), \bar{Y}(R)+\theta(R, \alpha)\} is the Confidence Interval with Confidence coefficient (1-$\alpha$), and stop simulation,
$\bar{Y}^{\theta(R, \alpha)} > \beta$ set $R=R+1$ and run $R+1$ times simulation, return to (2) step.
C. The Analysis of the Simulation Result: - 50 persons each group and repeat 50 times simulation. Get result as below
RESULT OF THE SIMULATION

<table>
<thead>
<tr>
<th>Satisfaction degree</th>
<th>Customers average waiting time</th>
<th>1m55s</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The average length of the queue</td>
<td>4.22</td>
</tr>
<tr>
<td>The parameter of cash desk</td>
<td>The probability of cash desk’s idleness</td>
<td>35.65%</td>
</tr>
</tbody>
</table>

Since system simulation is final state simulation, we adopt fixed sample method to analyze. Customers average waiting time $\bar{Y} = 115s$, sample variance $=5117.286$. $S^2=5117.286$ with $\alpha = 0.1$, 90% Confidence interval $(97,133)$ Units: second.

**D.** The final state of the simulation lasts shorter and the performances of the system is obviously determined by the initial state. Ordinarily the system cannot reach final state and the influence cannot be diminished until the simulation ends. In order to make the result of the simulation close to fact, the initial state must be cautious to select.

*N.B. Please see Appendix – A & B for program*

### 3.4 Multi-Server M/M/C Model

When the supermarket is in shopping fastigium, servers should have to be increased. Next, Multi-server M/M/C model will be studied.

**A. The Construction of the Model:** - In the system C servers are parallel-connected. There will be n customers reaching the system at the same time and the interval of customers reaching and the time of accepting service are random. Customers will select shortest queue to
stand in after reaching the system. If all the queues are of same length, customers will enter according to the ID of queue. Customers have to leave after once service. Too many servers or too few servers also lead to increasing spending.

Too few servers will make customers wait too long time with loss increasing. Too many servers will be a waste of human power and material power. So we must optimize the number of the servers. We use $h$ to stand for service spending each service window and each unit time, and $W$ to stand for spending of customer’s linger in the system for each unit time, and $c$ is the number of servers. Otherwise $L$ is the length of the queue as the number of servers is $c$ (get from the result of the simulation). Thus total spending is $f(n)=hn+WL(n)$. For the optimized $n$, we can adopt boundary-analysis method, namely make $n$ satisfy with two conditions:

1. $f(n)<f(n-1)$
2. $f(n)<f(n+1)$. 
Algorithm for mini type super market
Report Logic using Pseudocode

(Pseudocode ordering products from supplier)_

Begin
Numeric nItem Price, nTotal Price, nTotal Item, cProduct Name
nSum = 0
While // i >= 1
Begin
Display “enter the product name”
Accept Product name (i)
Display total Item
Accept Total Item (i)
Display Item price
Accept Item price (i)
nSum = sum + Item Price(i) * Total Item
i = 1+i
End
Display “total Payment is” nSum
End
End
(Pseudocode accounting cash back of customer money)

Begin
Numeric nItem Price, nTotal Price, nTotal Item, cProduct Name
Display enter customer money
Accept customer money
While customer money < total price
Begin
Display your money is not enough
Display "customer money"
Accept customer money
End
nChange = customer money - bill
Display "nChange"
End
(Pseudocode billing account in cashier)

Begin

Numeric nItem Price, nTotal Price, nTotal Item, cProduct Name

nSum = 0

While // i >= 1

Begin

Display “enter the product name”

Accept Product name (i)

Display total Item

Accept Total Item (i)

Display Item price of selling

Accept Item price of selling (i)

Calculate = sum + Item Price (i) * Total Item

i = 1+i

End

Display “Total Payment is” nSum

End

End