Discrete-Event System Simulation of Queues with Spreadsheets Combined with Simple VBA Code: A Teaching Case

การจำลองระบบคิวแบบไม่ต่อเนื่องด้วยสเปรดชีทร่วมกับโปรแกรม VBA: กรณีใช้ประกอบการสอน

นิธิภัทร กมลสุข
หัวหน้าสาขาวิชาวิทยาศาสตร์ทั่วไป
คณะวิศวกรรมศาสตร์และเทคโนโลยี สถาบันการจัดการปัญญาภิวัฒน์

Nithipat Kamolsuk
Chairperson, Department of General Science
Faculty of Engineering and Technology
Panyapiwat Institute of Management
E-mail: nithipatkam@pim.ac.th

บทคัดย่อ

บทความนี้อธิบายถึงการใช้สเปรดชีทร่วมกับโปรแกรม VBA เพื่อสร้างแบบจำลองระบบคิวแบบไม่ต่อเนื่องในซุเปอร์มาร์เก็ตหรือร้านสะดวกซื้อ โดยเลือกการจำลองระบบการทำงานในร้าน 7-eleven เป็นกรณีศึกษาสำหรับการสอนที่สถาบันการจัดการปัญญาภิวัฒน์

คำสำคัญ: การจำลองแบบไม่ต่อเนื่อง ระบบคิว ร้านสะดวกซื้อ

Abstract

This paper describes the use of spreadsheets combined with simple VBA code to simulate the queuing system in supermarkets or covenience stores. 7-eleven was selected in this case for teaching in Panyapiwat Institute of Management.

Keywords: Discrete-Event Simulation, Queuing System, Covenience Store

Introduction

The objective of queuing analysis is to offer a reasonably satisfactory service to waiting customers, because waiting for service is part of our daily life, such as waiting for seats in a restaurant, paying bills in supermarkets and waiting for money service in banks. Besides, the waiting phenomenon is not an experience limited to human beings only. For example, work waits to be processed on a machine, planes circle in stack before waiting time in a queue, and average facility utilization. The results of queuing analysis can be used in the context of a cost optimization model, where we seek the minimization of the sum of two costs: the cost of offering the service and the cost of waiting. Figure 1 depicts a typical cost model where the costs of service increase with the level of service (e.g. the number of service counters). At the same time, the cost of waiting decreases with the increase in level of service. The main obstacle in implementing cost models is the difficulty of obtaining reliable estimates of the cost of waiting, particularly when human behavior is an integral part of the operation. (Taha, 2007)

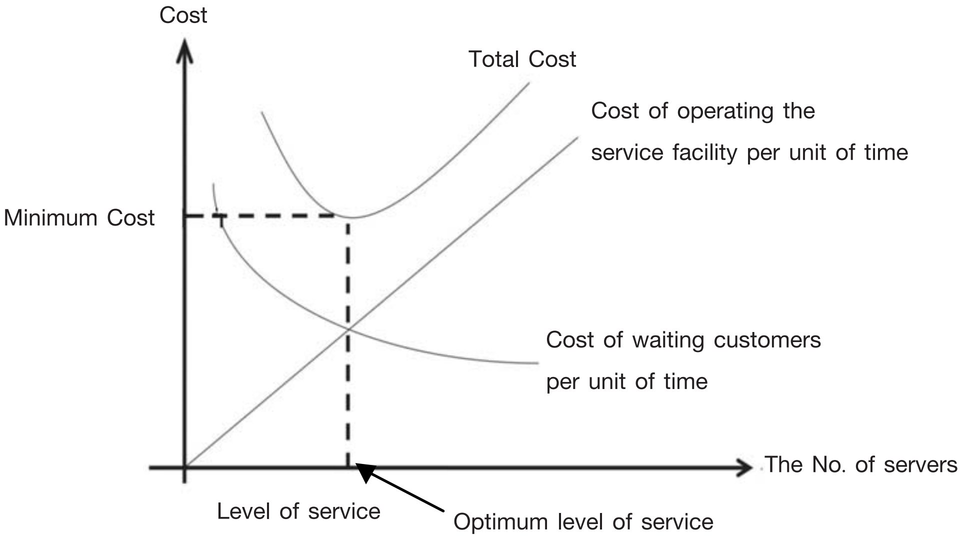

Figure 1 Cost-Based Queuing Decision Model

Discrete-event-simulation is designing a model that resembles a real system in certain important aspects. It can be viewed as the act of performing experiments on a model of a given system. It is usually taught by means of some dedicated simulation software.

Both queuing theory and discrete-event system simulation approaches have been criticized. The former would be very mathematical and "closed-form models would not be able to analyze most of the complex systems encoutered in practice" (Banks, et al., 2010). The letter would be prone to emphasize specific simulation software features instead of discrete-event simulation core concepts (Schriber and Brunner, 1998).

This paper presents a way to improve the discrete-event simulation of simple queuing systems deployed through the combined usage of native spreadsheet functions and simple Visual Basic for Applications (VBA code). The results are used to teach Quantitative Analysis for Business classes at Panyapiwat Institute of Management. The students apply problem-based learning by working at 7-eleven, so they can apply the problems from working to the case study in the class.

Background

The discrete-event system simulation is the modeling of systems in which the stated variables change only at a discrete set of points in time. The simulation models are analyzed by numerical rather than analytical methods. Analyzed methods employ the deductive reasoning of mathematics to "solve" the model. Banks, et al. (2010) and Centeno (1996) explain the two main approaches for discrete simulation (event-driven and process driven). Under event driven discrete simulation, to describe the model the modeler has to think in terms of an event that may change the status of the system. On the other hand, under the process driven approach, the modeler thinks in terms of the processes that the dynamic entity will experience as it moves to the system. Sadiku and Tofighi (1999) explain the two major of discrete-event simulation: trace-driven and event-driven. In a trace-driven simulation model, the model is advanced by an input sequence derived from trace data and event-driven simulation model. The model is advanced by a random sequence and the two methods can be used for time control.

Most widely used queuing models for which relatively simple closed analytical formulars are specified as M/M/s type (Hall, 1990; Lawrence and Pasternack, 2002). First "M" represent Markov since Poisson process is a particular case of a stochastic process with no "after-effect" or no memory, known as continuous time Markov process.

In the simulation of the queuing system, there are some conceptions usually being as the entity arrival mode and service mode. Entity arrival mode is often describes with arrival interval. The random arrival mode applied in the system appears very complex, and different probability distribution. In Poisson process, the distribution of the arrival rate is Poisson distribution. It can be established by using the formula

$$P(X) = \frac{e^{-\lambda}\lambda^{X}}{X!} \text{ for } X = 0,1,2,3,4,\ldots$$

Where $P(X)$ = probability of X arrivals

X = numbers of arrivals per unit of time

λ = average arrival rate

e = 2.7182… (Render, Star, and Hanna, 2009)

For the service mode, its character is described as certain or random service. The random service time is described with probability distribution For instance, exponential distribution, which is defined as:

$$f(t) = \lambda e^{-\lambda t} \;,\; t > 0$$

Where t = service time

According to Lawrence and Pasternak (2002), queuing models allow calculating the following steady-state characteristics an unlimited queue size served by possible providers:

- Probability that there are zero customers in system
- Probability that there are K customers in the system
- The average number of customers waiting in the queue
- The average time customers wait in the queue
- The average total time the customer spends in system
- The utilization rate of servers, i.e. the percentage a server is available

As more complexity is added into the system, the analytic formulas become less and less tractable. Available analytic formulas include, for example, limited queue size, entities leaving the system after waiting a specified amount of time, multiple queues with different average service time and different provider types, different service priorities, etc. (Hall,1990; Lawrence and Pasternack, 2002). Several authors have written about the use of the process-driven approach based on spreadsheets to teach queuing theory. Chase, Anquilano, and Jacobs (1998) and Grossman (2002), used process-driven approaches, implement queuing systems using only native speadsheet functions, and obtaining the respective ‘ad hoc simulation’ tables. The former simulates tandem queues, the latter a multi-server queuing system. Albright (2001) also used spreadsheets in multi-server queuing systems. The analyst can easily input system parameters and measures system performance measures. In such implementation, the simulation and performance evaluation are undertaken by pure VBA programming, which increases the complexity of the code. (Grossman, 2002), building on the previous work of Grossman, provides graphical interface in addition to basic tables, displays the customers and servers statuses. An interesting feature of such an approach demonstrates the posibility of replicating the simulation through the recalculating feature of Microsoft office Excel and observing the resulting changes. The graphical interface obtained in that implementation, in conjunction with the recalculating function, does provide “a richer understanding of queues”, as stated by the following authers. Mesquita and Hernandez (2006) present the use of spreadsheets combined with simple VBA code as a tool for teaching four different cases of queues: single server, parallel servers, tandem queuing and closed queuing systems.

The Basics for Generating Random Numbers

The first step in a simulation study is problem formulation, since every study should begin with a statement of the problem. The analyst must ensure that the beginning problem described is clearly understood and be aware that problem exists long before the nature of the problem is known (Banks, et al., 2010). After knowing the nature of the problem, the anlyst should collect data and define their characteristics. For example, the desire to learn about the length of waiting lines as the number of interarrival times (at different times of the day) includes the studies of service-time distributions for the tellers and historic distributions on the lengths of waiting lines under varying conditions. This last type of data will be used to validate the simulation model as well as the number of interarrival times and the service times that are variables. In order to build the queue model, these variables need to be generated. The generating of any random quantity begins with generating a random number uniformly distributed between 0 and 1. In general, the random number can be generated by any method, such as an implementation of the VB programming language in Microsoft Excel or the other programming languages. Microsoft Excel has a built-in worksheet function called RAND. In a cell formula. RAND generates values between 0 and 1. Typical examples of formulas using RAND include generating a random value between 0 and 1, as RAND( ). It also provides the worksheet function, RANDOM BETWEEN( ). The returns uniformly distribute integer values between a specified low and high value, for example: to generate random values between 3 and

5 as RANDOM BETWEEN(3,5), and in a more complex expressions, we can use function 'IF' to generate a random number as IF(RAN( )<=0.5,0,1). This function will check a condition in its first argument. If the condition is true, it will return the first value. Otherwise, it returns a second value. For the VBA functions and Excel worksheet functions for generating random values in a worksheet cell as:

1. In a worksheet cell in Excel, we can call a user-written VBA function that, in turn, calls an internal VBA library function such as Rad().

2. Discrete Uniform (min,max), to generate a random integer form the minimum value, min, to the maximum value, max, where min and max can be numbers or a single-cell range containing the numbers.

3. Discrete Emp (rCumProb, rValues), to generate samples from a discrete distribution specified by two ranges of cells, rCumProb for the cumulative probabilities, and rValues for the values.

4. Uniform(low, high), to generate a real value from the continuous uniform distribution between low and high, where low and high are numbers or a cell containing a number.

For generating random variable in Poisson process, we can simulate a Poisson process with intensity λ by simply generating the inter arrival times, X_i, where $X_i \sim$ Exp (λ). We have the following algorithm for simulating the first T time units of a Poisson process:

```
set $t = 0$, $I = 0$
generate $U$
set $t = t - \log(U) / \lambda$
while $t < T$
    Set $I = I + 1$, $S(I) = t$
    Generate $U$
    Set $t = t - \log(U) / \lambda$
```

After that, we use Run test for independence of variable and Chi-Square test for goodness of fit test.

Multi-Server Queuing Model

After generating a random number from VBA functions or an Excel worksheet, in order to model a multi-server queuing system in VBA code, one has to take into consideration two main events: the arrival event and the departure event. The basic logic involved in checking which event occurs first is shown below. The simulation runs until the following conditions are both satisfied, the closing time is reached and there are no customers left in the system.

```
Sub MMC( )
   Call Initialize
   Do
      Call NextEvent (Type, Server)
      Select Case Type
         Case 1
```

```
                Call ArrivalEvent
            Case 2
                Call Departure Event (Server)
        End Select
    Loop Until TimeTotal > CloseTime
                    And NumIn System = 0
End Sub
```

The initialization process includes reading the observed values from 7-eleven at Khwanpatthana branch at peak time (the time period which has the largest number of customers). The values are illustrated in Table 1, and consist of the first event, the arrival time and the service of the first customer. It was generated from individual distribution as a real system (i.e. Exponential, Poisson, Weibuli distribution). The procedure NextEvent determines which event will occur next, either type 1 for an arrival, or type 2 for a departure. In the case of a departure, the server from which it takes place is also determined.

Table 1 Input Data

Arrival rate (customer per min.)	2.95
Mean service time (min.)	1.49
Number of servers	2

Upon an arrival, the customer wouldn't joint the queue if servers were busy. On the other hand, if there is more than one idle server, the customer picks one of them randomly. The last step is to define the next arrival time. The code for this procedure is shown bellow.

```
Sub Arrival( )
    ArrivalTime = NextArrival
    NumInsystem = NumInsystem+1
    If NextArrival > CloseTime Then
            NextArrival = Infinite
    Exit Sub
        End If
    If NumInsystem > NumServer Then
            QueueLength = QueueLength +1
    Else
            Server = ChooseServerIdle
            ServerBusy(Server) = True
            NextDeparture = Clock+INVEXP(Mu)
    End If
    NextArrival = TimeTotal + INVEXP (Lambda)
End Sub
```

After service completion, there are two possibilities: the line is empty, in this case the server becomes idle, or there is at least one customer in the line, in which case the customer who arrived first is picked up by the server.

```
Sub Departure (Server)
        TimeTotal = NextDeparture(Server)
        NumInSystem = NumInSystem-1
        If QueueLength = 0 Then
                ServerBusy(Server) = False
                NextDeparture = Infinite
        Else
              ServerBusy(Server) = True
              QueueLength=QueueLength – 1
       NextDeparture = TimeTotal + INVEXT(Mu)
       End If
End Sub
```

The values taken on by the state-variables during the simulation are written directly to the spreadsheets. There are three main tables: event table, customer table and server table. These tables allow further statistical analysis through spreadsheet functions. Table 2 shows the event table for a given run (only ten customers). Table 3 shows the time for individual customers use of the system. Table 4 shows the server status throughout the simulation run (due to column width limitations, only data regarding one server is shown).

Table 2 The Events Table

Time	Event*	Next Arrival	Next Departure		
			Serv.1	Serv.2	Serv.3
18:00:00	Open	18:02:24	0:00:00	0:00:00	0:00:00
18:02:24	0	18:09:43	18:02:48	0:00:00	0:00:00
18:02:48	1	18:17:12	0:00:00	0:00:00	0:00:00
18:09:43	0	18:19:28	18:22:31	0:00:00	0:00:00
18:17:12	0	18:22:31	18:22:31	18:24:17	0:00:00
18:19:28	0	18:24:17	18:22:31	18:24:17	18:30:14
18:22:31	1	18:27:36	0:00:00	18:24:17	0:00:00
18:24:17	0	18:31:42	18:32:38	0:00:00	0:00:00
18:27:36	0	18:33:02	0:00:00	0:00:00	18:35:17
18:30:14	3	18:35:44	0:00:00	0:00:00	18:35:17
18:31:42	0	18:37:53	0:00:00	18:37:12	18:35:17

* Key for event column: 0 = arrival, i = departure from server i(i>0)

Table 3 The Time for Individual Customers Use of System

Customer	Arrival Time	Service Starting Time	Server	Service Finishing Time
Open	18:00:00	----	----	----
1	18:02:24	18:02:24	1	18:02:48
2	18:09:43	18:09:43	1	18:22:31
3	18:17:12	18:17:12	2	18:24:17
4	18:19:28	18:19:28	3	18:30:14
5	18:24:17	18:24:17	1	18:32:38
6	18:27:36	18:27:36	3	18:35:17
7	18:31:42	18:31:42	2	18:37:12
8	18:33:02	18:33:02	2	18:42:19
9	18:35:44	18:35:44	1	18:48:03
10	18:37:53	18:37:53	1	18:56:11

Table 4 Shows the Server Status

Time	Status	Customer
18:00:00	Idle	----
18:02:24	Busy	1
18:09:43	Busy	2
18:17:12	Idle	----
18:19:28	Idle	----
18:24:17	Busy	5
18:27:36	Idle	----
18:31:42	Idle	----
18:33:02	Idle	----
18:35:44	Busy	9
18:37:53	Busy	10

Data in Tables 1, 2 and 3 show Gantt's chart of waiting and service times for ten customers in Figure 1.

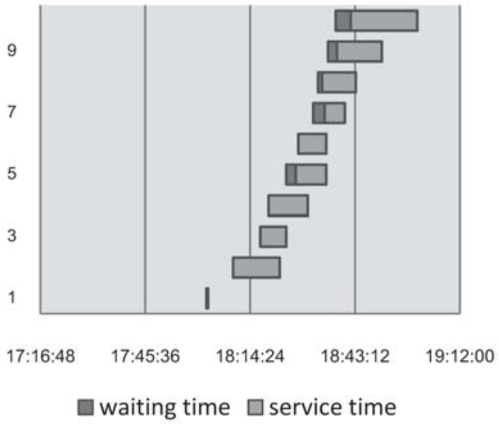

Figure 1 First Ten Customers Waiting and Service Times

A Teaching Case

The parts of quantitative analysis for business in an undergraduate student class in Panyapiwat Institute of Management are queuing theory and discrete-event system simulation. The first step is used to teach classic concepts on Markov chain and on queuing theory. The next step comprises teaching discrete-event simulation and part of the dynamic programming and decision analysis's dedicated. The discrete-event simulation model begins with the simulation of the classic M/M/1. The first approach adopted is conceptual, by hand process-driven simulation (ad hoc simulation). After the students understand this concept, the implementation on a Microsoft Excel/VBA began with a workshop. Following these activities, the students are required to develop and implement the multi-server. An interesting result is achieved when the exponential service time distribution is replaced by other types of distribution. For instance, through the use of normal distribution, the effect of the service time variance on performance measures can be easily noticed. For these activities, the students would simulate the real systems in 7-eleven with Microsoft Excel combined with VBA code. Finally this software is used in tutorial lectures where the students have the opportunity to implement the same models studied before. After that, more complex models with applicable software will be applied to enhance the students experience for simulation of the real service systems.

Conclusions

The queuing theory is an aspect of Quantitative Analysis for Business. This topic is usually taught with real systems. When teaching at Panyapiwat Institute of Management, Microsoft Excel was combined with VBA code to simulate the real service systems such as supermarkets or convenience stores. The 7-eleven at Khwanpatthana branch was selected for a case study. This case study is used in tutorial lectures where the students have the opportunity to implement the same models. Eventually, the students receive a procedure to construct the program used to find the optimal server in more complex systems.

References

Albright, S.C. 2001. **VBA for Modelers.** Pacific Grove, CA: Duxbury/Thomson Learning.

Banks, Jerry, et al. 2010. **Discrete-Event System Simulation.** 5th ed. Upper Saddle River, NJ: Pearson Prentice Hall.

Centeno, M.A. 1996. "An Introduction to Simulation Modeling." In J.M. Charnes, et al. (eds.), **Proceedings of the 1996 Winter Simulation Conference,** pp. 15-22. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Chase, R.B., Anquilano, N.J., and Jacobs, F.R. 1998. **Production and Operations Management: Manufacturing and Services.** 8th ed. Boston: Irwin/McGraw-Hill.

Grossman, T.A. 2002. "Teachers' Forum: Spreadsheet Modeling and Simulation Improves Understanding of Queues." **Interfaces** 29, 3: 88-103.

Hall, R. 1990. **Queuing Methods for Service and Manufacturing.** Englewood Cliffs, NJ: Prentice Hall.

Lawrence, John A., and Pasternack, Barry A. 2002. **Applied Management Science: Modeling, Spreadsheet Analysis, and Communication for Decision Making.** 2nd ed. New York: Wiley.

Mesquita, M.A., and Hernandez, A.E. 2006. "Discrete-Event Simulation of Queues with Spreadsheets: A Teaching Case." In L.F. Perrone, et al. (eds.), **Proceedings of the 2006 Winter Simulation Conference,** pp. 2277-2283. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Render, Barry, Star, Ralph M., and Hanna, Michael E. 2009. **Quantitative Analysis for Management.** 10th ed. Upper Saddle River, NJ: Pearson Education International.

Sadiku, Matthew N.O., and Tofighi, Mohammad R. 1999. "A Tutorial on Simulation of Queuing Models." **Electrical Engineering Education** 36, 36: 102-120.

Schriber, T.J., and Brunner, D.T. 1998. "How Discrete-Event Simulation Software Works." In J. Banks (ed.), **Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice,** pp. 356-357. New York: Wiley.

Taha, Hamdy A. 2007. **Operations Research: An Introduction.** 8th ed. Upper Saddle River, NJ: Pearson/Prentice Hall.

Nithipat Kamolsuk received his Master of Science Degree in 2004 from the Department of Science, Kasetsart University, Bangkok, Thailand, and Bachelor of Science Degree in 2000 from the Department of Applied Science, King Mongkut's University of Technology North Bangkok, Thailand. He is currently a lecturer and Chair of the Department of General Science, Panyapiwat Institute of Management, Thailand. His research interests cover applied mathematics and statistics.