Predictive Business Forecasting using Queue Simulation

AHMED TAREK
Eastern Kentucky University
Department of Computer Science
407,Wallace Building, 521 Lancaster Avenue, Richmond, Kentucky
UNITED STATES OF AMERICA

Abstract: Queues are not merely an object to study. Queues are real world problems faced by customers at a business station that kill their valuable time. This results in decreased customer interest and the progress trends towards a loss. This is particularly frightful with smaller investments, where a single loss may permanently eliminate the business. It is not enough to know how queue behaves, but we are also required to know how to eliminate queues or at least how to reduce them to a manageable size. It is possible to reduce the queuing delay by changing the service mechanism. This involves increasing the rate at which customers are being served. Queues grow when customers happen to arrive at a faster rate than they are being served. This type of service dependent queues may be eliminated or at least reduced by developing a demand responsive service strategy that reduces the service time variation. Practical business transactions are too complex to be studied analytically. Besides, analysis using the collective practical data is hugely expensive and prohibitively time consuming that may affect the investment incentives. In this paper, a queuing model to simulate a small commercial establishment has been proposed, and a C++ program, which is founded on the model, is implemented. Simulation data plots are presented and the data tables are analyzed to make predictive forecasting over the significant transaction factors. Mathematical analysis fundamental to the proposed model is also incorporated, so that the model may be extended to a multitude of other similar applications in future.

Key–Words: M/G/1 queue, Simulation and model analysis, Simulation program, Performance analysis, Queuing theory, Queue prediction.

1 Introduction

Queuing Theory is often used to analyze the performance of practical queues based upon prediction. Queuing applications are common to data-processing tasks for customer transactions, jobs, order processing, etc. Queue data structures are used in *simulation applications*, programs that model real-world events, and track their behavior over time. These models are often prone to high validity since they can track the true system behavior. Commercial real-world transactions are too complex to be studied analytically. Besides, analysis using practical data is hugely expensive and prohibitively time consuming that may affect the organization's profit goals. These realistic models are studied by simulation for economy and dynamism. In this paper, a single-server $M/G/1$ queuing model has been used for predictive analysis of a small commercial establishment. We are particularly interested in service time and service rate predictions as well as in estimating the queuing delays. A complete set of Visual C++.NET programs are developed and implemented for this simulation.

In section 2, terminology and notations used with the simulation model are discussed. Section 3 introduces objective behind the simulation and describes the model used for the simulation. Algorithms used for the simulation are also explored. Mathematical formulations depicted in section 4 is fundamental to the proposed model. Section 5 analyzes simulation data for predictive waiting and service time forecasting. Section 6 is predictive curve analysis in estimating significant transaction factors.

2 Terminology And Notation

Following notations are adopted all throughout this paper. Some of these notations appear in [3].
Arrival Rate: The rate at which customers arrive at the queue. This is denoted by λ.
Service Time: The average time required to complete a customer transaction. The time required to serve the jth customer is denoted by s_j. Average time needed to serve a single customer is denoted by W_s.
Mean Turnaround Time: Average time that a customer spends in the system. This is denoted by W_r.
t_j: Time of arrival of the jth customer, which is an integer.

a_j = t_j - t_{j-1} = Inter-arrival time between the $(j-1)$st and the jth arrival of customers.
d_j: Delay of the jth customer inside the waiting queue.
W_q: Average customer waiting time.
L_q: Average length of the waiting queue.
c_j = t_j + d_j + s_j: Time at which jth customer is served completely and leaves the store.
w_j = d_j + s_j: Time that the jth customer spends in the queuing system.
L_r: Average number of customers in the system.
$q(t_j)$: Number of waiting customers in the queue at time t_j.
q_{max}: Maximum number of customers inside the waiting queue over a period of time.
$l(t_j)$ = $q(t_j)$ + 1: Number of customers in the queueing system at time t_j.
n: Total number of customers served over a period of time.
T: Total time in minutes in a working day, during which, the customers are being served.
$B(t_j)$: *Busy function*. It is 1 if the server is busy at time t_j, and 0 if the server is idle.
ρ: Utilization factor of the server in serving customers.
μ: Service rate of the server.

3 Simulation Model and Algorithms

In this paper, queue simulation has been used for predicting significant business operation factors. The major focus is average service time, average customer wait time and the average service rate rendered by the clerk.

3.1 Problem Description

Following depicts the $M/G/1$ model studied for the simulation.
1. The model store operates 7 days a week. It opens at $8:00$ a.m. in the morning and closes at $10:00$ p.m. in the evening. The store effectively operates for 13 hours in a typical business day. Customer arrival times satisfy Poisson distribution [2]. It has been assumed that on average, a customer arrives at every ten minutes. An algorithm is used to generate a random number that lies in between 0 and 1. If the number is greater than $e^{-0.10}$ (as $\frac{1}{10}$ = 0.10), it has been assumed that a new customer has arrived at that minute.
2. In each business day, the store operates in three different cycles. The morning shift operates from $8:00$ a.m. until noon. Then there is an hour long lunch break. The second shift starts at $1:00$ and runs until $5:00$ in the afternoon. At $5:00$, there is a change of service clerk. The operation briefly pauses for about 15 to 20 minutes, and all customers inside the waiting queue are dispersed. Once accounting with the current clerk is over, a new service clerk takes over. The business initiates its normal operation with an empty queue and runs for another 5 hours (until around $10:20$ p.m. at night). For convenience, the last shift is assumed to start at 5, which runs until 10 at night.
3. Customer transactions start and stop in 1 minute interval with a precision of 1 minute that uses a Simulation Clock. Customers are assumed to arrive on the minute, and if necessary, wait an integral number of minutes inside the waiting queue, and require an integral number of minutes for being served. The service times s_j, $j = 1, 2, \ldots$ of the successive customers are independent of the inter-arrival times. If an incoming customer finds that the clerk is busy, he joins the waiting queue. Once the transaction with the current customer is over, the server picks up a customer from the waiting queue (if there is one) on a first-come first-served (FIFO) basis.
4. For each minute of operation, the program checks the following events:
a. Whether a new customer has arrived or not.
b. Whether a new customer transaction has started or not.
c. Whether an on-going customer transaction has completed or not.
5. For preciseness, longer simulation times are considered as follows.
a. Data is collected and the average parameter values are computed by running the code for 7 and 30 consecutive business days.
b. In applying regeneration to the simulation model for precise, accurate and realistic results, each day's operation has been subdivided into three different cycles, and the simulation is conducted over seven consecutive business days with three cycles on each day. Finally, the average parameter values over these seven business days are computed.
c. The daily average waiting time is denoted by t_{avg_i}, where $i = 1, 2, 3, \ldots, 7$ for seven, and $i = 1, 2, 3, \ldots, 30$ for thirty consecutive business days. We compute the average waiting time using the following equation:

$$t_{avg} = \frac{(\sum_{i=1}^{k} t_{avg_i})}{k}, \; k = 7 \; or \; 30 \qquad (1)$$

Transactions are made in two different occasions.
(1) The queue is empty and the server is free. In this event, customer is served immediately. This customer's waiting time is zero.
(2) A customer has just arrived, and either the queue is nonempty or the server is busy. This customer waits in the queue for the server to become free. This type of customers have nonzero waiting time.

3.2 Algorithms

Four different algorithms are used for the complete simulation program.

Algorithm runSimulation
Purpose: This algorithm simulates the business.
Begin
createQueue (BuyerQueue)
clock = 1
endTime = 780
customerNumber = 0
moreCustomer = false
while clock $\leq$ endTime or moreCustomer **do**
  newCustomer (BuyerQueue, clock, customerNumber)
  clerkFree (BuyerQueue, clock, customerStatus, moreCustomer)
  serviceComplete (BuyerQueue, clock, customerStatus, simulationStatistics, moreCustomer)
  if (not emptyQueue(queue)) **then**
    moreCustomer = true
  end if
  ++clock
end while
ReportStats (simulationStatistics)
return
End

Algorithm newCustomer
Purpose: This algorithm determines whether a new customer has arrived or not.
Begin
arrivaltime = (rand() /((double)(RAND_MAX + 1))) // RAND_MAX = 2147483647
if (arrivaltime $> exp(-0.1)$) **then**
  $++$customerNumber
  customerData.number = customerNumber
  customerData.arrivalTime = clock
  enqueue (BuyerQueue, customerData)
end if
return
End

Algorithm clerkFree
Purpose: This algorithm determines whether the server is idle and if so, starts serving the next customer inside the waiting queue.
Begin
if (clock$>$status.startTime+status.serviceTime-1) **then**
  if (not emptyQueue(queue)) **then**
    dequeue (BuyerQueue, customerData)
    status.customerNumber=customerData.number
    status.arrivalTime=customerData.arrivalTime
    status.startTime=clock
    status.serviceTime=random service time
    moreCustomer=true
  end if
end if
return
End

Algorithm serviceComplete
Purpose: This algorithm determines whether the current customer's processing has completed or not.
Begin
if (clock==status.startTime+status.svcTime-1) **then**
  waitTime= status.startTime-status.arrivalTime
  ++status.numCust
  stats.totsvcTime=stats.totsvcTime+status.svcTime
  status.totWaitTime=status.totWaitTime+WaitTime
  queueSize=queueCount(BuyerQueue)
  if (statistics.maxQueueSize$<$queueSize) **then**
    statistics.maxQueueSize = queueSize
  end if
  Report (status.customerNumber,status.arrivalTime, status.startTime,status.serviceTime,waitTime, queueCount)
  moreCustomer = false
end if
return
End

The first algorithm may be modified to simulate 7 and 30 business days. The algorithms may be tailored to apply the regenerative technique with three different cycles in each business day.

4 Mathematical Foundation

Following equations hold true for the proposed $M/G/1$ model. Next equation computes the *average delay* W_q.

$$W_q = \frac{\sum_{j=1}^{n} d_j}{n} \tag{2}$$

Average time that a customer spends in the system is given by:

$$W_r = \frac{\sum_{j=1}^{n} w_j}{n} \tag{3}$$

Average number of customers waiting and the average number of customers in the system are, respectively:

$$L_q = \lambda W_q \tag{4}$$
$$L_r = \lambda W_r \tag{5}$$

Here, λ =customer arrival rate. The equations involving L_q and L_r are known as the Little's Formula. Service clerk's efficiency is computed as:

$$\rho = \frac{Total\ service\ time}{Total\ business\ time} = \frac{T_{service}}{T} = \frac{\sum_{j=1}^{n} s_j}{T} \tag{6}$$

Average service time W_s is:

$$W_s = \frac{\sum_{j=1}^{n} s_j}{n} \tag{7}$$

Parameter W_s may be fractional as well. Number of customers served per unit time is given by:

$$\mu = \frac{1}{W_s} = \frac{n}{\sum_{j=1}^{n} s_j} \tag{8}$$

With the single-server queue, for *system stability*:

$$Arrival\ rate\ \lambda < Service\ Rate\ \mu \tag{9}$$

Therefore:

$$\lambda < \frac{1}{W_s} \Rightarrow \lambda < \frac{n}{\sum_{j=1}^{n} s_j} \tag{10}$$

The server efficiency ρ is given by:

$$\rho = \frac{\lambda}{\mu} = \frac{\lambda \times \sum_{j=1}^{n} s_j}{n} = \frac{\sum_{j=0}^{T} B(j))}{T} \tag{11}$$

For a *stable single-server queue model*:

$$\lambda < \mu \Rightarrow \frac{\lambda}{\mu} < 1 \tag{12}$$

The factor ρ is less than 1, and the following relationship is being satisfied:

$$\lambda \times \sum_{j=1}^{n} s_j < n \tag{13}$$

However, if:

$$\lambda > \mu = \frac{n}{\sum_{j=1}^{n} s_j}, \tag{14}$$

then the server goes further and further behind. Eventually, after certain time, the server will always be busy and the waiting line will increase at an average rate of ($\lambda - \mu$) customers per unit time.

5 Result Analysis

No. of Custs	Total Svc. Time	W_s	W_r	q_{max}
66	551	8	6	4

Table 1: Results over one business day (time values are in minutes)

Day	Tot cust	Svc. time	W_s	W_q	q_{max}
1	76	626	8	12	5
2	67	554	8	14	5
3	67	554	8	14	5
4	60	462	7	6	4
5	60	462	7	6	4
6	62	546	8	9	5
7	62	546	8	9	5
Avg.	65	536	7.8	10	5

Table 2: Simulation results for seven consecutive business days (time values are in minutes).

Initially, the model is simulated for a complete working day without regeneration (see Table 1). From Table 1, average service time is 8 minutes. For better accuracy, simulation was conducted over 7 consecutive business days. Table 2 shows the results.

From the accumulated statistics in Table 2, the average service time is 7.8 minutes. This deviates only by 2.5% from that in Table 1.

From the regenerative data in Table 3, average service time is 8.4 minutes from 8 to 12 p.m. This deviates by 5% from that in Table 1. In between 1 to 5, this average is 7.7 minutes, differing only by 3.75%. From 5 to 10 p.m., this factor remains identical to that in the previous cycle.

From Table 4, there is only 7.7% chance of average service time being 6 minutes in each hour. This is only 2.2% for the weekly average. The highest 84.6% probability is for the hourly service time of 7 minutes. There is only 7.7% chance of service time being 8 minutes. Average service time follows a bell-curve pattern within this 6 to 8 minutes range. Beyond this range, the probability is 0.

Table 5 is the frequency of waiting-time distribution estimated over a week. Out of 494 total customers served during this time, 140 didn't have to wait at all. This number is 28.34% of the total. However, 202 customers waited for more than 12 minutes. This is 40.89% of the total number served. Remaining $(100-28.24-40.89)$% = 30.87% customers have waited in between 1 to 12 minutes.

6 Curve Prediction

For accuracy in predictive forecasting, data graphs are plotted using *gnuplot version 4.0* software.

Figure 1 shows simultaneous plotting of the average queue size and the maximum waiting queue size versus days curves for 30 business days. Queue population was maximum for both the plots in between $20th$ to $25th$ business days with parallel bell-curves

Day	Slot	Tot cust	Svc. time	W_s	W_q	q_{max}
1	8-12	17	150	8	3	2
1	1-5	16	163	10	5	2
1	5-10	28	249	8	56	16
2	8-12	20	181	9	4	4
2	1-5	19	184	9	13	5
2	5-10	28	237	8	17	12
3	8-12	27	236	8	16	6
3	1-5	30	203	6	18	7
3	5-10	34	296	8	17	16
4	8-12	26	171	6	22	9
4	1-5	21	155	7	25	8
4	5-10	28	203	7	14	17
5	8-12	26	237	9	13	6
5	1-5	22	191	8	5	3
5	5-10	28	249	8	11	15
6	8-12	23	172	7	5	4
6	1-5	24	182	7	10	5
6	5-10	30	237	7	17	14
7	8-12	22	161	7	12	9
7	1-5	29	211	7	8	7
7	5-10	35	289	8	18	15
Avg.	8-12	23	187	8.4	10.7	6
Avg.	1-5	23	184	7.7	12	5
Avg.	5-10	30	251	7.7	21.6	15

Table 3: Simulation with regeneration over 7 days.

Service time	Hourly prob./day	Hourly prob./week
6	0.077	0.022
7	0.846	0.396
8	0.077	0.461
9	0	0.121

Table 4: Hourly service time distributions.

Waiting time	No. of customers waited	Probability
0	140	0.2834
1	15	0.0304
2	15	0.0304
3	18	0.0364
4	4	0.0081
5	13	0.0263
6	13	0.0263
7	16	0.0324
8	13	0.0263
9	13	0.0263
10	8	0.0162
11	9	0.0182
12	15	0.0304
>12	202	0.4089

Table 5: Individual waiting time distributions.

within this range. Therefore, hiring an additional service clerk may meet the increased service demand over this period.

With $M/G/1$ queues, longer waiting lines may be reduced by decreasing the server utilization, ρ. Figure 2 is the dual plotting of the average queue length and the expected queue length versus server efficiency curves. The curve with a lower slope, which didn't intersect the x-axis is for the average queue length. The curve with a higher slope intersecting the x-axis at $\rho = 0.68$ is the one for the expected queue length. The two curves intersect at $\rho = 0.81$, which represents the optimal server efficiency. At $\rho = 0.81$, the expected and the average queue lengths become equal, which is 2.4. With a realistic assumption of 8 minutes to serve a customer and an optimal server efficiency of 0.81, the number of total customers served during each business hour is, $\lfloor \frac{60\times 0.81}{8})\rfloor = 6$. In a typical business day with 13 hours of operation, the expected number of customers to be served is, $6 \times 13 = 78$.

From Figure 3, the number of minimum customers served in an hour is 3, which is 50% below the expected number. The number of maximum customers served is 7, which is $\frac{(7-6)\times 100\%}{6} = 16.67\%$ above the expected number. Table 6 lists the hourly customer counts over four business days.

Average number of customers served over 4 business days is:

$$L = \lfloor (\frac{\sum_{i=1}^{4}\sum_{j=1}^{13} n_{ij}}{4\times 13})\rfloor \tag{15}$$

Here, n_{ij} = number of customers served at the jth hour on the ith business day. Using equation (15), $L = \lfloor \frac{285}{52} \rfloor = \lfloor 5.481 \rfloor = 5$. This is $\frac{(6-5)\times 100\%}{6} = 16.67\%$ less than the expected number.

7 Conclusion

In this paper, a predictive approach in estimating business transaction factors is considered. The model simulated with computer programs steps through the behavior of the system and reports experience. The behavior is tracked in *computer variables* and *program*

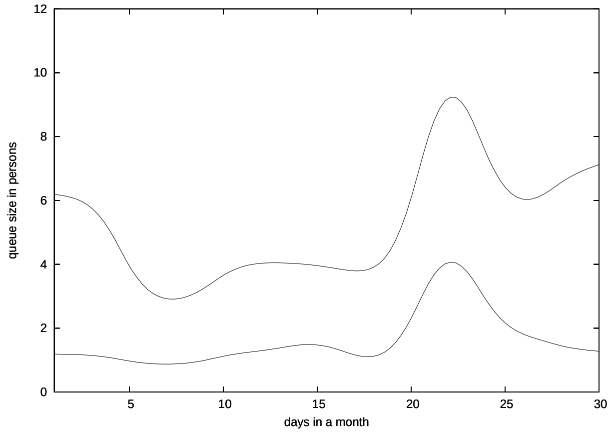

Figure 1: Average queue size (lower curve) and the maximum queue size (upper curve) in person plots.

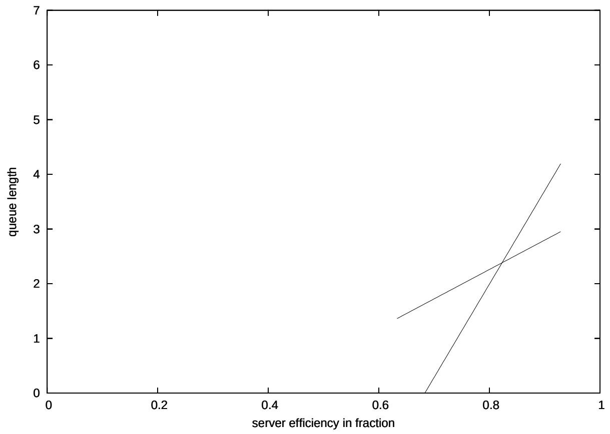

Figure 2: The expected queue length and the average queue length versus the server efficiency curves.

logic rather than as a physical system. For the prototype $M/G/1$ commercial system, the computer model presented can retain descriptive accuracy. This simulation only provides statistical estimates rather than exact results.

There are two good reasons for simulating the business model.

1) It is too expensive to conduct such a real-world study over a considerable period of time in a small service organization.

2) Computer simulation is efficient, and generates satisfactory results within reasonable amount of time. Small service stations are dynamic and fast growing. Therefore, conducting a real-world simulation would be too time consuming for such a small establishment.

As the time intervals are only 1 minute, therefore, the probability of two or more arrivals or two or more service completions during an interval is negligible,

Hour No.	Day 1	Day 2	Day 3	Day 4
1	5	4	6	6
2	7	5	7	5
3	5	5	8	5
4	5	4	5	4
5	11	2	6	7
6	6	5	7	2
7	4	9	4	6
8	7	7	4	5
9	8	6	4	5
10	5	4	5	7
11	4	6	2	6
12	3	6	5	8
13	3	8	3	7

Table 6: Customer counts per hour over 4 days.

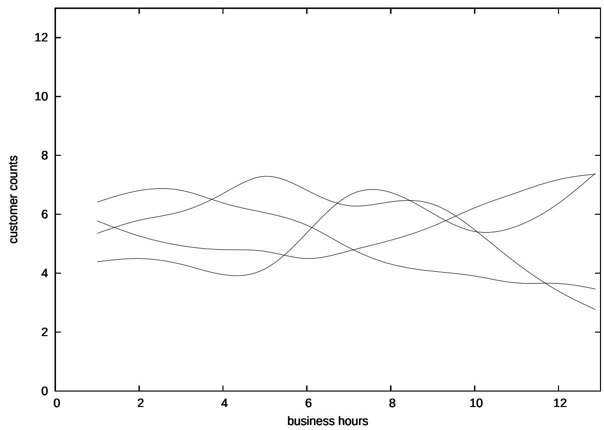

Figure 3: Number of hourly customers vs. corresponding business hour plot over 4 business days.

and we have discarded such possibilities. Future research plan incorporates error estimation in the predictive forecasting and techniques to reduce such errors to a minimum possible level.

References:

[1] Richard F. Gilberg and Behrouz A. Forouzan, *Data Structures – A Pseudocode Approach with C++*, BROOKS/COLE, Thomson Learning, California 2001

[2] Frederick S. Hillier and Gerald J. Lieberman, *Introduction to Operations Research, Seventh Edition,* McGraw-Hill, New York 2001

[3] Averill M. Law and W. David Kelton, *Simulation Modeling and Analysis, Third Edition,*, McGraw-Hill, New York 1986