A Simulation of Massy Stores Queuing System

Jonathan Earle, Kerschel James, Gerard Rique and Rondell Gulston
814000869, 814004296, 814002944 and 814001673

***Abstract*— Lines serve as some of the greatest annoyances of everyday life. Chief among these irksome lines are those that we encounter in grocery stores. Consequently, deciding how the lines in grocery stores are managed, created, and run is vital towards providing a good experience for customers.[1] This report details the method used to implement simulations of the Massy Stores, St.Augustine, queues. The simulation revealed ...**

I. INTRODUCTION

Queuing theory equations were used to generate simulations of queues at Massy Stores and a Monte Carlo simulation was done to evaluate the value of each queue. The simulations were based on data collected at the store. This report outlines the data collection process, calculations which were performed in the simulation, results from running the simulation using various parameters and finally recommendations to improve the queues at the store for both customers and the store itself.

II. DATA COLLECTION

The data collection method along with the data itself are detailed below.

A. Procedure

The queues were observed by two person at a time. They recorded each queue for a duration of 30 min. This was done for six (6) queue in total including one express lane from. The data was collected between 3PM and 4PM.
The data each person collected included the arrival time of each customer, their service start time, service end time, the number of items they had at checkout and if they used card or cash to pay for the items.
Data collection begins with the first customer observed by the surveyor to join the queue , regardless of if the queue was initially empty or not. If an individual drops out of the line their dropout time is also recorded.

B. Data

The data collected for each queue is stored in six CSV files. A sample of the data collected can be seen in Table I.

TABLE I
SAMPLE DATA

#	Arrival	Service Start	Service End	Drop Time	Items	Card
1	3:00:09	3:06:30	3:08:30	0	12	1
2	3:01:15	0	0	3:12:40	0	0

If a customer paid for their groceries with a card a one was recorded, zero otherwise. The number of items each customer has was counted at checkout. If a customer drops out of the line the time at which they do so is recorded.
The wait time in the queue and the time it takes for each customer to be served is calculated in the CSV file using spreadsheet software, this was done by subtracting the service start and arrival time and by subtracting the service end and service start times respectively.

C. Observations

It was observed that customers usually let other customers skip in front of them when a person has much less items to cash than them, this is because the person with more items empathizes with the other customer. Another important observation was that customers sometimes observed which queue was progressing quicker before choosing one. However this method did not always result in speedier service as the amount of items the customers in front of them had was the impacting factor for service times, this method also overcrowded the faster lanes thus increasing their average wait time. It was also theorized that observer-expectancy effect caused the cashiers to work faster than usual.

III. CALCULATIONS

A detailed description of the calculations performed to run the simulation are stated below.

A. Assumptions

The simulation assumes the following:

- The mean service rate is exponentially distributed.
- The inter-arrival time is exponentially distributed.
- Cashier behavior was not affected by observation.
- There was no unusual customer behavior at the time.
- The maximum number of items a customer can purchase is 1000.
- The probability of dropout is solely based on the number of items in front of a customer and the amount of time they have waited.
- The average service rate of customers is dependent on the amount of items they have and if they use a card or not.
- The arrival rate is dependent on the total number of items in the queue.
- The dropout rate is independent of the length and existence of other lines.

B. Equations

The arrival rate per minute is found by using the equation

$$\lambda = \frac{1}{\frac{\bar{x}}{60}} \tag{1.0}$$

Where $\bar{x}$ is the mean inter-arrival time based on the data collected. This rate was calculated based on the number of items in the queue at a given time. The reasoning behind this is that customers choose lines based on how much items a line appears to have.

The service rate was calculated as follows

$$\mu = \frac{1}{\frac{\bar{y}}{60}} \tag{2.0}$$

Where $\bar{y}$ is the mean service time based on the data collected. Different service times were calculated based on the number of items a customer has and the average difference in time between card and cash users was added on to the mean service time to create the final mean service time. Equation (2.1 outlines this.

$$\bar{y} = s(x) + c(y) \tag{2.1}$$

Where $s(x)$ is the average service time for a customer with x items and $c(y)$ is the mean extra service time for a customer that uses a card or not, where y is a boolean, 1 representing uses a card and 0 otherwise.This implies that the number of items to be checked out and the time difference between doing card and cash transactions affects the service rate.
A Bernoulli trial was used to determine if a person used a card or not, the probability of using a card was calculated form the data collected. Roulette Wheel sampling was done to determine the number of items a user entering the line has, again the probability of having a certain number of items was calculated from the collected data.

Both the service and arrival rate are used to calculated random exponentially distributed service and inter-arrival times respectively. This is done by plugging the rate into the equation (3).

$$-\frac{\log(1 - unif.rand(0,1))}{\Delta} \tag{3.0}$$

Where Δ is either the service rate or arrival rate. To avoid passing 0 to $\log$ we subtract 1 from the uniformly generated random number which is always less than 1.

The cost of service in a queue is found by finding the product between the mean service time of the queue in minutes and the cost of a cashier per minute. This can be seen in equation four (4).

$$Cs = \bar{s}0.25 \tag{4.0}$$

Where $\bar{s}$ is the mean service time of a queue. The cost of a cashier per minute is found by taking their pay per hour and dividing it into 60. $15 per hour is the average pay of a cashier, giving a pay of $0.25 per minute.
Similarly the cost of waiting in the queue is expressed as the product between the probability of a person dropping out of the queue and the mean wait time of the queue. This can be seen in equation five (5).

$$Cw = P_d(wait, queue)\bar{w} \tag{5.0}$$

Where $\bar{w}$ is the mean wait time and $P_d(wait, queue)$ is the probability that a customer drops out of the queue. Information on the customer used to calculate their probability of dropout includes their current wait time and the total amount of items in front of them in the queue.

C. Implementation

```
Input  : Duration of simulation in minutes
Output: W- wait times, S- service times
Queue ⟵ [ ];
W ⟵ [ ];
S ⟵ [ ];
Arrival ⟵ 0 ;              /* Next arrival */
Service ⟵ 0 ;    /* Service Completion */
while True do
    Event ← min(Arrival, Service) ; /* Returns
     param 1 if equal */
    foreach customer ∈ Queue do
        if dropout(customer) then
            Queue ← pop(Queue, customer);
            break;
        end
    end
    Elapsed ← Elapsed + Event;
    if Elapsed >Duration then
        break;
    end
    if Event==Arrival then
        cust ← GenerateCustomer(Elapsed);
        Queue ← push(Queue, cust);
        arrRate ← GenerateArrivalRate(Queue);
        Arrival ← Arrival + rand.exp(arrRate);
    else if Event == Service then
        cust ← pop(Queue);
        W ← push(W, Elapsed − cust.arrivalTime);
        servRate ← GenerateServiceRate(cust);
        service ← rand.exp(servRate);
        S ← push(S, service);
        if Queue.length >0 then
            Service ← Service + service;
        else
            Service ← Arrival + service;
end
```

Algorithm 1: Single Queue Simulation

Calculations were done for each queue data was collected for. The simulation of each queue was run through a Monte Carlo simulation for 1000 runs, which found the expected wait time, service time, number of sales and the cost of waiting and service for the queue. Algorithm 1 contains the pseudo-code used to simulate a single queue.

IV. RESULTS

A. General Results

Table II shows the results obtained from the simulation on each queue. The values are the expected values in minutes based on the Monte Carlo simulation. Each queue was simulated for 30 minutes.

TABLE II
SIMULATION RESULTS

Queue	Wait Time	Service Time	Length	Sales	Service Cost	Waiting Cost
1	1.35	1.41	1.31	101.50	0.35	0.00
2	2.40	4.96	1.17	55.67	1.24	0.00
3	1.14	2.25	1.07	52.09	0.56	0.00
4	0.48	0.75	1.05	66.31	0.19	0.00
5	4.91	2.90	3.48	135.35	0.72	0.00
Express	0.47	0.63	1.07	92.18	0.16	0.00

It can be seen that the cost of waiting for each queue is essentially nonexistent, this is because the cost of waiting is associated with the probability of dropout from the queue. This probability is very small and has no effect. This is rather intuitive because customers usually join a line and once progress is made in that line they rarely take the risk of joining a new line in the hopes of being served faster.
Furthermore it can be seen that Queue 4 had a low wait time, this was seen as an anomalous case as the cashier seemed to be moving faster than usual due to the fact they were being observed.

B. Accuracy of Simulation

Difference between the simulation and the actual queues can be explained by numerous factors, such as human error during observations, the observer-expectancy effect or unusual customer behavior during data collection. However the most probable cause of the discrepancies would have to be unassociated links created when calculating the dropout probability and the arrival and service rates. That is arrival rates were not linked to the number of items in the queue, service rates were not linked to the number of items to be cashed and dropout was not determined by the number of items to be cashed in front of a customer, the latter being the most probable.

V. RECCOMENDATIONS

Increasing the service rate of each cashier would help, however this isn't always a possibility as cashiers do not go through a thorough screening process and the cashier speed may vary from time to time. Having more cashiers wouldn't help either because it increases the cost of operation and customers overcrowd the faster cashiers making the extra cashiers redundant. A more robust solution would be to increase the number of express lanes during average traffic hours as the express lanes were generally quicker. From the simulation it can be seen that adding one more express lane would decreased both the cost of waiting and the cost of service. This is so because customers are less likely to dropout of the express lanes and cashiers work faster when the amount of items they cash is within a small consistent range.

VI. CONCLUSIONS

It was found that the queues at Massy Store, St.Augustine had great variation in the service rates of their queues. This variation caused queues with speedy service to become overcrowded, thus increasing the average wait time of fast queues. If there was a more uniformed service time between queues this would not be a problem. It was also found that if the supermarket added more express lanes during average traffic times they would see a decrease in the cost of service and the cost of waiting, as lines with 10 or fewer items generally moved smoother.

ACKNOWLEDGEMENTS

Special thanks to Massy Stores St.Augustine for allowing us to collect data and to Inzamam Rahaman for his guidance whilst doing this project.

REFERENCES

[1] Inzamam Rahaman, *COMP3950 Project Description,* March 16th 2017.