Simulation Study of a Gas Station

Final Project, BANA 7030 Simulation Modeling and Methods

By Rasesh Garg, M13437578

"On my honor, I have neither given nor received unauthorized aid in completing this academic work." - Rasesh Garg

Executive Summary

Our client wants to open a Gas station and is considering launching it with 4 fuel tanks. This study evaluate client's proposed set up by simulating it using the queuing model and calculating parameters like percentage customers served, expected waiting time and resource utilization. Further study analyzes alternate models and finds the expected increase in business revenue and decrease in waiting time of customers with each additional fuel tank.

Table of Contents

Problem Statement

My client Sahil, owns a gas station and is looking to expand his business by opening one more gas station at a busy highway in India. Currently he is planning to open it with 4 fuel tanks, 2 in each lane. Sahil wants me to evaluate the setup during busy hours and provide insights and recommendations, if any.

Objective

- Simulate a gas fueling station and evaluate it based on key metrics like waiting time, percentage customers served, queue length and resource utilization.
- Simulate alternate models, compare and provide recommendations for improvement.

Approach

- Estimate customer arrival time, customer's patience level and the service time through client's input, and primary survey from target customers.
- Simulate the current set up, and observe key metrics like waiting time, percentage customers served and resource utilization.
- Simulate and compare different models (with higher or lower number of tanks) based on above metrics.

Estimating input parameters for the model

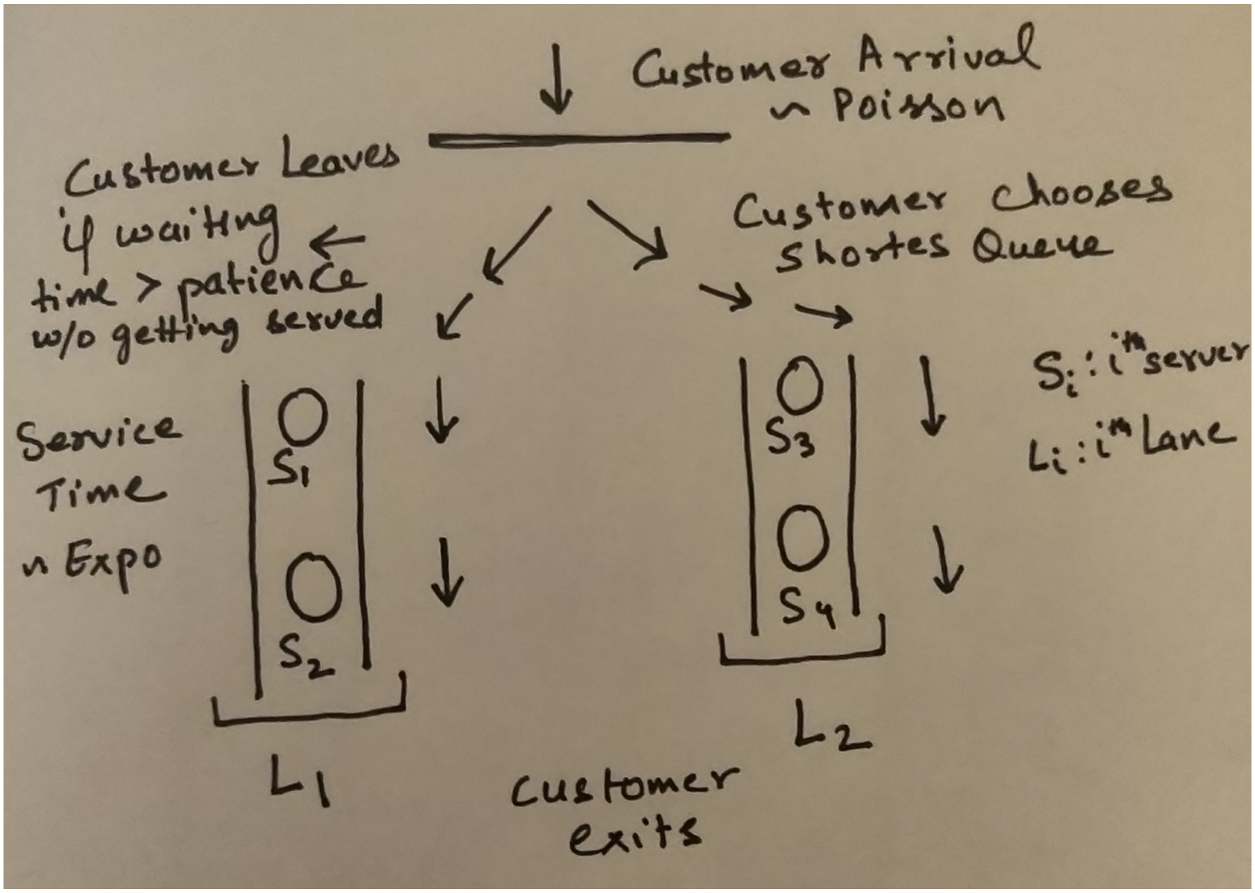
Secondary research on gas station shows that the service time typically follows an exponential distribution and the customer arrivals follows a Poisson distribution. Client has provided that based on his experience and given the high demand at the location, he is expecting on an average 3 cars every 2 minutes.

To estimate service time parameter and customer patience level, I designed a small survey. In total 20 responses were recorded from friends in India. Below are the questions that were asked:

1. Typically, how much time does it take you to refuel your car at a gas station?
2. What is the maximum amount of time you would wait at a gas station, before leaving?

Collating the responses from survey, we found the mean service time as 4 mins and patience level to be around 15 min.

Initial Proposed Setup



Summarizing Features of the above setup:

- We have two queues, each with two servers (i.e. 2 tanks)
- If the waiting time exceeds customer's patience level than the customer exits the system, without getting served.
- Patience level of the customer is considered fixed as 15 min (from the results of the survey)
- Queue length can be infinite.
- Customer arrival follows a Poisson distribution with mean of 3/2 cars per min.
- Service time follows an exponential distribution with mean service time of 4 min.

Simulating entire setup

I took this project as an opportunity to learn simulating queuing models in R. Skimmer package in R provides customizable and extremely fast functionality for discrete event simulation. Hence, I have used the same for this case study.

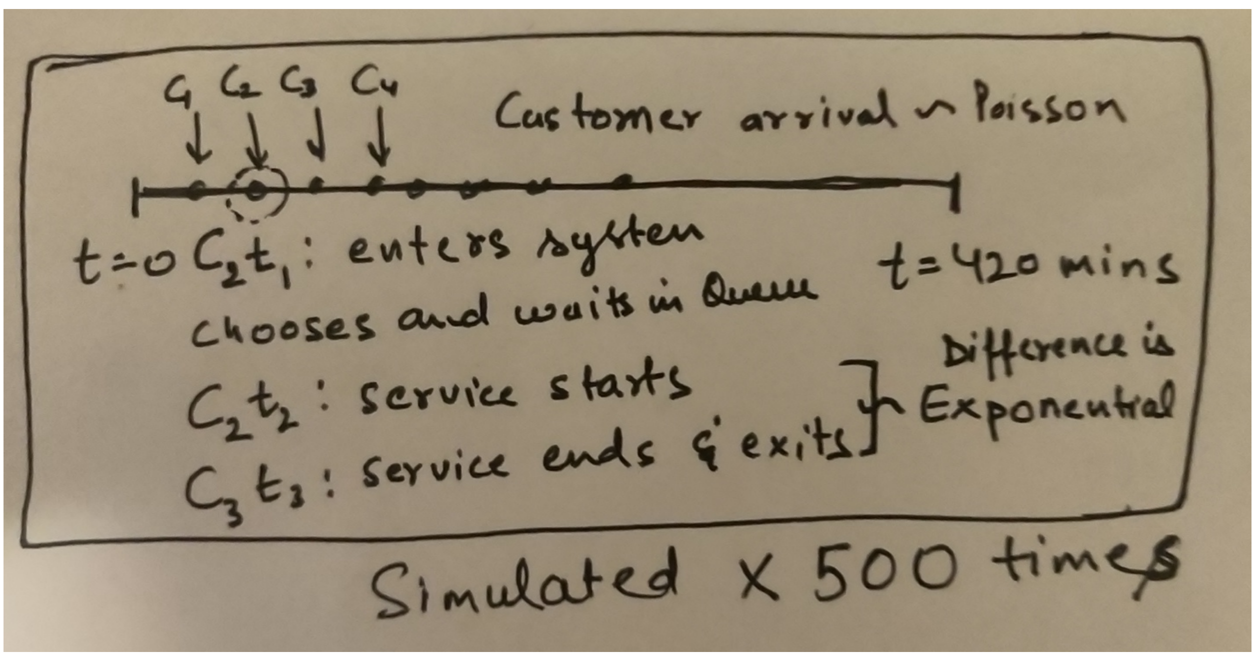
[Entire code with detailed comments is attached in the appendix. Results for a single simulation of all the models considered are also recorded in excel workbook for quick glance.]

How Simulation works

Customer arrival time and service time is considered to be a random variable belonging to a certain known distribution, for the duration of the experiment (i.e. 7 hrs. in our case). Thus, customers start arriving into the system at time t = 0 min, until t = 420 mins (7 hrs.) based on randomly chosen values from the know distribution that it follows (Poisson). Time taken to serve each customer is also chosen randomly from an exponential distribution. As the customers keep arriving continuously and the pace of service is slower than arrival, there is a queue that builds up. Each customer waits in this queue until he gets a chance to get served. If his waiting time exceeds his patience level than he leaves the system, before getting served. As in our setup there are two queues, the customer choses the one which is shortest. If both queues are of same length, the customer is assigned one randomly.

The above simulation is run 500 times, to minimize error and estimate the distribution of the parameters of interest.

I have tried depicting above steps in the graphic below also:



A thing missing in above graphic is that the customer leaves the system if his waiting time exceeds the patience level

Describing few key terms:

experiment_time = time window for which the experiment is run (420 minutes in our case)
start_time = time at which customer enters the system (C_2T_1)
end_time = time at which customer exits the system (C_2T_3)
service_time = time taken to serve the customer ($C_2T_3 - C_2T_2$)
waiting_time = time customer waits in the system ($C_2T_2 - C_2T_1$)

Current Model Evaluation

Let us first define our key decision metrics:

1. Mean Waiting time: This is the expected waiting time for a customer who gets served in the system (can be in any queue)
2. Time spent in system: This is the expected total time for a customer who gets served in the system.
3. Percentage customers served: Out of the total customers that arrived in the system, percentage of customers who had to wait for time less than their patience level and did not exit the system without getting served.
4. Resource utilization: percentage amount of time for which the tanks (servers) were occupied by the customers.
5. Mean queue length: Expected number of customers standing in a queue.

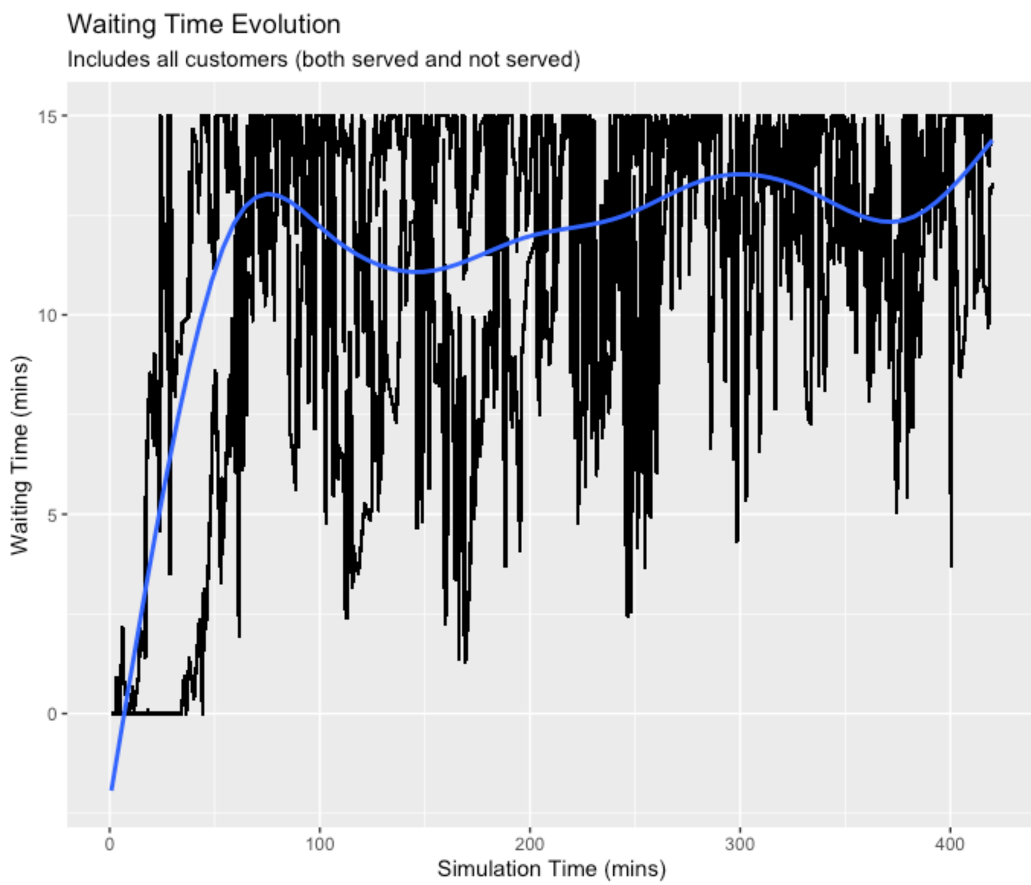
Each simulation will give us the mean waiting time, percentage customers served, resource utilization and mean queue length. **Since the simulation is done 500 times, we get distribution of these parameters** and therefore can calculate the 95% confidence interval (C.I). I assume each of this parameter is normally distributed and thus 95% C.I is calculated as mean(x) +/- 1.96*sd(x), where sd stands for standard deviation.

Below is the summary of key parameters

Parameter	Expected mean	95%_CI_lower_bound	95%_upper_bound
Waiting_time (mins)	10.4	9.8	11
Time_spent_in_system (mins)	14.5	14.2	14.8
Percentage_of_customers_served	69.1	68.8	69.5
Queue_length	8.6	7.7	9.5
resource_utilization (%)	98.9	97.5	100

- **Only around 70% of customers are getting served**, that means 30% of the customers are leaving the system because of waiting time exceeding their patience level (15 mins)
- Those who get served, have waiting time between 9.8 to 11 mins (95% C.I)
- Customers who get served, spend around 14 mins in the system
- There are two queues. Each has an average length between 7.7 and 9.5 (95% C.I)
- Resources (i.e tanks) are almost 100% utilized

Also, it would be interesting to note how does waiting time evolves with respect to the simulation time.



Each black line corresponds to a single simulation and blue is the average trends of 500 simulations. We can see that waiting time slowly increases to its peak as the queue starts building up. There are ups and downs in the waiting time post the first peak because some customers leave the system when their waiting time exceeds their patience level. This brings down the overall waiting time for some time.

Alternate Models

Let's simulate alternate models and compare how our parameters of interest will change. Certainly if we add more fuel tank we will be able to serve more customers and if we reduce fuel tanks, the served customers will decrease. Let's simulate and check the amount of this change in key parameters.

Model A: Decreasing one fuel tank
Number of lanes: 2
Tanks per lane: 2 in first and 1 in second
Total Fuel tanks: 3

Parameter	Expected mean	95%_CI_lower_bound	95%_upper_bound
Waiting_time (mins)	11.6	10.9	12.3
Time_spent_in_system (mins)	15.6	15.3	15.9
Percentage_of_customers_served	53.1	52.2	53.9
Queue_length	9.6	8.7	10.5
resource_utilization (%)	99.6	99	100

- **We will lose around 17%** more customers (as percentage of customers is now only around 53%) and the waiting time for those being served will increase by around 1 min.

Model B: Increasing one fuel tank
Number of lanes: 2,
Tanks per lane: 3 in first and 2 in second
Total Fuel tanks: 5

Parameter	Expected mean	95%_CI_lower_bound	95%_upper_bound
Waiting_time (mins)	7.6	7.6	7.6
Time_spent_in_system (mins)	11.6	11.4	11.8
Percentage_of_customers_served	85.8	85.2	86.3
Queue_length	6.3	5.9	6.7
resource_utilization (%)	96.8	94.9	98.7

- We are now able to serve around 86% of customers compared to only around 70% in the proposed set up. And the waiting time is also reduced by around 3 mins.

- We see that average queue length is much lower, this is because there are 3 queues instead of two. So this drop is not significant to us.

Model C: Increasing one fuel tank, with a different placement
Number of lanes: 3,
Tanks per lane: 2 in first two and 1 in third
Total Fuel tanks: 5

This model has same number of tanks as the previous one, but the placement is different. Here there are 3 lanes instead of 2 (in the previous model)

Parameter	Expected mean	95%_CI_lower_bound	95%_upper_bound
Waiting_time (mins)	6.8	6.6	6.9
Time_spent_in_system (mins)	10.8	10.7	10.8
Percentage_of_customers_served	84.9	83.7	86.2
Queue_length	3.8	3.5	4.1
resource_utilization (%)	95.9	95.1	96.8

- Placement difference does not change the key metric when compared to the previous model significantly. Thus, we will focus on number of fuel tanks in total and operational feasibility can be given more priority while deciding the placement.

Model D: Adding two additional fuel tanks
Number of lanes: 3
Tanks per lane: 2
Total Fuel tanks: 6

Parameter	Expected mean	95%_CI_lower_bound	95%_upper_bound
Waiting_time (mins)	5.2	4.9	5.4
Time_spent_in_system (mins)	9.1	8.6	9.6
Percentage_of_customers_served	96.1	94.8	97.4
Queue_length	2.6	2.6	2.6
resource_utilization (%)	91.9	88.7	95.2

- We are now able to serve around 96% of customers compared to only around 70% in the proposed set up. And the waiting time is also reduced by around 5.3 mins.
- Our resource is still being utilized up to 90%, that is only 10% of the time they are idle.

Conclusion and Recommendation:

With the client's proposed model, we are only able to serve 70% of the customers which means we are losing around 30% of the business. If the client increases one tank at the gas station, he will be able to serve 16% more customer (i.e. 86% in total) and with 2 additional tanks he can serve almost 96% of his customers. Thus, there can be a 26% business gain if our client adds two additional fuel tanks to his setup.

A cost benefit analysis can be done to decide how much time it will take to breakeven the initial investment of two tanks and calculate absolute profits thereafter. But certainly, in long term adding up two more tanks will help client earn profits, given the demand remains same or more as the time passes. Thus, my recommendation based on above analysis is that the client should add two additional fuel tanks.

Acknowledgement

- https://r-simmer.org/articles/simmer-04-bank-1.html#several-service-counters-1
- https://r-simmer.org/articles/simmer-04-bank-2.html#balking-and-reneging-customers-1
- https://r-simmer.org/extensions/plot/articles/plot.simmer.html
- https://r-simmer.org/articles/simmer-05-simpy.html#gas-station-refuelling-1

Appendix: (Code)

```
#install.packages('simmer')
#install.packages('simmer.plot')
#install.packages('parallel')
library(simmer.plot)
library(simmer)
library(dplyr)
library(tidyr)
library(parallel)

### Setting initial parameters of the model
max_wait_time = 15
```

```
simulation_time  = 60*7 # 7 hours, say friday 5-12 pm
mean_service_time = 4
customer_inter_arrival_time = 2/3 # i.e 3 cars coming every 2
minutes
no_of_simulations = 500
output_file_name = "Model_C.xlsx"

### Setting up the system (customer trajectory and environment)

customer <-
  trajectory("Customer's path") %>%
  log_("Here I am") %>%
  set_attribute("start_time", function() {now(bank)}) %>%

  renege_in(max_wait_time,
            out = trajectory("Reneging customer") %>%
              log_(function() {
                paste("Waited", now(bank) - get_attribute(bank,
"start_time"), "I am off")
              })) %>%

  simmer::select(c("counter1", "counter2","counter3"), policy =
"shortest-queue") %>%
  seize_selected() %>%
  renege_abort() %>%
  log_(function() {paste("Waited: ", now(bank) -
get_attribute(bank, "start_time"))}) %>%
  timeout(function() {rexp(1,1/mean_service_time)}) %>%
  release_selected() %>%
  log_(function() {paste("Finished: ", now(bank))})



bank <-
  simmer("bank") %>%
  ### Adding resources (counters with respective number of
servers)
  add_resource("counter1",2) %>%  ### 2 is for number of servers
per queue
  add_resource("counter2", 2) %>%
  add_resource("counter3", 1) %>%
```

```
  ## generating customers
  ## If number of customers follow poisson distribution, then
the interarrival time follows a exponential distribution.
  add_generator("Customer", customer, function() {c(0,
rexp(1000, 1/customer_inter_arrival_time), -1)})

## WSimulating above set up many times

envs <- mclapply(1:no_of_simulations, function(i) {
  bank %>% run(until = simulation_time) %>%
    wrap()
})

#=======================================================

## Estimating Target Parameters
# ---- waiting time
waiting_time <- function(x) {
  a = get_mon_arrivals(x) %>% filter(finished == TRUE,
activity_time > 0 ) %>%
    mutate(waiting_time = end_time - start_time - activity_time)
  return(mean(a$waiting_time))
}
x  = lapply(envs, waiting_time )

waiting = c('Waiting_time', mean(unlist(x)),mean(unlist(x)) -
1.96*sd(unlist(x)),mean(unlist(x)) + 1.96*sd(unlist(x)))

# ----- % customers served

customers_served <- function(x) {
  a = get_mon_arrivals(x) %>% filter(finished == TRUE)
  y = (length(which(a$activity_time > 0))/nrow(a))*100
  return(y)
}
x  = lapply(envs, customers_served )

srvd_cust = c('Percentage_of_customers_served',
mean(unlist(x)),mean(unlist(x)) -
1.96*sd(unlist(x)),mean(unlist(x)) + 1.96*sd(unlist(x)))
```

```
# -------- mean time in system

system_time <- function(x) {
  a = get_mon_arrivals(x) %>% filter(finished == TRUE,
activity_time > 0) %>%
    mutate(sys_time = end_time - start_time) %>%
    filter(finished == TRUE, activity_time > 0)
  return(mean(a$sys_time))
}
x  = lapply(envs, system_time )
system= c('Time_spent_in_system',
mean(unlist(x)),mean(unlist(x)) -
1.96*sd(unlist(x)),mean(unlist(x)) + 1.96*sd(unlist(x)))

#### -------- Queue Length
qu_length <- function(x) {
  a = get_mon_resources(x)
  return(mean(a$queue))
}
x  = lapply(envs, qu_length )

qu = c('Queue_length', mean(unlist(x)),mean(unlist(x)) -
1.96*sd(unlist(x)),mean(unlist(x)) + 1.96*sd(unlist(x)))

#### -------- utilisation

utilisation <- function(x) {
  a = get_mon_resources(x)
  return((sum(a$server)/sum(a$capacity))*100)
}
x  = lapply(envs, utilisation)

util_res = c('resource_utilization',
mean(unlist(x)),mean(unlist(x)) - 1.96*sd(unlist(x)),min(100,
mean(unlist(x)) + 1.96*sd(unlist(x))))

##### Combining all metrices into a dataframe

final = data.frame(rbind(waiting, system,srvd_cust, qu ,
util_res ))
```

```
colnames(final) = c('Parameter', 'Expected mean',
"95%_CI_lower_bound", "95%_upper_bound")
final[,-1] = apply(final[,-1], 2,as.numeric)
final[,-1] = round(final[,-1],1)
rownames(final) = NULL

##### Writing output to a Excel File
customer_arrival_monitor <- envs[1] %>% get_mon_arrivals()
customer_arrival_monitor <-
customer_arrival_monitor[order(customer_arrival_monitor$start_ti
me),]
resource_monitor <- envs[1] %>% get_mon_resources() %>% select(-
c(queue_size,limit))

### Writing monitor data for a single simulation
require(openxlsx)
list_of_datasets <- list("customer_arrival_monitor"
=customer_arrival_monitor, "resource_monitor" =
resource_monitor, "summary_post_1000_simulation" = final)
write.xlsx(list_of_datasets, file = output_file_name  )

##########   Evolution of waiting time plot

arrivals <- get_mon_arrivals(envs)
plot(arrivals, metric = "waiting_time")+
  labs(subtitle="Includes all customers (both served and not
served)",
       y="Waiting Time (mins)",
       x="Simulation Time (mins)",
       title="Waiting Time Evolution")

##=================================================
```