# Assignment 3

Computer Science 531/601

Due: 23:55, Friday November 13, 2015

Instructor: Majid Ghaderi

# 1 Objective

The purpose of this assignment is to learn how to use simulations for system optimization. You will analyze the output of your simulation program to find the optimal configuration for the system under study. The system considered is a variation of the system considered in Assignment 2. Thus, you should be able to reuse most of the code you developed for Assignment 2 and only add a few extra components required for the simulation of the new system.

# 2 System Description

Recall the Polar Burger food truck in Calgary, which was described in Assignment 2. As a result of your simulation study, the owner of the food truck decided to hire another staff to help serving customers faster. However, instead of hiring another chef, the owner decided to hire a cashier so that the current chef can focus on just cooking and making sandwiches.

The new arrangement works as follows. Customers arrive and wait in a line beside the truck. Let us refer to the customer line as the *queue*. A customer who is in front of the queue goes to the cashier window first and pays for the sandwich he/she wants to order. If the chef is idle then the customer moves in front of the chef window and waits there for his/her sandwich. If the chef is busy, *i.e.*, there is already a customer waiting in front of the chef window, then the customer waits in front of the cashier window until the chef becomes idle. While a customer is waiting in front of the cashier window for the chef to become idle, the cashier pauses and does not process any new customer orders.

While this new arrangement worked well for some time, the neighboring stores complained to the city that the long customer queue beside the food truck causes problems for them (*e.g.*, littering, noise, *etc*.). The city has consequently decided to charge Polar Burger a monthly fee to allow its customers wait beside the truck. The fee is directly proportional to the maximum length of the customer queue, the longer the queue the higher the fee.

The owner of the food truck has decided to limit the number of customers that can wait in the queue beside the food truck. If a customer arrives while the queue is at its maximum length then the customer will not be allowed to wait in the queue and has to go somewhere else for food. The owner thus would like to carry out a study to find the minimum queue length that results in maximum chef utilization.

# 3 Modeling Assumptions

We assume that the patience time of a customer is known at the time the customer arrives at the system. When the service starts for a customer (*i.e.*, the customer is in front of cashier or chef window), the customer waits until he/she gets a sandwich before leaving the system regardless of the customer's patience time.

For simulation purposes, we further assume that:

- The customer arrivals happen in batches (*e.g.*, customers arrive together in a car). The batch arrivals follow a Poisson process with the rate $\lambda$ batches per minute. The number of customers in each batch is uniformly distributed between 1 and $n$ customers.
- The service times of customers at the cashier window are constant and equal to $c$ minutes.
- The service times of customers at the chef window are IID and follow a normal distribution with the mean and standard deviation $\mu$ and $\sigma$ minutes, respectively.
- All customers belonging to the same batch have the same patience time. That is, either they all wait and get service or leave together without getting service. The patience times of customer batches are IID and follow an exponential distribution with the mean of $1/\gamma$ minutes. However, if at least one customer in a batch starts getting service, then no customer belonging to the same batch leaves before getting service.

## 4 Software Interfaces

Skeleton code for a Java class called `PolarBurger` is provided to you. You are asked to write code to complete the implementation of this class. You can define other variables, methods and classes if required. The following public methods in this class need to be implemented:

-
  ```
  PolarBurger(int queue, double lambda, int n, int c
              double mu, double sigma, double gamma)
  ```
  This is the constructor to initialize the simulation. The parameter `queue` denotes the queue length (in customers), while the other parameters are as defined in Section 3.

-
  ```
  double run(int time, int seed)
  ```
  Starts the simulation. The simulation runs for the specified length of time given by the parameter `time` (in minutes). The parameter `seed` specifies the seed (a positive integer) for your random number generator. You should put the code for simulation initialization and main loop in this method. It should be possible to run the simulation for different parameters `time` and `seed` by repeatedly calling this method. Once the simulation is finished, this method returns the utilization of the chef as a `double` value.

Your implementation should handle all possible exceptions/errors by catching them and printing appropriate messages to the standard output. A simple tester class, named `Tester`, is also provided so that you can see how we are going to test your code.

# 5 Report

In addition to your simulation code, you are expected to submit a report (use an editor, handwritten reports are not accepted) as well. The report should have two parts as described below.

## 5.1 Part I

Provide an analysis of the relation between the queue length and chef utilization for the system. For this study, use the values specified in the table to the right to initialize your simulation. For the queue length, start with the length of 1 customer and increase the queue length gradually until you do not see any significant change in the chef utilization. Specifically, stop if the absolute improvement in utilization is less than 0.5%.

| Parameter | Value |
|---|---|
| $\lambda$ | 1/5 |
| $n$ | 4 |
| $c$ | 1 |
| $\mu$ | 5 |
| $\sigma$ | 1 |
| $\gamma$ | 1/30 |

To compute the chef utilization for a given queue length, run your simulations 5 times (*i.e.*, 5 replications) and compute the average utilization $\overline{U}$ and its 95% confidence interval. Next, plot the average utilization $\overline{U}$ (as a number between 0 and 1) versus the queue length. Use an appropriate software to draw the plot. For each point on your plot, depict the corresponding confidence interval as well.

## 5.2 Part II

Provide answers to the following questions:

1. **Random-Number Streams:** Each simulation run requires multiple streams of random numbers, *e.g.*, for inter-arrival and service times. Explain how you have generated multiple streams of random numbers for each simulation run.
2. **Length of Runs:** How long was the total length of each simulation run? Briefly justify your decision. There is no need for initial data deletion in this assignment. Collect statistical data from the beginning of the simulation.
3. **Number of Runs:** It is desired to have confidence intervals whose width is less than 1. Using your results for Part I, how many simulation runs are required to achieve the desired confidence interval width for the optimal queue length (*i.e.*, the queue length that achieved the highest utilization)?