Project 1: Simulation of a Coffee Shop on College Hill

Professor Xia

Zainab Hussein

3-28-2017

Introduction

This project aims to optimize the shop profit through proper decision of quantity of cashiers to order. The constrains are set for a 16-hour day with minutes as the time unit of the simulations. To closely mimic an actual shop where events of arrival intervals for customers are random, and the service time depending on the customer, these two parameters are calculation using a Poisson distribution. At the end during runs, three of the parameters (profit per customer, daily cost of a cashier and average customers served per minute) and the other two (average number of customers arriving per minute and number of available cashiers) are varied to realize the objective of this project. This program returns the arrival and departure times of all customers, the amount of daily net profit, the rate of overflow when customers are turned away and the average and maximum wait times of all the customers.

The data used for this report was generated from Cameron Zurmuhl's code. Specifically, using the methods running simulation which is written in the Experiment controller class. My own run simulation method did not work steadily resulting into inconsistent data, thus my decision to choose this aspect to borrow from his code. I borrowed also the orderly structure of how the data is printed in terminal.

Approach

There are 3 public classes and one private one, Launcher, Customertracking, testSim and Event classes respectively. The Launcher class is just a separate main class with a purpose of testing the rest of the simulation for flexible input arguments. CustomerTracking class keeps track of the time status of a customer, such are arrival time, wait time, service time, depart time and turn away time. CustomerTracking is used in the testSim class which contains a private Events class. My testSim is like the one discussed in [2] to run an event driven simulation which prioritizes customers based on their arrival time. Average and maximum wait times are calculated within the runSim() method, and the cost of cashiers, total and net profit calculated within their own private methods in the testSim class, then called in the runSim() when the simulation is run. I have all customers stored in an arraylist of CustomerTracking at any given time, and upon shortage of cashiers, customers are stored in a queue as they wait for a cashier to become available for serving. [1]

Methods

For examinable results, the static main method in the testSim class is examined over a range of 1 to 10 cashiers at intervals of one for every given lambda value. Each of these runs was done 5 times and an average of them taken for the data written out in table 1. Plots of data in table 1 are done to find the optimum number of cashiers per lambda value, these are tabulated in table 2 and used to analyze the optimum number of cashiers that maximize profit.

Data Analysis

Net daily profit plots against cashiers

The six plots for average net profit versus number of cashiers in figure 1 to 6 are plotted from data at table 1 in the appendix.

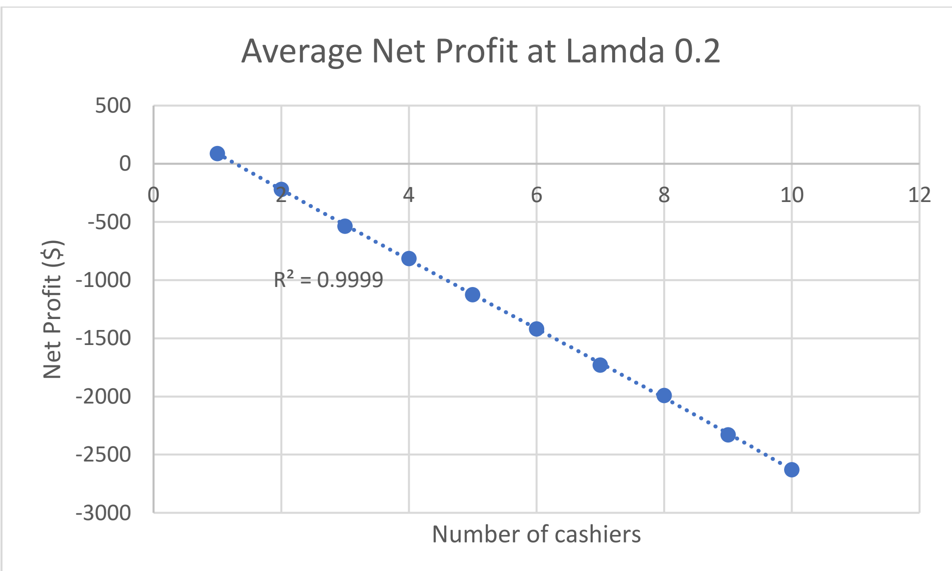

Figure 1 Average net profit at lambda 0.2

For set lambda of 0.2, the optimum number of cashiers is at 1 at net profit of $89.60. The rate of arrival is so small that it would be uneconomical to hire more than one cashier, any more results into daily loss by the coffee shop because the cost of hiring them takes away from the shop.

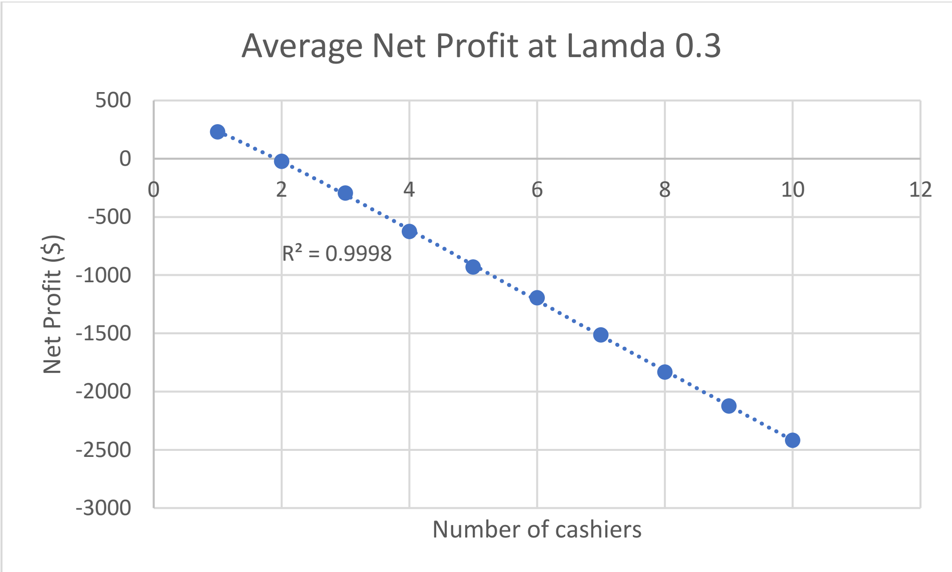

Figure 2 Average net profit at lambda 0.3

For set lambda of 0.3 the optimum number of cashiers is at 1 at net profit of $232. One cashier results into some profit, but with two cashiers there is exactly 0 profit. Any more cashiers results into daily loss by the coffee shop because the cost of hiring them takes away from the shop.

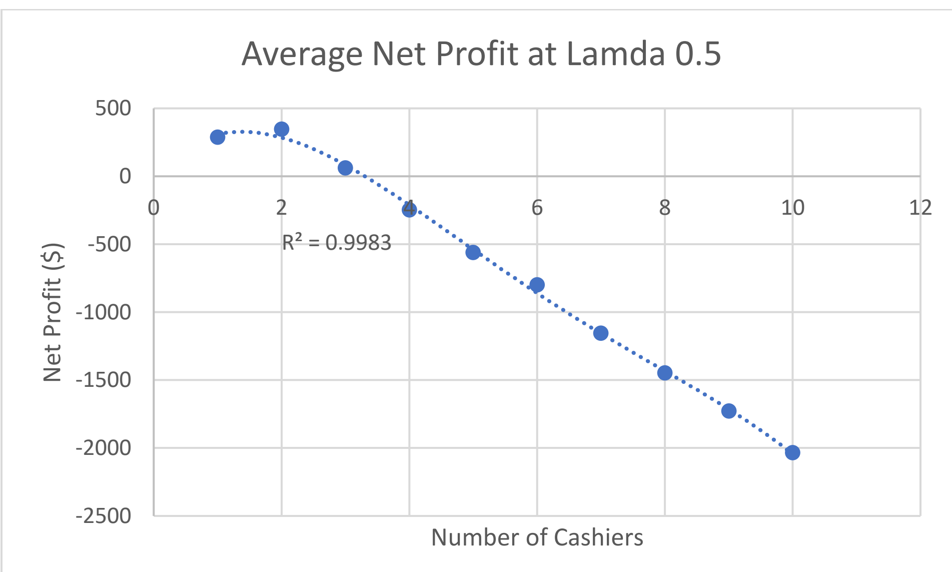

Figure 3 Average net profit at lambda 0.5

For set lambda of 0.5 the optimum number of cashiers is 2 at net profit of $345.60. From the 4th cashier, the coffee shop losses money because the cost of hiring extra takes away from the shop.

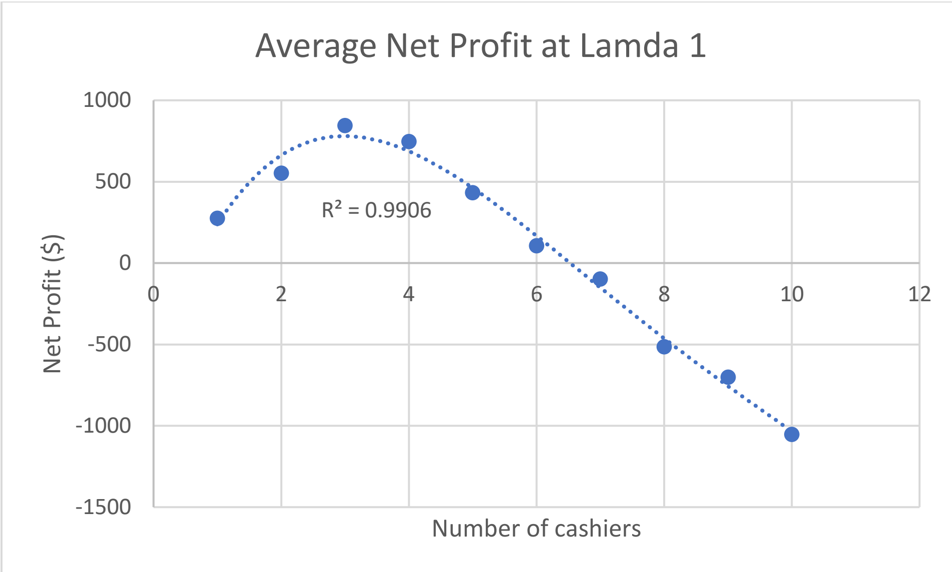

Figure 4 Average net profit at lambda 1

For set lambda of 1 the optimum number of cashiers is at 3 for net profit of $846. From the 7th cashier, the coffee shop losses money because the cost of hiring extra takes away from the shop.

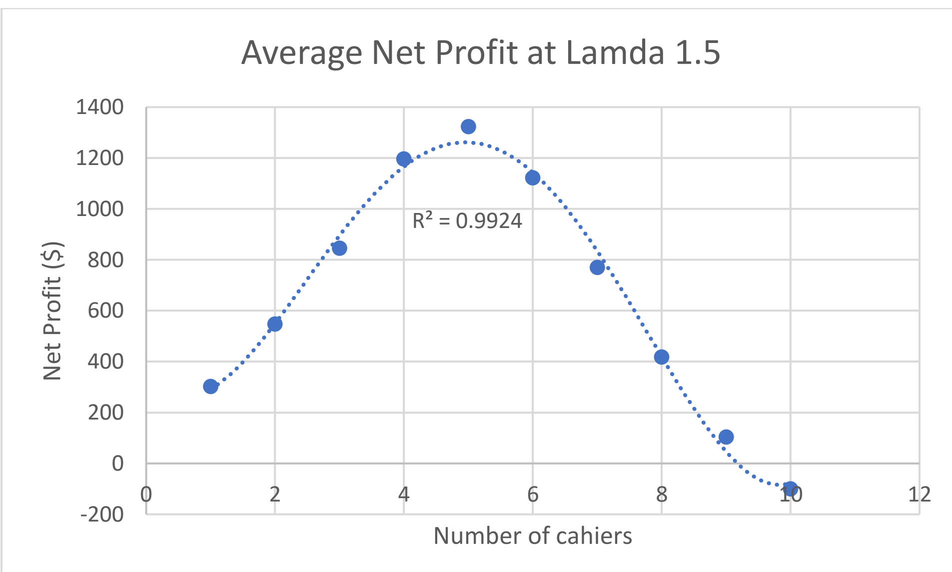

Figure 5 Average net profit at lambda 1.5

For set lambda of 1.5 the optimum number of cashiers is at 5 for net profit of $1323.60. Adding a 10th cashier to the coffee shop losses money because the cost of hiring extra takes away from the shop.

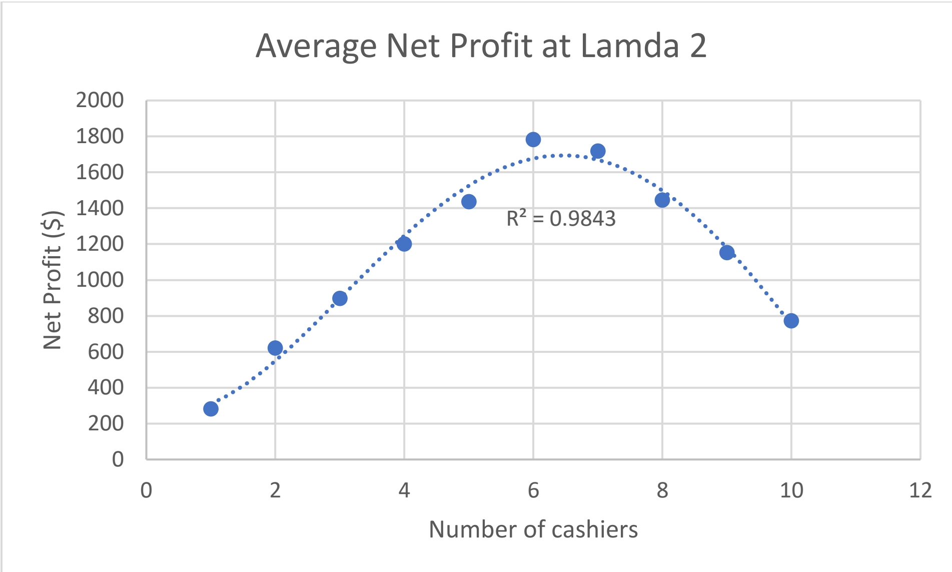

Figure 6 Average net profit at lambda 2

For set lambda of 2 the optimum number of cashiers is at 6 for net profit of $1782.40. All cashiers are used and bring some profit to the coffee shop.

Optimum cashiers that maximize profit

The optimum number of cashiers versus lambda plot in figure 7 is plotted from data at table 2 in the appendix.

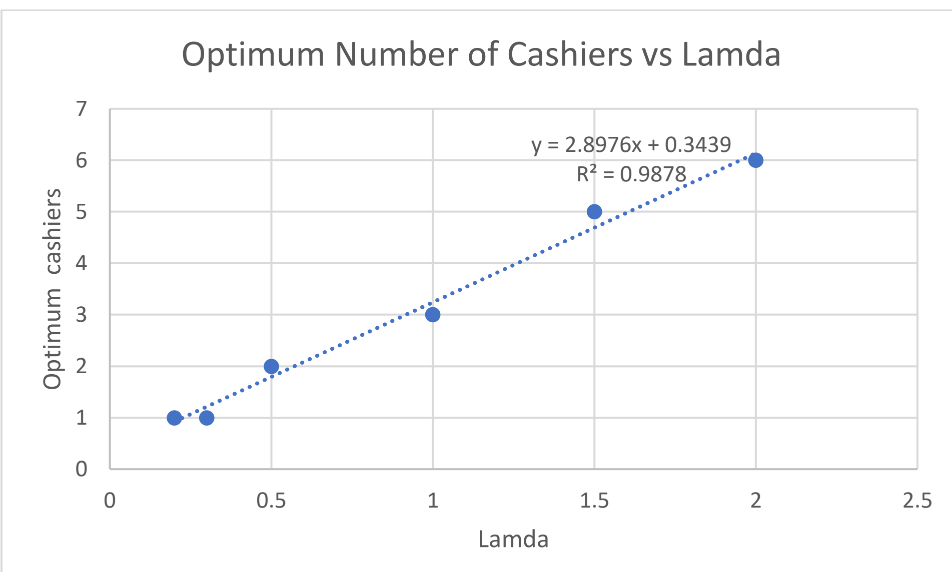

Figure 7 Optimum number of cashiers versus lambda

There is a linear relationship between optimum cashiers and lambda as expected. This suggests that with increased rate of arrival of customers, the optimum number of cashiers to serve them also should increase to maximize profit for the coffee shop.

Conclusion

An analysis of a range of one to ten cashiers and the average net profit that corresponds to them, there are clear optimum number of cashiers per rate of arrival. These optimums can be observed in figure 1 to 6. The relationship between the optimum number of cashiers and the rate of customer arrival reveal a linear relationship to maximize profit for the café that is consistent to how a real coffee shop works.

References

1Queue interface, ArrayList, LinkedList and Math classes in java 8 API

2Data Structures and Problem Solving using Algorithm, chap. 13.

Appendix

Table 1

0.2	1	2	3	4	5	6	7	8	9	10	**Cashiers**
	89.6	-219.6	-534.4	-814	-1123.2	-1417.2	-1729.2	-1990.4	-2328.4	-2629.2	**Net profit**
0.3	1	2	3	4	5	6	7	8	9	10	**Cashiers**
	232	-21.2	-294	-626	-930.4	-1193.2	-1513.2	-1833.2	-2122.4	-2417.2	**Net profit**
0.5	1	2	3	4	5	6	7	8	9	10	**Cashiers**
	287.6	345.6	61.6	-248	-561.2	-798.8	-1154.8	-1448.4	-1729.2	-2035.2	**Net profit**
1	1	2	3	4	5	6	7	8	9	10	**Cashiers**
	275.2	552.4	845.2	747.6	431.6	106.4	-98.4	-514	-702	-1053.6	**Net profit**
1.5	1	2	3	4	5	6	7	8	9	10	**Cashiers**
	302	548	846	1196.8	1323.6	1122.8	770	418.4	104	-100	**Net profit**
2	1	2	3	4	5	6	7	8	9	10	**Cashiers**
Lamda	281.6	620.4	897.2	1200	1436.4	1782.4	1718.4	1445.6	1152.4	772	**Net profit**

Table 2

Lamda	**Optimum cashiers**
0.2	1
0.3	1
0.5	2
1	3
1.5	5
2	6