Abiola Gabriel Olofin

CS 150-02

Professor Frank

04/20/2020

Project 2 Report

Introduction

In this project, we are asked to be consultants for a popular coffee shop. As consultants, we are asked to decide how many cashiers to hire in order to optimize the profit of the coffee shop. There were know conditions which included the shop opening at 6am and closing at 9pm, which customers would be included as overflow, etc. In order to decide the optimal number of cashiers to buy, we have to create an event-based simulation using a data file to compute the daily net profit, the rate of "overflow", the average and maximum waiting time of all customers served. After computing those values, we have to test different amounts of cashiers to see which number of cashiers provided the most profit and then we have to graph it.

Approach and Methods

When designing the simulation, I need a Shop class that would take in a text file that would determine the values that we need to run the simulation. This also means that this class would need to hold all the information in file in order to give the other classes the data they need.

First, I instantiated an arraylist<Event> that would hold the different events that would occur in this simulation. Then, I instantiated an arraylist<customer> that would hold all the

customers from the text file. Finally, I created different variables that would hold the different values of the text file. These variables include the lower and upper bounds of the estimated profit of serving each customer and of the average time for a cashier to serve customer in seconds and etc.

Second, I needed to create a constructor in order to give some variables values. The constructor takes in the opening and closing hours (in the 24-hour format) of the shop (i.e. 6:00am = 6, 9:00pm =21), the number of cashiers, and the name of the file with the data in it. The constructor then creates a new arraylist<Event> and a new arraylist<Customer> and those values in the parameters are given to the appropriate variable that needs to hold that value.

Third, I needed to know the breakdown of the file. The first line of the text file is the lower and upper bounds of the estimated profit of serving each customer in dollars. The second line of the text file is the cost of staffing a cashier counter per day in dollars. The third line is the lower and upper bounds of the average time for a cashier to serve a customer in seconds. From the fourth line to the last line, each line is the arrival times of customers in a string format "hh:mm:ss" in the 24-hour format. After knowing the breakdown, I created a method called runFileReader() that reads the file and sets each value needed for the simulation and starts to create the events of the simulation using the method createEvents(). In createEvents(), we are creating the an event based on the hour of the day and the customers. This method then runs the method runEvents() which runs each event in the arraylist<Event> and sets values such as daily net profit, average customer service time, and etc.

After the Shop class, I created the Event class. This class stores what happens in what event of the shop. The constructor takes a parameter of an int and uses it as its time stamp for

the hour of the day. The constructor also creates a new arraylist<Customer> to hold customers at that specific event. The run() method returns the total profit from that event and sets values such as the overflow rate and the serving time.

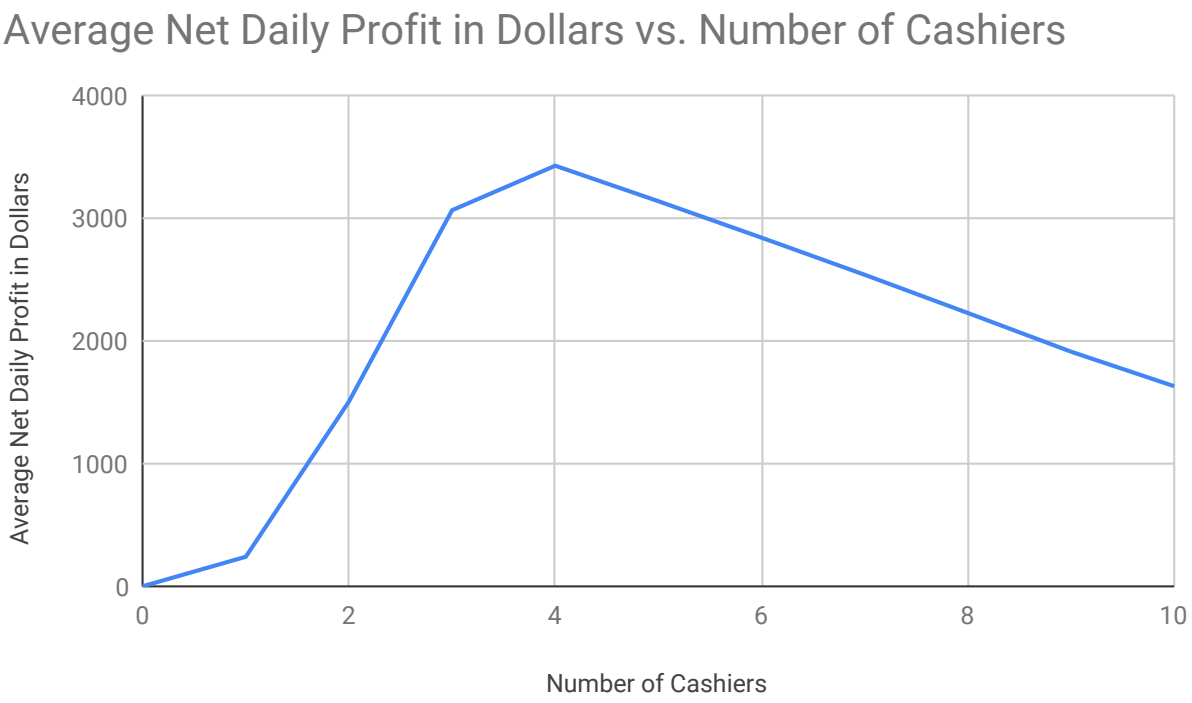
The next class was the Customer class. This class stores what happens to a customer and the arrival time of a customer from the text file. This class implements comparable in order to compare the times of the customers in the compareTo() method. The Customer constructor takes in integers that represent the customer's hour, minute, and seconds of arrival.

The next class was the Cashier class. This class stores what a cashier does at the shop. The Cashier constructor only take in one integer as a parameter which is the total number of cashiers in general. This means that every cashier knows how many cashiers there are in total at the restaurant. The constructor then goes on to create a new PriorityQueue<Customer> and set the total number of cashiers equal to the int parameter.

The final 2 classes run the entire shop and all its events while producing a CSV file that holds the data for the graph. Those classes are the Main Class which takes an integer input for the number of cashiers and returns the output based on the number of cashiers, and the CSVFile class which creates a CSV for the different trials. Finally, I used excel to generate my graph.

Data and Analysis

To collect the data needed for the graph, I created the CSVFile class that created a csv and returned the average of running the simulation 10 times based on the number of cashiers. This means that I ran the shop simulation having 0 cashiers 10 times and got the average of each output, then having 1 cashier 10 times, then 2 cashiers, etc. up to n cashiers. The CSV file shows the average net daily profit of having a certain number of cashiers from 0-10 inclusive.



I noticed that the shop which had 4 cashiers on average produced a higher net daily profit. They made $3426.20 in profits. Of course, there are outside factors that affect in real life how this data came out, but because the data is generated based on a live event, it is pretty accurate about the profit generated. Being more realistic and accounting for unknown circumstances, having between 3-5 cashiers generates good profits. I also noticed that profit increased as the number of cashiers increased; however, after 4 cashiers, the profit started to decrease at a

steady rate. This may be because sometimes having more does not always mean better and in a shop setting it might actually clog up the service process which could lead to a decrease in profit.

Conclusion

To answer the question about the optimal number of cashiers, we have to look at it from the perspective of the data and from a realistic prospective. Based on the data, it is optimal to have either between 3 and 5 cashiers or 4 cashiers because they had better profits than other options; however, the program only accounts for one data file. In order to improve it, we would need it to take in different data files and generate more accurate averages of net daily profit in order to make the graph more representative of the coffee shop's customer pattern. This will allow for a better representation of a shop's profits and current state. It would allow for more analysis of different factors and also be a driving force for decisions in the shop.