Cafeteria Simulation

Jan Johannsen

January 2024

Contents

1 Introduction **1**

2 Problem Description **1**

3 Software structure **2**

4 Experiment Description **3**

5 Results **3**

List of Figures

1 Cafeteria structure example 2
2 Equal distribution result 4
3 Normal left distribution result 4
4 Bimodal normal distribution result 5
5 Triangular distribution result 5

1 Introduction

Be it at work or in school, most people visit the cafeteria during pauses throughout their day, to take breaks and get something to eat during the day. While it might seem like a visit to the cafeteria is rather mundane, the problem of serving a large number of customers during short time frames is more complicated than one might think. In the following I will discuss how a possible simulation for a cafeteria could look like.

2 Problem Description

A common goal when visiting the cafeteria is minimizing wait times. This holds true for both the staff and especially the customer. In this paper decisions inside the cafeteria will be described through choices, which will branch into either a new choice or a station inside the cafeteria. Each branch of a choice is given a percentage with which it will be followed. When moving to a station there are two options, either the station is empty and the customer can start the process at the given station or another customer is currently blocking the station and new customers have to form a queue and wait until their turn. Each station will contain two values deciding the time a customer will spend during the entailed process. First a flat delay and second a value describing by how much the flat delay can increase or decrease. For example getting food at a station could take 60 seconds with the potential of being 20 seconds faster or slower depending on the customer. Finally when paying for their meal customers won't be sorted based of chance, but will rather be moved to a cashier based on the length of the queue. The figure 1 contains an example structure for a cafeteria, which will be used in the experiments for the software

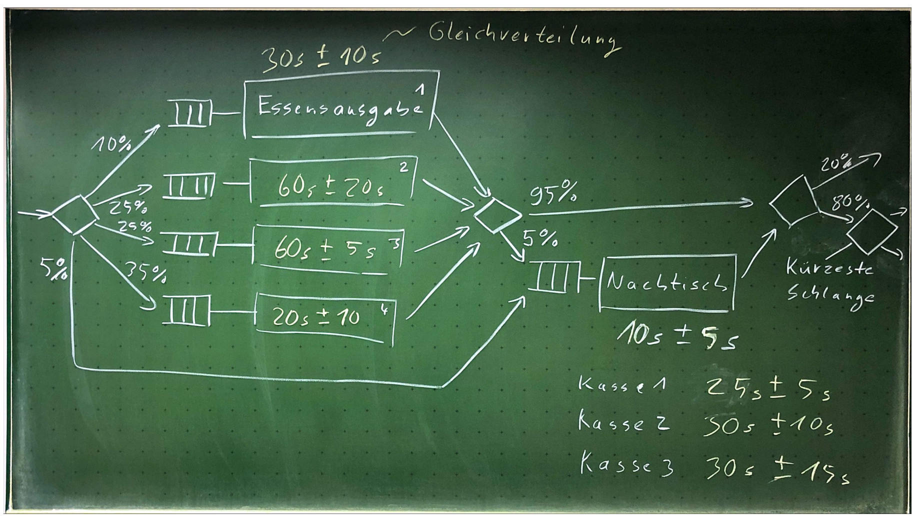

Figure 1: Cafeteria structure example

3 Software structure

Given a structure for the cafeteria, one possible way to simulate the cafeteria is using events managed by a central manager to move students from station to station. For this to be possible the software first needs stations and choice gates. Each station contains a queue to hold students, values for the delay and a reference to the choice gate following the station. In addition to this student objects will be used to fill the queues at the stations. Each student object will contain an id, their time of entry and timestamps set by the stations on entry into the queue. Students moving from one station to the next one will be managed through add and pop events, with an add event telling the eventmanager to add a student into the queue of a following station and a pop event telling the eventmanager to figure out the following station for a student who finished their process at a given station. The eventmanager contains two methods to add and execute events. When executing events the eventmanager will call methods from the specified station to either add or pop a student from the queue. If the queue was empty when adding a new student, the station will return a pop event for the time the student finishes the process at the station. If the queue is not empty and the station is blocked no new pop event will be created when adding the student since a pop event for this specific station will already be in the event queue. When popping a student from the queue inside of a station a new add event for the following station determined by the choice gate will be added into the event queue. In addition to this a new pop event will be added if there is a following student inside the queue of the station.

4 Experiment Description

To test the described implementation one can now set up different test cases for student behaviour. Example distributions for arriving students used in the following experiments are equal, triangular, normal and bimodal normal distribution. For each case the distribution is imported through a csv file containing the a unique id and entry time for 800 students visiting the cafeteria in a 2 hour time window. To find the best entry time for the cafeteria the average wait time for each minute will be calculated and plotted for each of the student distribution. Further numerical analysis would be preferred, but will be omitted from this paper due to time reasons.

5 Results

The results for each test case are shown in the figures 2, 3, 4, 5. Visible in figure 2 is a constant rise in wait duration throughout the two hour time window. This lines up with the constant amount of students entering the cafeteria each minute. The rise in wait time also indicates that at least one station is bit able to work through all customers before new ones arrive. This leads to this station being a bottleneck for the cafeteria. Visible in 3 is a steep rise in wait time throughout the first hour, which starts slowing down after around 30 minutes. After around 70 minutes the wait times start decreasing again. This is probably due to the bulk of the students arriving around the first 30 minute mark of the cafeteria opening. With less students entering during later times, the bottleneck is then able to finish more processes than new arriving students. For the bimodal normal distribution in figure 4 this effect is visible again. However this time the increase and decrease in time is happening twice since the bulk of students arriving is now split into two different times at around the 30 and 90 minute mark. Finally the triangular distribution in 5 shows an accelerating rise in wait time until slowing down around the 60 minute mark. This does align with the number of students entering the cafeteria increasing until the 60 minute mark with this number dropping of afterwards.

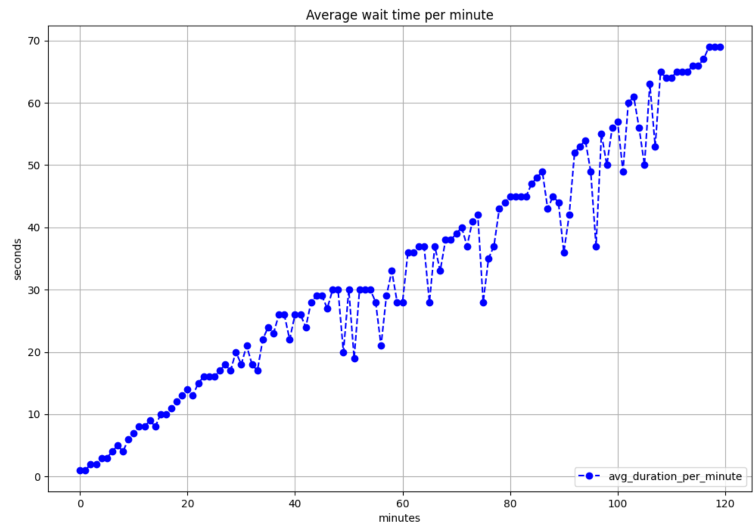

Figure 2: Equal distribution result

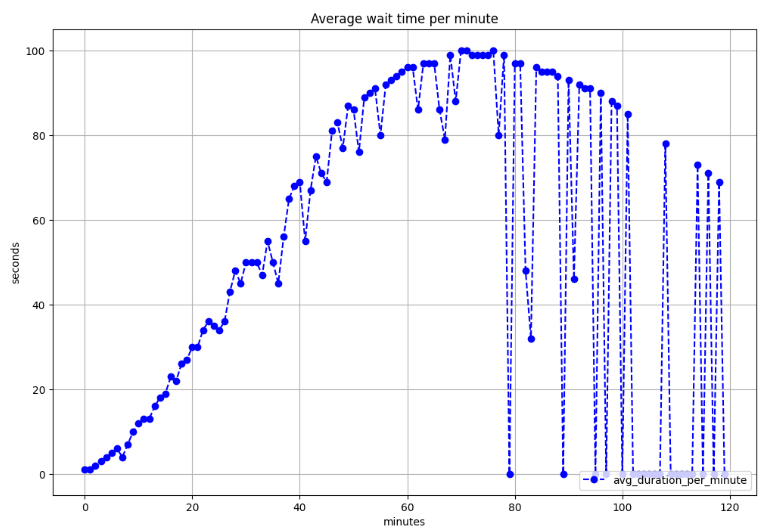

Figure 3: Normal left distribution result

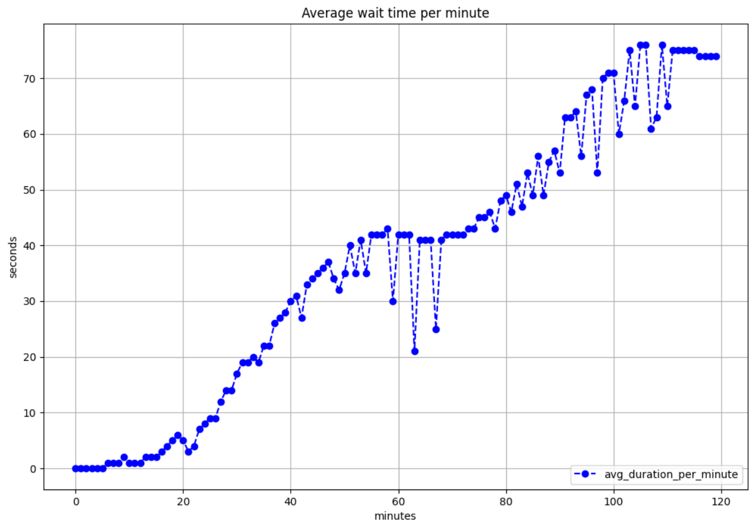

Figure 4: Bimodal normal distribution result

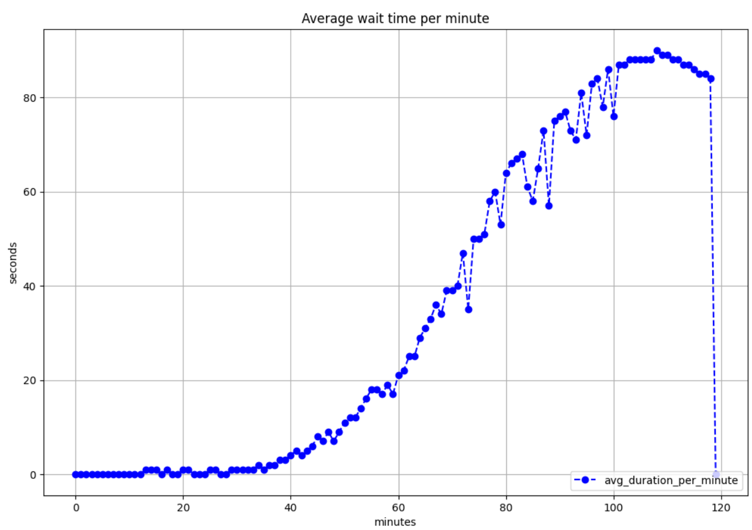

Figure 5: Triangular distribution result