Erik Wright, Eli Coyle, Aaron Noll, William Kurek

Professor Rebecca Hwa

CS 1538 Final Project

April 29, 2017

Giant Eagle Checkout Center

**INTRODUCTION:**

We based our simulation on the checkout center at a Giant Eagle grocery store. When customers want to buy items, they may choose between paying for them in a cashier lane or a self-checkout lane. Our goal was to try and improve purchasing items in two ways: 1) improve customer satisfaction by reducing the amount of time customers need to wait. 2) Improve profits by ensuring the store does not have multiple employees idle for long periods of time. With this in mind we came up with the following questions:

1. How many stations should be run by a cashier as opposed to being self-checkout? *Solving this problem will increase overall efficiency in how quickly customers can pay for their food. Additionally, if it is found that more lanes should be self-checkout than there are currently, the store could save money by hiring less cashiers*
2. What is the optimal number of lanes to have open on an average day? *The goal of this question is to find the optimal balance between maximizing speed of service and minimizing cashier idle time. Speed of service will make customers happy, while minimizing the time cashiers are idle will save the company mon*ey.
3. How many more lanes are required during rush hour as opposed to normal hours? *Solving this problem is important because rush times at the supermarket are when most revenue is generated. If an approximate number of lanes for rush time is known, scheduling can be tailored to this value which will prevent having insufficient servers (and also prevent having too many servers during slow hours). This makes service as efficient as possible while preventing the company from overspending on unnecessary shifts.*
4. What is the average number of times that an employee must provide input on a self-service station? *Self-service checkouts are not completely automated- sometimes some error will occur or a special item code must be keyed in by a person supervising the self-checkout area. There are usually only one or two of these employees working, and they often will not see a customer that needs assistance until they have waited for some time. If there are too many self-checkout slowdowns, customers will be less satisfied with service and may go to a different store. Excessive hang-ups on a high-traffic day might even mean pulling manpower away from the cashier lanes for additional assistance Reducing the number of items that require employee input or improving the software (less bugs) will reduce the number of slowdowns*

**APPROACH:**

To collect data, we visited a nearby Giant Eagle in person for a few hours. We calculated the following variables:

1. Time between arrivals into the system. This let us calculate the average inter-arrival time and fit those arrivals to a distribution. The average inter-arrival time: 22.48319328 seconds
2. The avg. amount for a cashier to handle a customer and the avg. number of items they scanned, which  let us calculate an average service rate for cashier checkout, which came out to approximately 0.273311 seconds/item
3. The avg. time for a customer to go through self-checkout (not including time spent in line) and the avg. number of items they scanned. This let us calculate an average service rate for self-checkout, which was 28.81818 seconds/item.

Unfortunately, we were not able to collect sufficient data to answer question 3. Rush times varied too much from day to day, and the data we did collect could not be satisfiably fitted to any distribution. However, we did talk to a few store managers who gave us a very useful log of data. This data helped us with the much more difficult parts of the data that we needed. The log showed us that approximately 40% of customers in self-checkout need assistance, and approximately 25% out of all customers chose to use the self-checkout lanes.

We decided to write our simulation in Python. We used the SimPy and SciPy.Stats modules to help. We designed our program to be easily modifiable in how many cashier-checkouts and self-checkout lanes there are, as well as how long the simulation runs. We also needed to keep pieces of data about each customer, such as number of items they had, speed at which they might go through the self-checkout, and their arrival time. We kept all of that data specific to each customer through the use of a Customer class that held all the data we needed. Both the cashier checkout lanes and the self-checkout lanes were modeled as FIFO queues, and each “cashier” queue had a value tied to it, which was the speed (service time/items) of service for that one cashier. Multiple different “cashier” queues had different speeds of service. “Self-checkout” queues were different. The queues had no inherent value tied with them, rather, each customer had a speed of service, and that speed of service dictated when that customer would leave the queue. In effect, the “self-checkout” queues had variable rates of both customer arrival and departure. We also needed to model the decision customers made when choosing between the two types of lane. We decided that the customer would be far more likely to choose a cashier lane if they had many items and more likely to choose a self-checkout if they had very few items.

**EXPERIMENT:**

We set the experiments up as follows:

1. *Which results in shorter wait times: cashier or self-checkout?*

The pseudo-random arrival will be the same for different combinations of checkout stations. For this problem we run the simulation for a “week” once with one cashier, once with one self-checkout, and do a paired t-test on the resulting wait times.

1. *What is the optimal number of lanes (of each type) to have open on an average day?*

Since there are multiple combinations of self-checkout and cashier kiosks, we can’t do a paired t-test. Instead, we will run the simulation for a “week” with each combination—the one with the lowest average wait time and highest number of customers served is the optimal combination

1. *How often does a customer at the self-service station require an employee’s help with a transaction?*

We can easily figure this out by running our simulation for several “days” with several self-checkout lanes and counting the average number of employee interactions/day

Experiment results can be freely viewed in the “Results” folder in GitHub; however, the most important results will be discussed in this report. As for what the optimal number of self-checkout lanes are vs. cashier-run checkout (question two), we actually found that the best result matched the layout of the Giant Eagle we collected data from. This turned out to be 12 cashier lanes and 3 self-checkout lanes. While cashiers proved to be vastly better in terms of efficiency (maximum efficiency meant having every lane have a cashier), around a fourth of the customers did want to use self-checkout. So while the answer to question 1 was undoubtedly cashier-run lanes, it is not optimal to have every lane be run by cashiers. We found that this setup was the best balance between having enough lanes for those people and minimizing wait times in both types of lanes. Calculating the answer to question 3 was fairly simple. As stated above, roughly 40% of customers in the self-checkout lanes required employee assistance. This number was given to us from a store manager. From our samples and the data we calculated, the average number of customers in self-checkouts that needed employee assistance is 39%. We can be confident that this number is correct, as we proved it was significant with a chi-square test and it also matches the number given to us by the Giant Eagle managers.

**CONCLUSION:**

It was very interesting that the setup that provided the most efficient flow of customers through the system was the current set up that Giant Eagle already had. They had clearly run similar tests to us and come to the same results. Although we did not realize it at the time, the flow of customers through the system was extremely stable. Most customers that waited in the cashier lines only waited for short times, and often lines never got longer than two or three people waiting. Self-checkout similarly almost never had wait lines, and when there were customers waiting, there were only one or two customers waiting in line to use a scanner. Customers took a long time at the self-checkout, but it was ok that they took a long time because customer arrival to the self-checkout lines was rare enough that at least one scanner would be open when a customer did arrive to use a scanner.

Although we did not run tests on them, we did notice a few things that could be improved upon; these observations were corroborated by what the store managers told us. Primarily, the software could be modified so that it is simpler for customers using self-checkout for the first time. Additionally, customers purchasing fruit or vegetables in the self-checkout lane took much longer than when buying other items. Making it less complex to weigh and scan these items could drastically reduce the amount of time customers spend in self-checkout. Overall, we were satisfied with our results as they matched both the current layout of the store as well as the information given to us by the managers.