Schenley Café Student Efficiency Comparison: Previous Versus Current Set-up

Andrew Harper, TakYee Ho, Orin Guo

Zip file Emailed by 11:59 Sat 14 Dec 13

Abstract:

Many students visit the Schenley Café, a school cafeteria with several food vendors, for their dining experience at the University of Pittsburgh. In previous years, it was very convenient for a student to get a variety of foods from any of the food vendors and pay for their overall purchase at the end. The set-up of Schenley Café has changed in which one must pay at each individual food vendor. The goal of the simulation is to determine whether the new set-up is more time efficient for the average customer. As frequent Schenley Café customers, Andrew, TakYee, and Orin feel as though the new set-up is more time consuming for any or all the customers.

Introduction:

The experiment consists of simulating and comparing Schenley Café’s set up from previous years to the current set up. The old set up consisted of separate lines for the food vendors, and a separate line for registers, similar to the set up of a grocery store. The new set up has a register at the end of every food line, similar to what one would see at the food court at the mall. As actual customers at Schendley Café, it feels as though the lines for every food vendor is longer with this new set-up. By gathering data based upon customer arrival rate and service time of each individual vendor, two models can be built that resembles the old versus the new set-up of Schendley Café.

The old set up made it easier for students to grab a variety of different types of foods from multiple vendors, and be able to check out quickly at the separate paying lines if they wanted to just “grab a quick meal and go”. Using this same case in the current/new set-up, the customer would need to wait in several different lines for each of the separate food vendors, making multiple smaller purchases versus one large, cumulative purchase at the end. If a customer desired a quick grab and go meal in the old set-up, they would essentially grab/get the food simultaneous to their arrival. The old set up would not require a line, and they could just simply wait in the register line. If this were the case in the new set-up, the customer would not only have to wait for the people in front of him or her to purchase their food, but also for those people to receive their food upon ordering, due to the fact that the people in front of him or her may have wanted food that was made and served rather than just a pick up. Both models seem to have pros and cons, but one must be more time efficient for the average customer than the other.

The experiment models the old set up versus the new one. The same input data is used for both models to ensure that the data itself would not become an extraneous factor or deter from our main objective of measuring time efficiency. Inputs of abnormally high and low arrival rates were used when simulating the program to see the extreme natures of the two models. The factors that will play a role in the results not only include arrival rates and service rates of the food vendors, but primarily the action that the customer wishes to take. The “action” is referring to the different scenarios a real customer do at Schenley Café, like purchasing food from one food vendor or purchasing food from multiple food vendors. The program outputs the data in an easily readable table for easy comparison. It was found that, on average, the old method was actually faster in most scenarios.

Background:

The simulation’s problem domain consists of the individual characteristics of each vendor, the logistics of the cafeteria as a whole, the level of traffic, and the distribution of various customer types. Due to the fact that none of the vendors themselves have changed, the assumption that the individual characteristics observed in the current system will be identical to the previous system is made.

With the information about each individual vendor, the change in logistics of the cafeteria from the previous system to the current system becomes the main focus. The previous system consisted of a pseudo-buffet style enclosed area. Drinks were isolated from the vendors as well as a miscellaneous array of snack items. When the customer was finished selecting what they wished to purchase, they would travel to the communal registers to pay. The previous system had between one to four registers open depending on the traffic level.

Simulation models of the previous and current system will be static and various levels of traffic with various customer type distributions will be fed in as parameters. The observed data of the simulations will be compared with actual data collected in reality to be compared. Once the verification that the simulated data is accurate with the real world, conclusions can be made regarding the effectiveness of the two systems for the typical customer.

The specific vendors of interest for the simulation consists of Taco Bell, Pizza Hut, Strutters, and Nicola’s Garden. There is also the vendor Sub Connection that shall be excluded from the simulation. Because Sub Connection was its separate entity within the cafeteria in both systems, it does not affect either system. Taco Bell’s process consisted of arriving to wait in line, ordering and paying at the register, and finally stepping aside to await the delivery of product. For Pizza Hut, the products are made ahead of time and pushed out onto a stand. The customer would wait in line, approach the stand to select a product, then proceed to the register. The assumption that Pizza Hut has infinite supply and instantaneous creation speed of product is made.

Strutters and Nicola’s Garden both consist of arriving to wait in line, ordering the product, waiting for the product, and arriving at the register to pay and order a drink if desired. There is a nearly negligible “grab-n-go” section that consists of a small selection of sushi and another “daily special” food line that all feed into the Strutters line. The simulation will consolidate all three lines into the single Strutter’s register.

Along with various levels of traffic, the customer type distribution is another aspect of interest. Simulations will be made with customers that solely wish to purchase their meal at a single vendor and customers that wish to mix their selection across vendors. Once extensive experimentation is completed, results are compared with real data of waiting times, and analyzed statistically, the conclusion of whether the previous or current system, or even a hybrid of the two systems would be best for the customers can be made.

Approach:

Before model conceptualization, data was collected at Schendley Café. Over the course of two hours on two separate days, the amount of time a customer spent in line, time it took a customer to pay, and time it took a customer to receive their food was timed and recorded for each food vendor in Schenley Cafe. Time was recorded using a stopwatch and was originally measured in minutes and rounding any time to either a full minute or half a minute. It was anticipated that service time of food service and the purchasing transaction would take a long time. It was soon realized that many of the services were much faster, therefore changing our measurement for time from minutes to seconds for more precision.

This process was repeated at all of the different food vendors. Using the chi-squared goodness of fit test, we made a histogram of the gathered data and received interesting conclusions from the test. The chi-squared goodness of fit test confirmed that the little data gathered may not be entirely too reliable because data collection was restricted to only several hours in total. In an ideal world, several weeks of data collection would have been preferred.

Once data was collected and ready to go, the model was constructed. Our program consists of two different models of Schenley Cafe, the old set up and the new one. By setting up our model in this manner, we are able to compare both models side by side for the same parameters. Something we did not account for that may happen in real life is human decision making based on the situation. If a line is very long, a person may see it and decide to go to a food vendor with smaller lines. Another situation includes the fact that a person may not wish to visit multiple food vendors because it is inconvenient to go through multiple lines regardless of how fast those lines may end up being. In the final product, the only parameter decided at run time is what the chance of visiting multiple vendors is. All other parameters are either taken from the actual data and hard coded in (such as the 1000 hour runtime), or the program already runs for multiple levels.

Another thing we decided to do was to control the rate of arrival. We chose this as our controlled variable because it’s difficult to gather arrival rate as data, and the arrival rates of students last year is inaccessible data. By controlling arrival rates we get to witness/observe and compare the extreme cases of high and low arrivals. This is beneficial because it allows examination of a vast variety of situations for both models. Extreme cases of high and low arrivals will demonstrate the true nature of the old and new set-up of the union.

Originally, the model was intended to also involve a financial element. That would involve including which set up would have made the union more money. That portion of the original plan was scrapped due to the lack in knowledge in terms of Schenley Café’s financial situation. The rates at which the servers are paid and the cost of the food they prepare are unknown. The original model was also desired to see if rates of theft had risen or lowered due to the fact that there are more cashiers. Once again, the theft rates are not accessible information. Because of this, the model was simplified to just work with and measure time efficiency.

Experimental:

Some of how the experiment was already mentioned. The model was developed and built to run the experiment basically by itself. Since multiple runs of both models are performed for a variety of customer arrival rates. The selection of different arrival rates is somewhat arbitrary, the basis simply being small intervals between different rates, increasing until the model “breaks”, or becomes unstable, in all situations. Outside the earlier small intervals, the larger number increase more rapidly as situations become unstable to avoid redundancy. The smallest was chosen on the basis the there would essentially never be a line, meaning that the average time to go through the simulation should be about the total service time.

The only other issue outside of that was how long to run it for. For simplicities sake, a significantly large run time of 1000 hours was chosen. The relative simplicity of the program mean that this was not an unreasonably long run in terms of computational time, and it was sufficiently large that multiple runs need not put together for validity, and that the particularly low arrival rates, such as 0.05, would have a sufficiently large sample size, despite so few customers per hour.

Besides the run time, another factor that is important for the variables is the chance of people going to get food from multiple places. We had no data on this due to it becoming a nonsensical practice in the current setup of the union, nor had a consensus of what was an acceptable general value to use for all situations. So that is left up to user input at run time. However, the program is designed so data on people going to multiple vendors is separate from the various specific vendors, so this variable only matters for data on going to multiple vendors and the total average. Certains tests of the model looked at runs using different values to come to a final conclusion, but the data included is two runs using a value of .1 success bias of deciding to visit another vendors using Bernoulli trials, and then another test where no one visits multiple vendors.

So with those parameters, the experiment was setup in order that each run could be taken as a valid set of data, and outside testing proved that separate runs were very similar in results. When the data is collected, a conclusion is able to be drawn from the simple averages of the situations displayed in the data. Further examination of averages could provide additional details, but with the massively large sample size simple average is sufficiently telling.

Experimental Results and Discussions:

The goal of this experiment was primarily to decide which of the two setups was faster, but to do that and be able to examine why. The results compiled will tell you exactly which set up is faster on average for both setups, in addition to telling you exactly which vendors is faster on average for the various vendors, as well as the difference in average between people visiting multiple vendors. The data is output to an html file, and is clearly labeled in what data is preset, so the table is entirely easy to read and completely clear in meaning.

The table is organized by new and old setups being separate files, with labels for the average time spent in the system for each of the 5 vendors, the average time taken by people attempting to get food from multiple places, and the average time taken by all people trying to go through the system. All times are presented in seconds, and not limited to the first few digits but rather display full accuracy. The included files containing the table presents extremely typical results to alternative runs.There is the tableNo.html file which contains the data for when a run with 0 chance of multiple venders being visited is conducted. And another table in tableMult.html has data for a .1 chance of a person deciding to visit multiple venders. First we will talk about the model with no people attempting to visit multiple vendors.

In this run the first thing to be noticed is that the old checkout system, the one with less cashiers, has a faster average run time across the board. Fortunately, the various vendors average arrival times are as expected (with the knowledge that old checkout is faster) and become unstable at the same traffic levels. By stable we mean that the time does not grow to unreasonably (an hour or more) large wait times, a phenomena which occurs at a traffic arrival rate of 2.5. This happens when the system gets new customers faster than they can serve them, and since the system runs until all generated customers are served, a backup can result in a very large average wait.

For the situation of some people trying to go to multiple vendors (at the Bernoulli success bias of .1 for adding an additional vendor to go to), the data does not tell a much different story, which is surprising. Here the old checkout was supposed to be faster in this situation relative to the difference between the systems when no one uses additional vendors. The data for this chart, in relations to the first table, is all over the place. The lowest traffic case reports similar average times which is different than the first. And shockingly and more noticeably is the fact that at high traffic level of 2.5, the table with people visiting multiple vendors has lower totals weight times, and lower times across the board. This is the exact opposite of the proposed behavior, and no valid explanation for why this has happend comes to mind.

So at first, this data is not exactly what was expected. As the new setup has a total number of cashiers double that of the old system, it is assumed superficially that it would be faster on average overall. However it was speculated that the grab all the food you want before having to pay state the old system was in could be faster in three cases. One that the traffic is sufficiently low that the limiting factor in how quickly people move through the system is food service time, and the cashiers will never form a line, except in rare cases where one vendor sees enough business that its cashier is too slow. So consider a situation where there are only 5 people in the system at a time, but all are trying to get the more popular pizza item. In that case for the old system, there would still be two available cashiers, while in the new set up those 5 people would all have to be served by the same cashier, which may cause delays.

Another important point is the behavior of two vendors feeding into the same cashier lane. This was expected to be slightly slower in the new setup, but as it turns out this was observed slightly, but only in the very low traffic situations and even then in low enough amounts that it could be attr. It is fair to attribute this to the fact that one of the two vendors that share a lane, the special place Culinary Classics, is extremely unpopular and doesn’t see much traffic causing it to be not much of an issue.

The final case is the semi obvious one. In the old setup people could get food from multiple vendors without having to go through a cashier twice. Basically the whole reason of putting this option in was to show how superior this action was for the old setup. However, vast differences in the average time it takes to get food from multiple vendors was not observed. While it did appear faster, not by a fair proportion to the data suggesting that the old checkout system was faster in the overall average, in fact the time reduction was at certain traffic levels almost the same as the old checkouts overall average time reduction.

Conclusion and Future Work:

So the rough conclusion that the data led to was that the older set up of the Schenley Cafe, was in all actuality faster on average. And not only that, it was faster in almost all cases, including ones that the new setup was seriously expected to indeed be faster.

Oddly the case that the old setup was expected to show the most gain, the Strutters and Special vendors which fed into the same line, actually seemed to show the least improvement from the old setup at the higher traffics. The actual conclusion to be drawn from this, is that there is some error present in the model, either it is design flawed of there are programming errors. In order to improve the experiment, another look at the model would be required. We would need to look at the design for any potential errors, and do any bug fixes or even write up a new program for the model. In addition the test conducted with people have a chance of getting food from multiple places shows minimal gains over the test with people not getting additional food, less gains in runs without the multiple vendor setting. All this could implies potential error in the model, either in the design of bugs in the actual code.

Most important however, is the fact that our system does not dynamically allocate workers. The number of prep chefs does not decrease or increase in the model, meaning that food prep time does not change. In a realistic scenario, food prep time would likely be dependent on how many people are working at any given time. In addition, there are scenarios which occur such as customers being allowed to pay at a different register when traffic occurs at one location in a low traffic overall situation in the new system. Our model also does not handle this, possibly explaining the received results to a certain extent.