Modeling "Fox's Hole": The Simulation of a Campus Cafeteria

By:

Ashley Davis

Allison Pitt

Emily Webelhuth

Senior Systems Engineering Design Project

Washington University, Spring 2008

Advised By: Dr. T. Eugene Day

Abstract

Written by: Miss Pitt

Computer simulation is often used to model the operations of complex systems. This paper describes the application of simulation to a campus eatery, Fox's Hole. Fox's Hole customers currently experience long wait times for service during the dinner hours. Simulation using MedModel© was used to model the current system, and then adjustments to availability of resources were made in an effort to decrease the amount of time a customer spends in the system.

The possibility of a nearby eatery, Median Market, closing has been proposed by University administration. Should Median Market be closed, its customers would likely eat at Fox's Hole instead, thus increasing the volume of customers at Fox's Hole, thus putting additional stress on the system. The operation of Fox's Hole was then modeled taking into account the increase in the number of customers that would result from the closing of Median Market. Based on the results of this simulation, the current allocation of resources at Fox's Hole would not be able to handle the increase in customers, causing the system to become unstable. Changes to the availability of resources were made in an attempt to bring stability to the system.

For both the current situation at Fox's Hole and the hypothetical case presented by the closing of Median Market, simulating various possible changes to each of the systems revealed well-defined adjustments to the allocation of resources that would improve customer experience by decreasing their time in the system.

Table of Contents

List of Figures and Tables

Figures

Tables

Introduction

Written by: Miss Pitt

A particular university campus eatery which will be referred to as 'Fox's Hole', serves as the primary establishment for students living on campus to obtain meals. During peak times, Fox's Hole is extremely crowded, and service times are long. Thus, there appears to be room for significant improvement in efficiency in terms of service times and server utilization. In addition to Fox's Hole, there is another restaurant, which will be referred to as 'Median Market', very nearby to Fox's Hole. Median Market is frequented by far fewer customers, and as a result, the university has considered closing it down. If Median Market were to be closed down, it would likely increase the number of customers at Fox's Hole, thus potentially straining the Fox's Hole system further. Simulation via MedModel©, a ProModel© product, was used to model the operations at Fox's Hole. While MedModel is typically used to simulate the operation of a medical facility, its flexible structure is conducive to the simulation of many types of processes. After modeling the operation of Fox's Hole, taking into account the possible increase in customer volume that may result if Median Market were closed down, targeted adjustments to Fox's Hole will be simulated in order to determine which changes improve efficiency.

Background

<u>Fox's Hole</u>

Fox's Hole is the main dining location on the residential side of campus. Its hours of business are Sunday from 11:00 am – 1:00 am, Monday through Thursday from 7:00 am - 2:30 am, Friday from 7:30 am - 3:00 am, and Saturday from 11:00 am – 3:00 am. It is busiest during dinnertime. It features eight food stations offering different types of food a la carte. When a customer enters the system, he must wait in line at each station he desires to pick up food from. After obtaining the food from each the station of his choosing, he proceeds to one of two cashier stations. Some customers stay to eat in the dining area, while others opt to eat their food elsewhere. The dining area can accommodate approximately 150 people, however capacity is rarely a factor that customers consider in deciding whether to eat in the facility or not. Most people who decide to eat in the dining area chose to do so if they plan to eat with their friends.

<u>Median Market</u>

Median Market is located upstairs from Fox's Hole. Its business hours are Sunday through Thursday, from 5:30 - 8:30 pm for dinner, and Saturday and Sunday from 11:00 am - 2:00 pm for brunch. It features several food stations in buffet style. Upon entering the eatery, the customer, he pays the cashier a fixed price. He is then free to pick up food from as many stations as he desires. After obtaining food, while he must eat in the dining area, he is free to return to the stations to pick up additional food as he wishes. The dining area can accommodate more than 300 people, which is far more capacity than is ever needed. Median Market is particularly popular for large groups of students who meet to dine together.

Problem Statement

During the peak dinner hours, Fox's Hole is usually extremely crowded, and customers must wait long times for their food. If Median Market were closed down due to there being relatively few customers to support their operating costs, it is expected that most of these customers would instead eat at Fox's Hole for dinner, thus increasing the number of customers Fox's Hole must serve. It is suspected that by changing the allocation of resources (chefs and cashiers) the system could be made more efficient, thus enabling it to better handle the current volume of customers, and even able to handle the potential increase in customers if Median Market were closed.

Methods

Before attempting to determine what changes to Fox's Hole should be made to improve efficiency, a method of analysis had to be chosen. Both analytical (theoretical) and simulation methods were considered.

Analytical Methods Consideration

Analytical methods based on queuing theory allows for the least labor-intensive modeling of simple systems. However, few real-world systems are simple enough to be modeled this way and so in order to use queuing theory formulas, many simplifications of the systems would have to be made and the formulas necessary to model complicated distributions may still be too unwieldy. Additionally, the output of queuing theory formulas are only long-term averages, and do not give the modeler an indication of the range or variability of a particular parameter.

Simulation Consideration

Simulation allows the modeler far greater flexibility in modeling complicated systems than analytical methods do. A simulation is able to model one of infinitely many scenarios from start to the user-defined finish and collect a wide variety of metrics throughout. Additionally, the user has more freedom to build complex operations and distributions into the model than theoretical models will allow for. In light of the great complexity of the Fox's Hole system, simulation was selected as the more appropriate method for analysis.

Simulation Program Determination

There are a variety of simulation software packages available. These include those with and without a graphical representation. It seemed that a package with a graphical representation would be a great asset for the modeling of Fox's Hole because it would allow the user to visualize the bottlenecks. MedModel by ProModel was selected for the Fox's Hole simulation because it provides a graphical representation and it allows for simple coding to process various entity and resource operations, thus granting significant flexibility in modeling moderately complex operations.

ProModel

ProModel provides simulation-based decision-making tools and techniques for improving performance of various systems. MedModel is one of several products offered by ProModel for this purpose. While MedModel is designed specifically for evaluation, planning and design of healthcare facilities, its use was readily extended to suit the simulation needs for modeling Fox's Hole.

MedModel

Modeling with MedModel starts with the definition of each location in the system, with parameters such as capacity, the number of entities permitted at that location at one time, and units (essentially the number of sub-locations within that location). The locations are then connected by path networks. These consist of nodes, paths, and interfaces. Each location is assigned a node, and connected to that node by an interface. Nodes are connected to each other by paths. To move from location to another, entities and resources must move along these path networks. Resources are then defined and given icons. Additionally, information like the quantity of each resource and the node at which it is based is input. Entities are defined and given icons, and then their arrivals are created. For each entity, the modeler specifies how many entities arrive at one time and the frequency at which arrivals occur. The greatest complexity in model assembly is presented by the processing of entities in the system. This involves giving the entity directions on where to go and what to do while in the system. All of these aspects of building the model will be subsequently explained in great detail in the context of the Fox's Hole system.

Data Collection

Conducted by: Miss Davis, Miss Pitt, and Miss Webelhuth
Written by: Miss Davis

Raw data was collected by hand at Fox's Hole from January to March. This was essential because in order for our simulation to produce valid results, we needed as much information about Fox's Hole as possible. The following subsections explain in more detail how we went about collecting all the data for this project.

Interarrival Data

In this instance, it was not possible to physically keep track of the time between arriving customers. This is because many customers arrive as a group or come at such a fast pace that obtaining non-zero and accurate interarrival times would be quite difficult. Instead, we decided to break the four hour dinner/simulation window into five minute increments and count how many customers arrive during each five minute period. In taking such data over the course of multiple weeks, we were able to find an accurate average number of customers that arrive at Fox's Hole for each five minute interval of the simulation window. Additionally, since the two entrances have different arrival frequencies, arrival information was collected via this process for both doors separately.

Service Times

Fox's Hole is primarily a 'made-to-order' campus eatery. Therefore, for our simulation to behave like the eatery, we needed to have some idea as to how long each station requires to fulfill each order. We manually timed with stopwatches how long it would take for every customer to complete an entire service process at each station that requires assistance. By 'entire service process,' I am implying that we will start timing the process when the customer approaches the ordering station whether or not the station's chef is prepared to service them or not and stop the timer once the customer receives their food. While this was our methodology, we realized when collecting data that in restricting our simulation to Fox's Hole's busiest time of the day, dinner service, station chefs rarely ever neglect their posts due the long waiting lines. Hence, the data we obtained mostly resembles that of the individual's 'made-to-order' food service distribution with a few 'entire service process' time outliers. Further, every station has an extensive menu. Due to this, we must obtain enough service time data to incorporate the times of each food item's service time into the overall station distribution. In total, we completed this service time collection process for the following stations: Pasta, Deli, Tacqueria, Fryer, Grill, Salads, Cashiers, and Soda. This process was also applied in the Dining Area so that we could find an accurate distribution for dining times as well.

Resources

We observed how many staff members were on duty during dinner service and where they were located. For the most part, we found that every location that required supervision was monitored by one staff member with one notable exception:

Cashiers: Fox's Hole is equipped to have two Cashiers on duty at all times. By observation though, we realized that when the Cashiers' line was not very long, one of the Cashiers tended to wander away from their post either for a break or to restock various items around the dining hall.

Fryer Chef: The Fryer Station is usually covered by one chef, but occasionally, this chef must leave the station unmonitored while he goes to the kitchen to replenish food items.

Customer Preferences

Fox's Hole, as aforementioned, presents customers with a variety of cuisine options. Thus, to determine an accurate arrival rate to every station, we needed to find the probability that an entering customer will approach every station. Our method to find this probability was to keep a tally of the type of cuisine customers were paying for at the cashier during various dinner services. Additionally, upon paying the cashier, customers have the option of leaving, getting a drink and leaving, getting a drinking and staying to dine, or simply dining in. A similar tally was collected in this instance as well and, along with our personal observances, we were able to find the probabilities for these customer's reactions as well.

Initial Model Assumptions and Simplifications

Conducted by: Miss Davis, Miss Pitt, and Miss Webelhuth

Written by: Miss Davis

The following subsections explain the various assumptions and simplifications we decided to initially define for our simulation.

Resource Availability

<u>Chefs</u>

In observing the system these past months, we found that chefs for the most part stay at their station whenever there is a waiting line for their cuisine. When no one is waiting for services, they tend to wander to other stations or even take a seat out in the dining area. To incorporate this into our model, we will assume that if no customers are arriving at a chef required station, the station's chef is allowed to wander. However, as soon as customers require service, the chef will immediately return to their post.

Cashiers

Fox's Hole is capable of having two cashiers available at the Cashier station to service customers, yet as previously stated, this station usually only has two cashiers available when the station's waiting line is long. We will therefore assume that the second cashier will only work at the station when the waiting line is longer than six customers.

Service Models

Waiting Line Logic

When a customer approaches every station, we will assume that if the chefs are busy with someone else, the customer will filter into one waiting line/queue that follows FIFO "First In First Out" logic to mirror the actual process.

Grab and Go Stations

At Fox's Hole, there are two 'Grab and Go' food stations where customers who do not have time to wait for 'made-to-order' food items can quickly obtain pre-packaged meals. For our model, we will assume that such stations are always stocked, and customers approaching these stations take a minimal amount of time at the station.

No Chef Required Stations

Similar to the 'Grab and Go' stations, there are some major cuisine stations that do not require full-time chef assistance. These stations are: Pizza, Soup, and Coffee and Juice. Additionally, by our observations, if customers do not see food available at these stations, they will not wait for the food to be replenished, but instead will just choose another station. As a result, we will treat these stations like 'Grab and Go' stations in the sense that approaching customers will have to spend minimal time at the station to obtain food items.

The Fryer

The Fryer, from our observations, acts as a combination of a 'Chef Required Station' and a 'Grab and Go' station. The Fryer chef cooks various menu items in bulk every once in a while and fills meal requests when not cooking. When the Fryer chef is serving customers, the time required to obtain meals is very low. When the Fryer chef must cook to replenish the stock of food though, customers must wait a longer period of time for their meals, and unlike the 'No Chef Required Stations,' customers seem to be willing to wait the time required. Therefore, when modeling this situation, we will assume that the Fryer chef leaves his post once every five minutes and spends about two minutes in the Kitchen preparing additional food items. While this happens, we will further assume that no customers leave the station and they will wait whatever time it takes for the Fryer chef to return.

Entity Station Decisions

As this is a 'made-to-order' eatery, we feel that it is safe to assume that entering customers will have one particular cuisine in mind of their meal. Thus, we will assume that entering customers will only go to one station before checking out of the eatery.

Model Setup: Locations and Path Networks

Conducted by: Miss Davis, Miss Pitt, Miss Webelhuth
Written by: Miss Pitt

Locations

Locations at which entities and resources are found were created in the model. The following locations were built:

- *Entrances* – Where customers enter the system. Two locations.
- *Pasta* – Food service location in which a chef serves pre-cooked pasta with the toppings chosen by the customer. Each customer is served individually, thus the capacity is one.
- *Deli* – Food service location in which a chef prepares sandwiches to customer specification. This often involves placing the sandwich in the conveyer oven. Multiple sandwiches are typically prepared at a time since the oven racks are large in size, but few in number. Capacity is eight, based on the number of sandwiches that can be undergoing preparation at one time.
- *Pizza* – Food service location in which customers may pick up pizzas that a chef has prepared in advance (self-serve). The chef rarely interacts with the customers, thus he is not modeled as a resource in the model. The capacity is one since there is only space enough for one customer at a time to select a pizza.
- *Tacqueria* – Food service location in which chefs prepare Tex-Mex-style meals to customer specification. This may involve grilling items. There is typically one chef who interacts directly with the customer, while the other chef tends to the grill. The chef tending to the grill is not modeled as a resource, since his services are not requested by customers directly. The grill chef tends to the food of many customers at once, and the capacity is set to four.
- *Fryer* – Food service location in which chef serves fried (from frozen) foods. This often requires the chef to stop service of customers in order to fry more items. Items are fried in

anticipation of near-future customer demand; only the less-popular items are made to order. Note that this chef leaves his post frequently to go to the backroom. This was modeled by having the chef wander off to the back room for two minutes, every five minutes. The capacity of this location is one, since customers are typically served one at a time.

- *Backroom* – Where the fryer chef goes when he needs to pick up more food to fry, or when he gets bored.
- *Grill* – Food service location in which grilled items such as hamburgers, chicken sandwiches, steak, and fish are served. There are two chefs – one that directly interacts with the customers (taking orders and assembling sandwiches), and another who tends to the grill. As at the Tacqueria, the chef that tends to the Grill is not modeled as a resource. The Grill chef tends to the food of several customers at once. The capacity is set to five customers.
- *Salad* – Food service location in which a single chef prepares custom-made salads for customers. Each customer is serviced individually, thus the capacity is one.
- *Soup* – Food service location in which customers have a choice of two pre-made soups. The customer must fill the cup with the soup of his choice (self-serve). There is only enough room for one customer at a time to get soup, so the capacity is one.
- *Coffee & Juice* – Drink service location in which customers may fill a cup with coffee or juice (self-serve). There is only room for one customer at a time; therefore the capacity is set to one.
- *Grab & Go* – Food service location in which customers may pick up pre-made items such as fruit, cookies, yogurt, etc. (self-serve). The capacity is three customers, since there is room for three people to select items at once.
- *Confused* – Some potential customers enter the eatery, but after contemplating the various food selections, ultimately decide that they do not want to purchase anything, and then leave the system. This is modeled by the creation of a location within the eatery called "Confused", in which customers simply stand and wait, and then leave.
- *Cashiers* – Check-out location in which customers pay for their food and drink items. There is always at least one cashier person ready to ring up the items. The second cashier person is typically not at the station unless a significant line begins to form at the first cashier. This was modeled by having the second cashier wander off when the cashier line is less than six

customers. This cashier will not return until the line for the first cashier reached six people. One person may be served by each cashier at one time, thus each is given a capacity of one.

- *Break room* – Where the second cashier goes if he feels he is not needed.
- *Soda and Pop* – The cup for the soda must be purchased at the cashier, thus the customer is not able to fill the cup (self-service) until he has checked out. There is room for one person at a time to fill their cup, thus the capacity at the Soda and Pop stations is each set to one.
- *Dining* – After entities have picked up and purchased their food and drink items they may proceed to the dining area if they chose to eat in the facility. Customers seat themselves. The dining area is able to accommodate 150 people, though many people who sit in the dining area have not purchased food.
- *Exit* – Where entities leave the system. Two locations.

All of the food and drink service stations and the cashier station have queues in which a customer must wait until the location has room in its capacity to begin serving that customer. Realistically, as the length of the queue increases at a particular station, customers are less likely to select that station for service (with the exception of the cashier station). Unfortunately, it is very difficult to model this phenomenon. Instead, balking was modeled by setting capacities for several of the queues that tend to reach excessive lengths. The following queue capacities were set: 20 for the grill, 10 for the Tacqueria, 8 for the Deli, 15 for Pasta, 15 for Salad. Pictured below in Figure 1 is the model with the locations as queues visible.

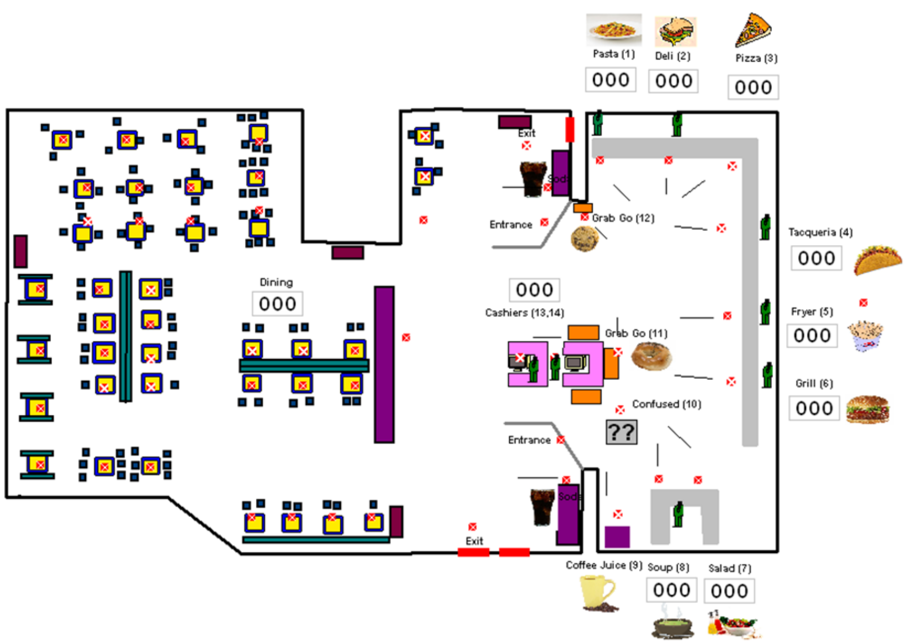


Figure 1: Model with Locations

Path Networks

To allow entities and resources to move from location to location, they must move along a "path network". Path networks are set up as lines connecting a node to another node. While paths could be used to connect each node to every other location, they need only connect nodes that an entity or resource might logically move between. The entities in the system move along the shortest path to get from one node to another. To associate a particular node with a location, "interfaces" are created that connect a particular location to the path network. Shown in the Figure 2 below is a screen shot of the model showing the locations and path network.

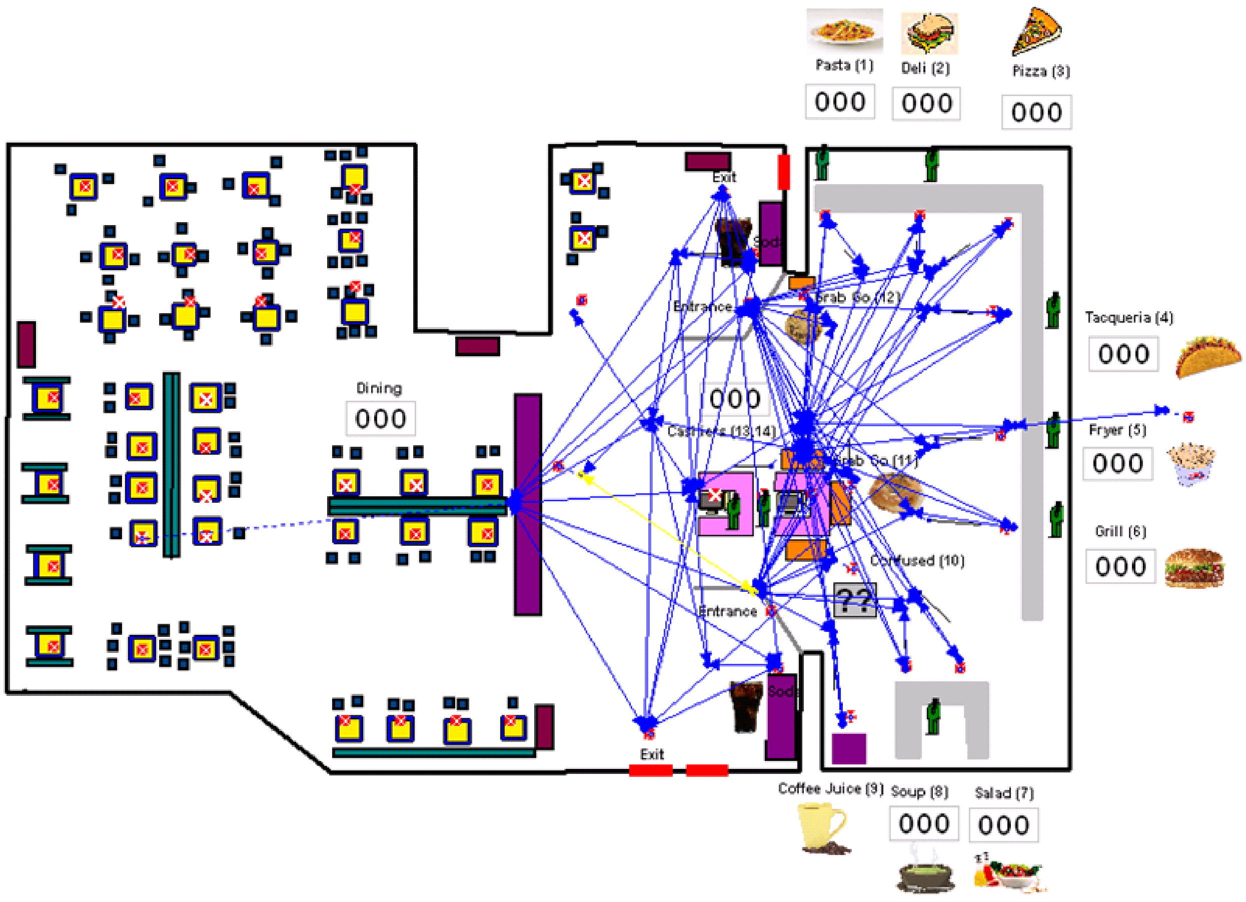


Figure 2: Model with Locations and Path Network

Statistical Evaluation of Data

Conducted by: Miss Webelhuth
Written by: Miss Webelhuth

Sorting Through the Numbers

After the raw data was collected, it needed to be organized and analyzed to become usable statistical data. We used the tool StatFit, a distribution fitting software, to help us with the task of converting the raw data into probability distributions that describe the possible paths of the stochastic processes within our model. These distributions and random processes direct the inner workings of our model, and needed to be reasonably reliable to produce realistic results.

Station Service Time Distributions

StatFit requires a minimum of 10 data points to fit a distribution, however, sometimes it requires more points to achieve the desired level of acceptability or "rank." We decided to set a goal of a minimum rank of 95% for all of our distributions to assure good fits. The following table lists the distributions we used to describe the service times at the different stations. As you can see from the ranks listed in Table 1, we were able to find distributions that accurately fit the data we collected.

Table 1: Distributions Used in Model

Station/Event	Distribution	Rank
Pasta	LogLogistic (0.2, 4.6, 0.998)	100%
Deli	LogLogistic(1, 3.07, 2.07)	99.1%
Tacqueria	Weibull (3, 2.18, 2.27)	100%
Fryer	LogLogistic (0, 3.52, 0.315)	99.5%
Grill	Weibull (1, 0.953, 5.11)	98.9%
Salad	Inverse Weibull (0, 3.95, 0.928)	100%
Soup	Gamma (0, 8.06, 0.0848)	100%
Cashier	Pearson 5 (0, 8.81, 0.992)	97.4%
Soda	Inverse Weibull (0, 4.89, 4.84)	96.8%
Dining	LogLogistic (0, 4.33, 15.7)	98.3%

Arrival Distribution Determination

The arrival data posed a different sort of problem because we did not have the required 10 data points for every 5 minute interval within the 4 hours; however, we devised way to fit the data to distributions. We realized that the distribution we used needed to be bounded so that there were no negative values and no extremely large values that could make the system unstable. StatFit offered a distribution viewer which provides a way to manually fit a distribution for situations where little data exists. The Beta distribution is a bounded distribution that is included in StatFit's distribution viewer and describes our data relatively well. We created a Beta distribution for all the 5 minute intervals for both Fox's Hole arrivals and Median Market arrivals by inputting a mean, standard deviation, lower bound, and upper bound for each. The means and standard deviations were calculated in the normal fashion for each interval, the lower bound was always 0, but the upper bound was up for interpretation. We noticed that the mean was more centered for the Fox's Hole arrival data, but for Median Market, people usually came in small numbers but sometimes showed up in large groups making the mean closer to the lower bound. Therefore the Median Market data had a bit of skewness to it. We decided to best reflect these observation, we would set the upper bound to be twice the mean for Fox's Hole arrivals, but for Median Market arrivals, the upper bound was set at three times the mean. To see our distributions in more detail, please refer to the appendix.

Model Processing

Conducted by: Miss Davis
Written by: Miss Davis

Model Processing is devoted towards designing and implementing the logic which will recreate the flow of entities and resources in the system. This logic will provide the methods by which every customer and resource will be able to move on the path networks of the simulation for every one of their desired actions. The following subsections explain in detail how the various processing elements were completed for this model.

Arrivals

To simulate the arrival of customers into our model, I used the Beta distributions prepared by Miss Webelhuth under the following methodology. Arrivals were defined in MedModel via the 'Build Arrivals' table. In the table, shown in Figure 3 on the following page, I specified the location where customers should arrive, the quantity of arrivals to be created at any one time, when the first of such arrivals are to be created, the frequency at which this quantity is produced, and the number of occurrences or total entities the arrival process will create in the model. Initially, a unique schedule of arrivals at Entrance1 and Entrance2 were created using their individual Beta distributions, but in doing so, MedModel could not function easily. Therefore, a 'Faux Entrance' was created for the model which represented the total arrivals we anticipated coming into the model from all entrances, and in using the Beta distribution schedule for total arrivals, the schedule was implemented in MedModel using the following parameters:

- Entity: Customer
- Location: Faux Entrance
- Quantity: One Customer
- First Time: Individual five minute increments between 5pm and 9pm.
- Occurrences: Individual Beta Distribution
- Frequency: 5/Individual Beta Distribution

Additionally, since the original task was to create arrivals at Entrance1 and Entrance2, arrivals from Faux Entrance were routed to either location. During the entity's routing to their corresponding entrance, customers were represented by a small dot graphic so that customers can only be seen in their true form once they reach Entrance1 and Entrance2. Finally, to determine how many customers enter through each entrance, I consulted my group and we decided that 60% arrive at Entrance1 while the remainder arrive at Entrance2.

Arrivals

Entity...	Location...	Qty Each...	First Time...	Occurrences	Frequency
Hungry	Faux_Entrance	1	Wk 1, Sun @ 05:00 PM	B(2.31, 2.3, 0.	5/B(2.31, 2.3,
Hungry	Faux_Entrance	1	Wk 1, Sun @ 05:05 PM	B(2.39, 2.4, 0.	5/B(2.39, 2.4,
Hungry	Faux_Entrance	1	Wk 1, Sun @ 05:10 PM	B(4.52, 4.49, 0	5/B(4.52, 4.49,
Hungry	Faux_Entrance	1	Wk 1, Sun @ 05:15 PM	B(4.31, 4.33, 0	5/B(4.31, 4.33,
Hungry	Faux_Entrance	1	Wk 1, Sun @ 05:20 PM	B(3.12, 3.1, 0.	5/B(3.12, 3.1,
Hungry	Faux_Entrance	1	Wk 1, Sun @ 05:25 PM	B(1.21, 1.21, 0	5/B(1.21, 1.21,
Hungry	Faux_Entrance	1	Wk 1, Sun @ 05:30 PM	B(4.96, 4.98, 0	5/B(4.96, 4.98,

Figure 3: Fox's Hole Arrival Table

User-Defined Distributions

For any major decisions customers will have to make during the simulation, User-Defined Distributions were developed to input the probabilities that customers will take certain paths in the model. The three main distributions created were the following: WhatStation( ), WhatNext( ), and WhatExit( ).

WhatStation( )

This distribution is used to determine how customers will be routed to the various food stations upon entering Fox's Hole. Following the probabilities determined during data collection, below is the distribution itself in Table 2 and an example of its implementation in our MedModel simulation in Figure 4.

Table 2: WhatStation( ) Distribution

Station	Probability	Route Number
Pasta	15.5	1
Deli	9	2
Pizza	5	3
Tacqueria	12.6	4
Fryer	6.7	5
Grill	16.9	6
Salad	13.9	7
Soup	2.9	8
Coffee and Juice	0.1	9
Confused	8	10
Grab and Go 1	5.7	11
Grab and Go 2	2.9	12
Cashiers	0.8	13

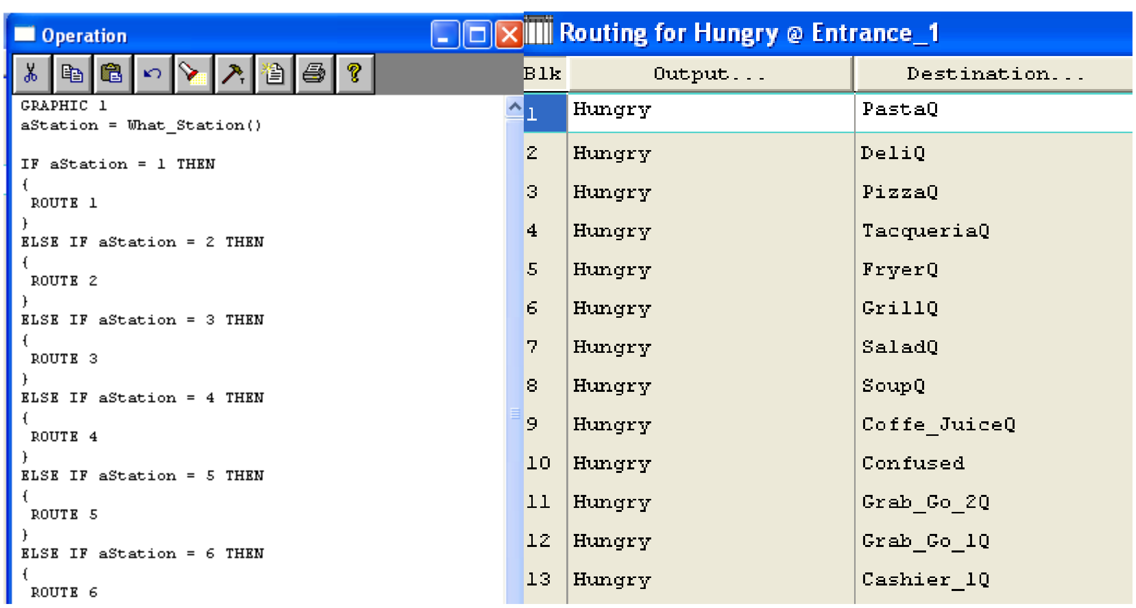


Figure 4: Portion of Routing Logic for Customers at Entrance1

WhatNext( )

Customers are presented a variety of options once leaving the Cashier stations as described in the Data Collection section. Following the probabilities determined from our data collection and observations, below in Table 3 is the distribution and in Figure 5 is an example of its implementation in the routing logic of MedModel.

Table 3: WhatNext( ) Distribution

Station	Probability	Route Number
Exit 1	9.9	1
Dining Area	16.75	2
Soda, then Dining Area	25.125	3
Pop, then Dining Area	25.125	4
Soda, then Exit 1	6.6	5
Pop, then Exit 2	6.6	6
Exit 2	9.9	7

```
aNext = What_Next()
IF aNext = 1 THEN
{
   ROUTE 1
}
IF aNext = 2 THEN
{
   ROUTE 2
   inc vNumInLineDining
}
IF aNext = 3 THEN
{
   ROUTE 3
}
IF aNext = 4 THEN
{
   ROUTE 4
}
IF aNext = 5 THEN
{
   ROUTE 5
}
```

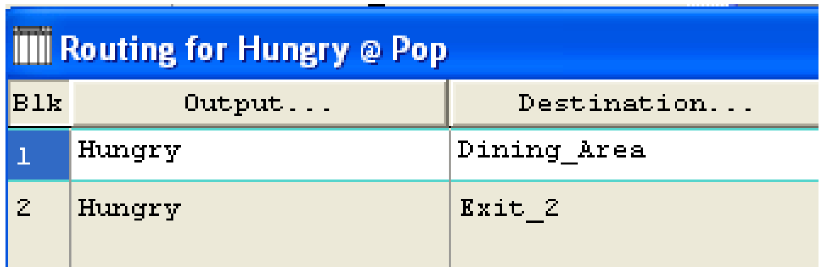
Routing for Hungry @ Cashier_1

Blk	Output...	Destination...
1	Hungry	Exit_1
2	Hungry	Dining_Area
3	Hungry	SodaQ2
4	Hungry	SodaQ1
5	Hungry	SodaQ2
6	Hungry	SodaQ1
7	Hungry	Exit_2

Figure 5: Portion of Routing Logic for Customers from Cashier station

If the customer selects Routes 3 through 6, additional routing is required in the processing logic once customers reach the Soda or Pop Stations. An example of such routing from the Pop station can be found in Figure 6 on the next page.

```
IF aNext = 4 THEN
{
   ROUTE 1
   inc vNumInLineDining
}
IF aNext = 6 THEN
{
   ROUTE 2
}
```



Routing for Hungry @ Pop

Blk	Output...	Destination...
1	Hungry	Dining_Area
2	Hungry	Exit_2

Figure 6: Routing Logic for Customers Leaving Pop Station

<u>WhatExit( )</u>

Those customers who decide to 'dine in' have a final decision as to what exit through which they would like to leave. By observation, it was decided that 60% will leave through Exit 1 and the remainder of customers will take Exit 2. Then, this logic is used to route customers as shown below in Table 4.

Table 4: WhatExit( ) Distribution

Station	Probability	Route Number
Exit 1	0.60	1
Exit 2	0.40	2

Entity (Customer) Processing

Every entity, or customer, arriving to the eatery will need processing logic for every possible action in Fox's Hole. Beyond the Arrival and User-Defined Distribution routing logic described in the above sections, the flowchart, in Figure 7 on the following page, represents the general framework of the entity processing logic that was implemented in MedModel. For your reference, stations with similar characteristics are given the same shapes which are the following:

- Multi-Step Chef Required Stations:
- Chef Required Stations:
- Grab and Go Stations:
- Combination:
- Confused Station:
- Cashier Station:

Following the flowchart, I will explain the processing logic for each of these station groups in more detail.

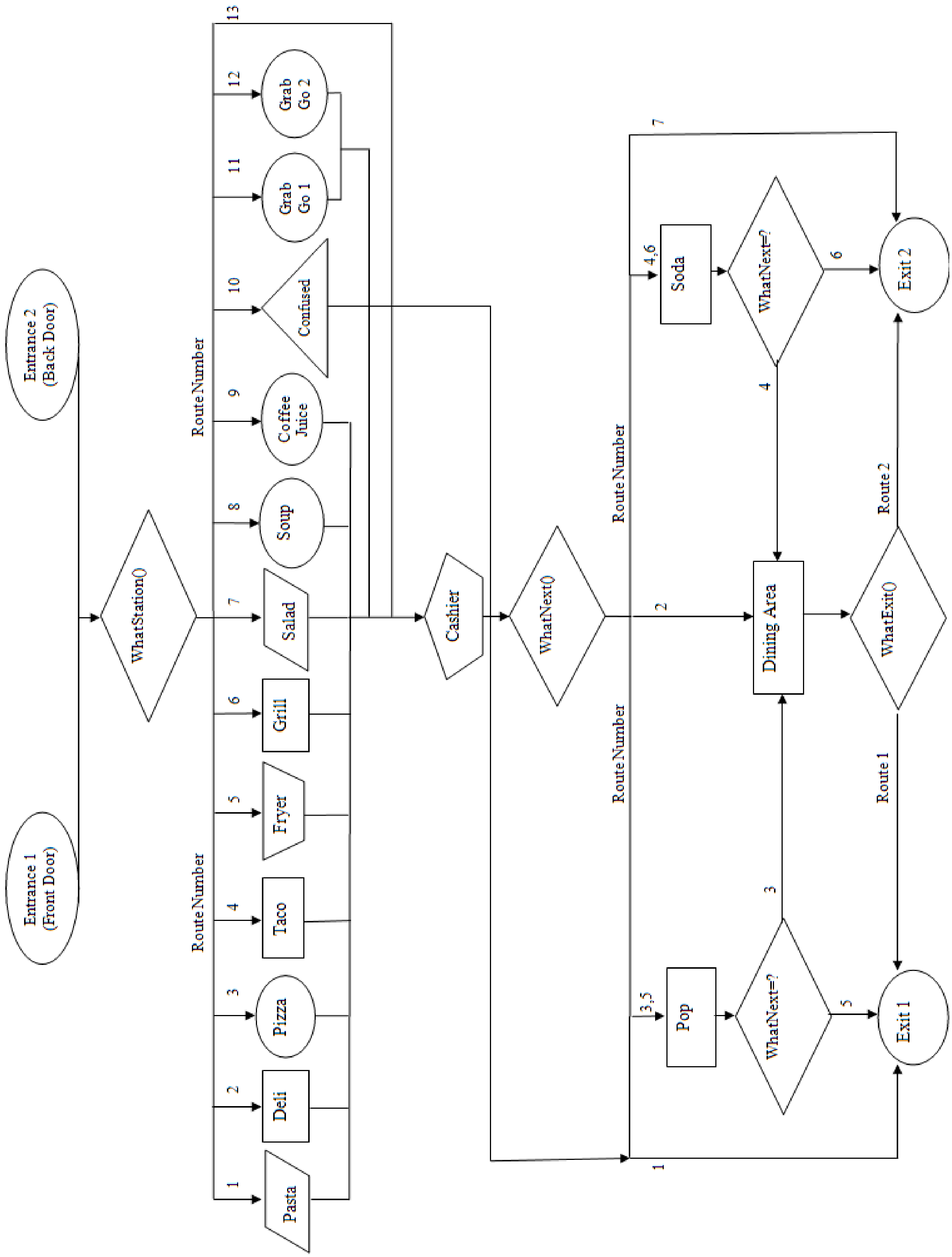


Figure 7: Flowchart of Entity Processing

Multi-Step Chef-Required Station Processing

In our case, these stations include the Deli, Tacqueria, and Grill. Such stations include food items which involve resource required preparatory/delivery services and additional cook time in an oven or on a grill that does not require chef supervision. To model this situation, prioritization was used to differentiate customers. Those customers approaching the station to order their meal possess a lower priority for the resource(s) in comparison to those customers whose meal is just coming out of the oven or off the grill. The diagram below in Figure 8 is a representation of how such logic will be implemented for each of the stations. Further, below the diagram is an implementation of such logic for the Grill station in Figure 9.

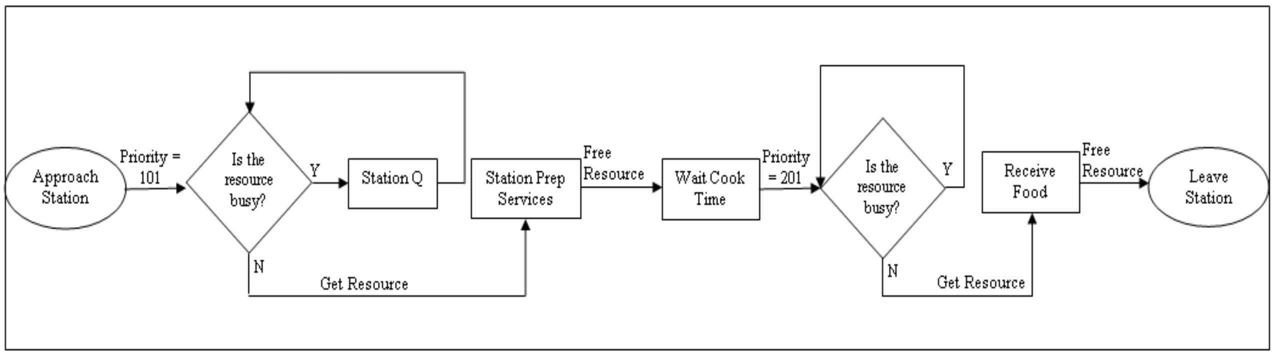
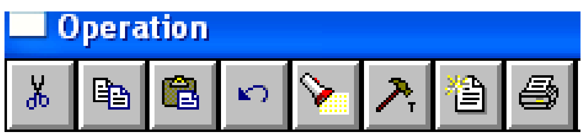


Figure 8: Flowchart of a 'Multi-Step Chef-Required Station' Processing




```
GET Grill_Chef,101
WAIT .2
FREE Grill_Chef
WAIT 0.7+W(0.953, 5.11)
GET Grill_Chef,201
WAIT .2
FREE Grill_Chef
dec vNumInLineGrill
```

Figure 9: Customer at Grill Station Processing

Chef- Required Station Processing

In our simulation, the Pasta and Salad stations are 'Chef-Required Stations.' Unlike in the previous case, customers at such stations require the resource(s) at all times to prepare their meals. No additional wait time is required for the meal to be warmed. The diagram below in Figure 10 is a visualization of how the processing logic runs in MedModel. Following the diagram is an example of this processing logic's implementation on the Pasta station in Figure 11.

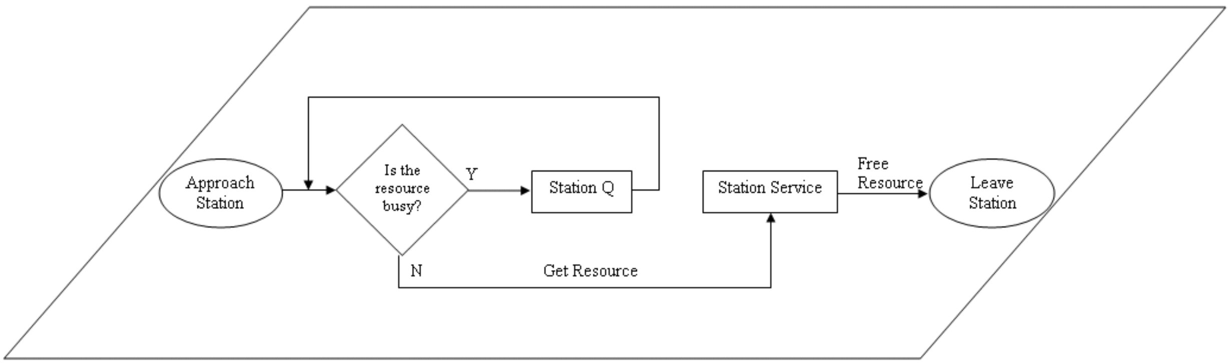
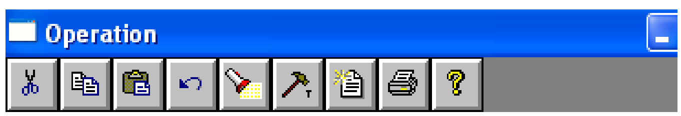


Figure 10: Flowchart of a 'Chef-Required Station' Processing




```
GET Pasta_Chef
WAIT  0.2+0.998*(1./((1./U(0.5,0.5))-1.))**(1./4.6)
dec vNumInLinePasta
FREE Pasta_Chef
```

Figure 11: Customer at Pasta Station Processing

Grab and Go Stations

In Fox's Hole, 'Grab and Go' stations require no chefs or further resources and also minimal wait time. Such stations include the two Grab-and-Go carts, the Pizza station, and the Soup station. The diagram below in Figure 12 presents how customers will react in such an instance:

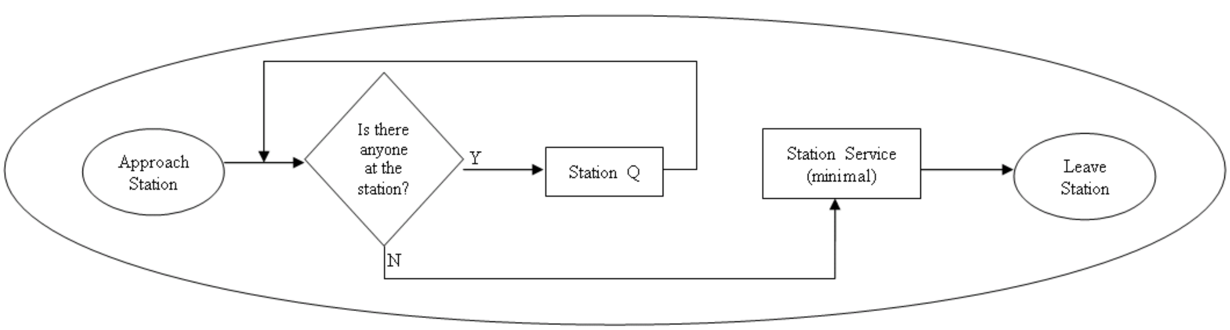


Figure 12: Flowchart of a 'Grab and Go Station' Processing

Combination Station

By 'Combination Station,' I am implying that the station's customers will require a resource to cook their food items but, since most of the food items at the station are cooked in large portions, the wait time for the customer, in most cases, is minimal like a Grab and Go Station. As discussed in the Model Assumptions section, the Fryer station acts like a Combination Station as the Fryer chef cooks in bulk a variety of fried foods every five minutes and only when not cooking is he available to service customers. To model this situation, an additional entity, '*BackRoom*,' will arrive at the location '*Kitchen*' once every five minutes. Then, every time a '*BackRoom*' entity is created, it requests the Fryer chef for two minutes which is the assumed time that the Fryer chef needs to be away from his post to replenish the food supply. Once this task is completed, the Fryer chef will be free to help customers once again and the '*BackRoom*' entity exits the system. Below in Figure 13 is a flowchart that discusses this process in more detail. Following the flowchart in Figure 14 is the processing logic in MedModel for the Fryer station.

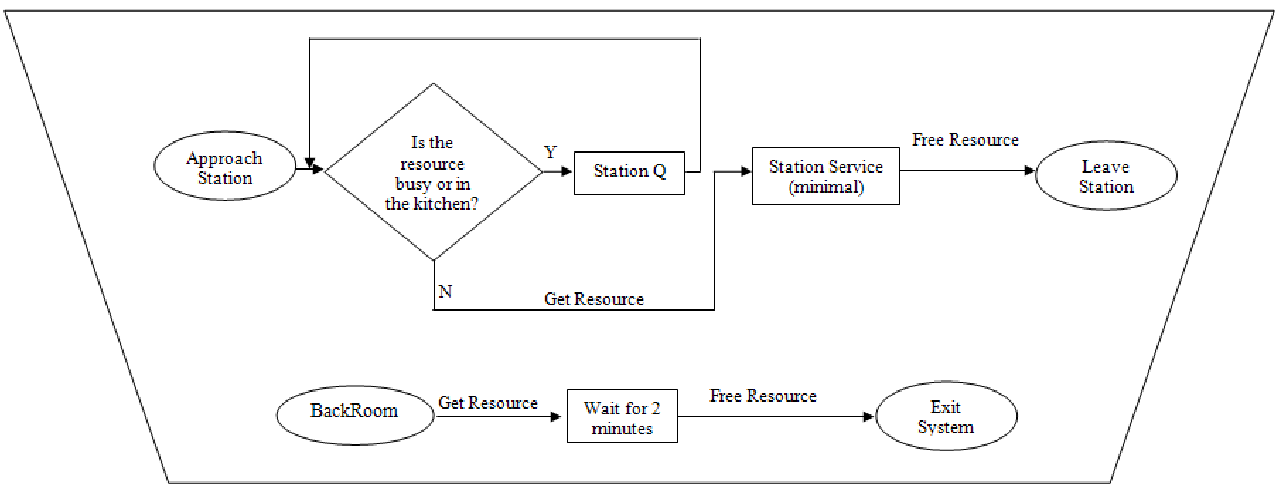


Figure 13: Flowchart of a 'Combination Station' Processing

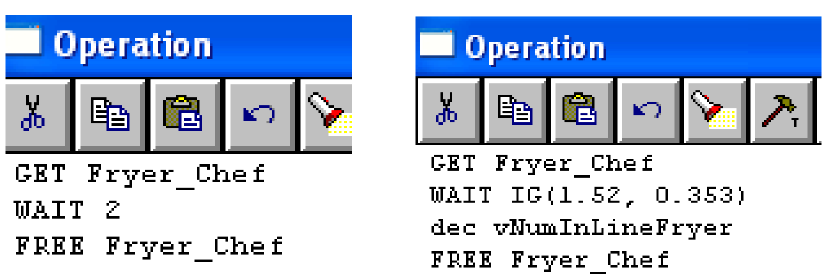


Figure 14: Processing Logic for the BackRoom (Left) and Fryer Station (Right)

Confused Station

If a customer approaches the 'Confused' station, we will handle them using the following logic in Figure 15.

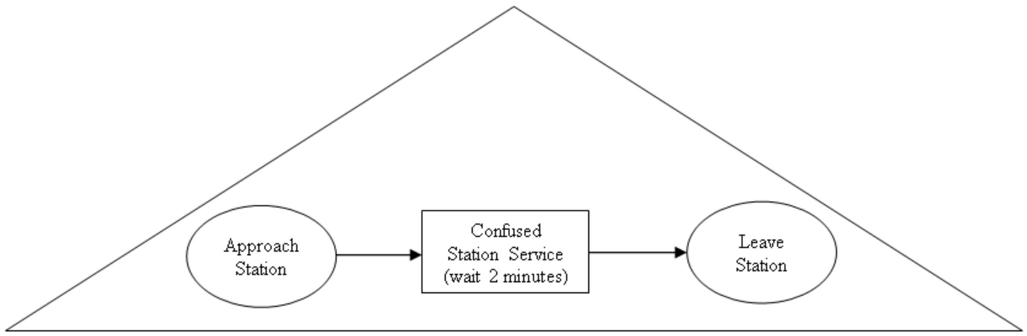


Figure 15: Flowchart of the 'Confused Station' Processing

Cashier Station

As described in the Model Assumptions section of this report, the cashiers act differently than all other resources in the model as they do not remain at their posts at all times. Instead, when the line is less than seven customers, we will assume that only one cashier is available to serve customers. To model this situation, an additional entity called '*BreakRoom*' was created at the '*BRoom*' location and whenever the Cashier station waiting line is less than seven customers, the '*BreakRoom*' will request a cashier and hold the cashier until the line becomes greater than six customers. Then, the additional cashier will be freed to help the other cashier cope with the long line of customers and the '*BreakRoom*' entity will exit the system. On the next page in Figure 16 is a diagram of how the overall process will be implemented in MedModel. Further, the MedModel processing code for the '*BreakRoom*' entity can be found following the diagram in Figure 17.

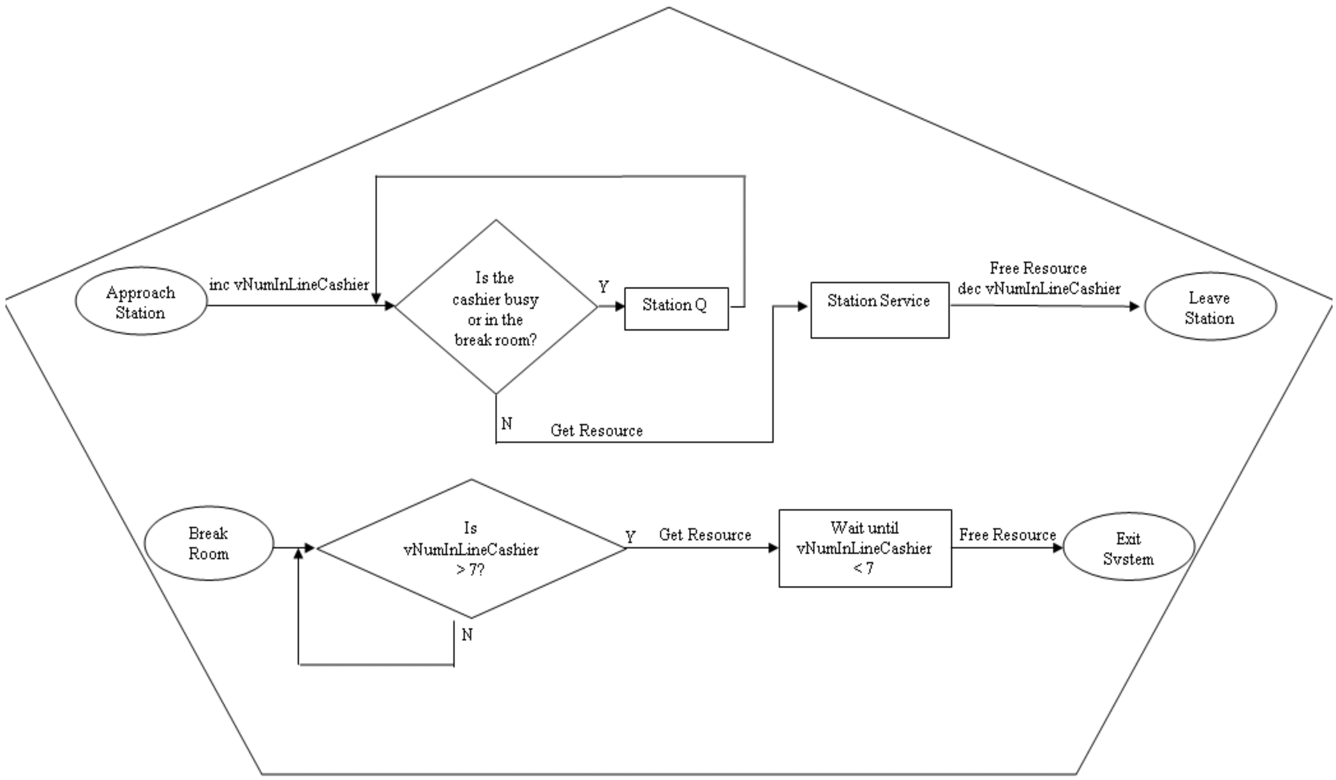


Figure 16: Flowchart of the 'Cashier Station' Processing

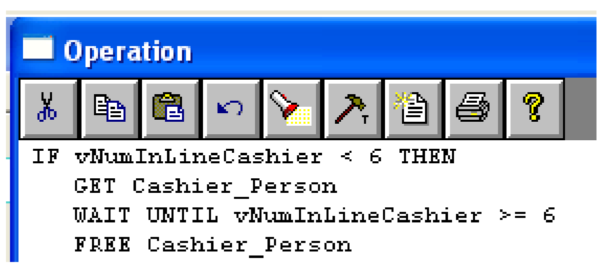


Figure 17: BreakRoom Processing

Resource Processing

Logic behind the actions of all resources is also required in this simulation. By our assumptions, all entities, or customers, will only arrive in the eatery from 5pm to 9am on a daily basis. Thus, we will schedule Fox's Hole's resources, or chefs and cashiers, to work at the dining hall from 5pm to 9:30pm so that all entering customers can be served. Additionally, rather than scheduled coffee breaks, workers are allowed to wander from their posts when no line has formed at their station. To model such actions, a shift was first defined and defined to all workers. The Shift created for use in MedModel can be found in Figure 18 below.

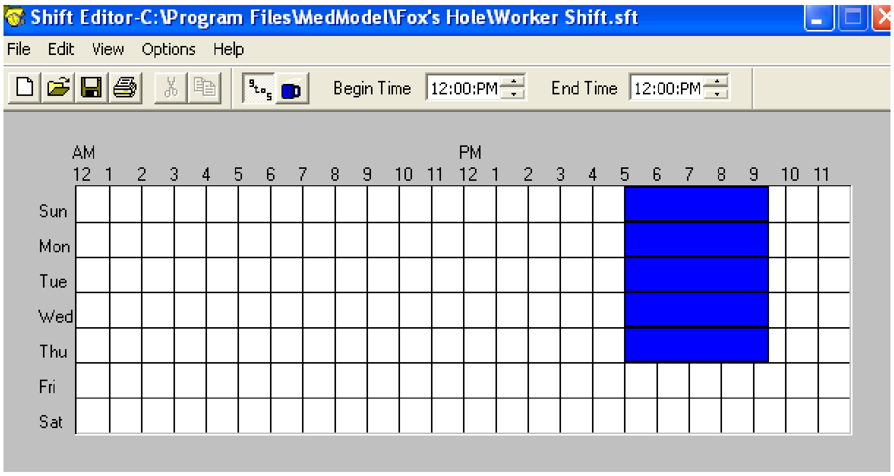


Figure 18: Fox's Hole Workers' Shift

Further, upon creating every resource and selecting he or she's 'Home' or 'Station,' we will not restrict them to '*Return Home if Idle*' to allow them to wander if they are not currently busy.

User Interface

Conducted by: Miss Pitt
Written by: Miss Pitt

The intended users of this simulation are the designers of the model, and potentially Fox's Hole management. The designers intend to use the simulation to identify changes to the system that will improve efficiency and to determine the effect of an increase in the number of customers. Management could potentially use the model to predict the effect of day-to-day changes in the system (e.g. a cashier who calls in sick).

The Graphical User Interface

To make the simulation user-friendly, a graphical user interface was developed to represent the operation of Fox's Hole. It allows the users to see a model of the system that is in an animated as opposed to a mathematical form, which will greatly increase its appeal to the non-technical managers. Users will be able to relate to an animation of the system far better than they would if the output was only numbers. For example, if a particular customer queue is getting very long, the user is able to quickly identify this as a bottleneck by simply watching the animation instead of having to look through several spreadsheets before identifying the problem.

Development of Animation

Developing this animation required rough measurements of the dimensions of the eatery and the location of the furniture and appliances in order to make graphics in the model look like the real system. Appropriate graphical representations of various system elements (e.g. customers, employees, service locations, etc.) were chosen for incorporation into the animation. Path networks over which the customers and employees move from location to location were also created.

Station Counters

To allow for easy, real-time analysis of the system, entity counters were placed at many of the service locations. Each time an entity arrives at a location (or its associated queue) the counter is incremented, and as that entity leaves the location after service, the counter is decremented. Thus the counter counts the total number of customers, in line and being serviced, at a particular location. Figure 19 is a screen shot of what the user sees as the model runs.

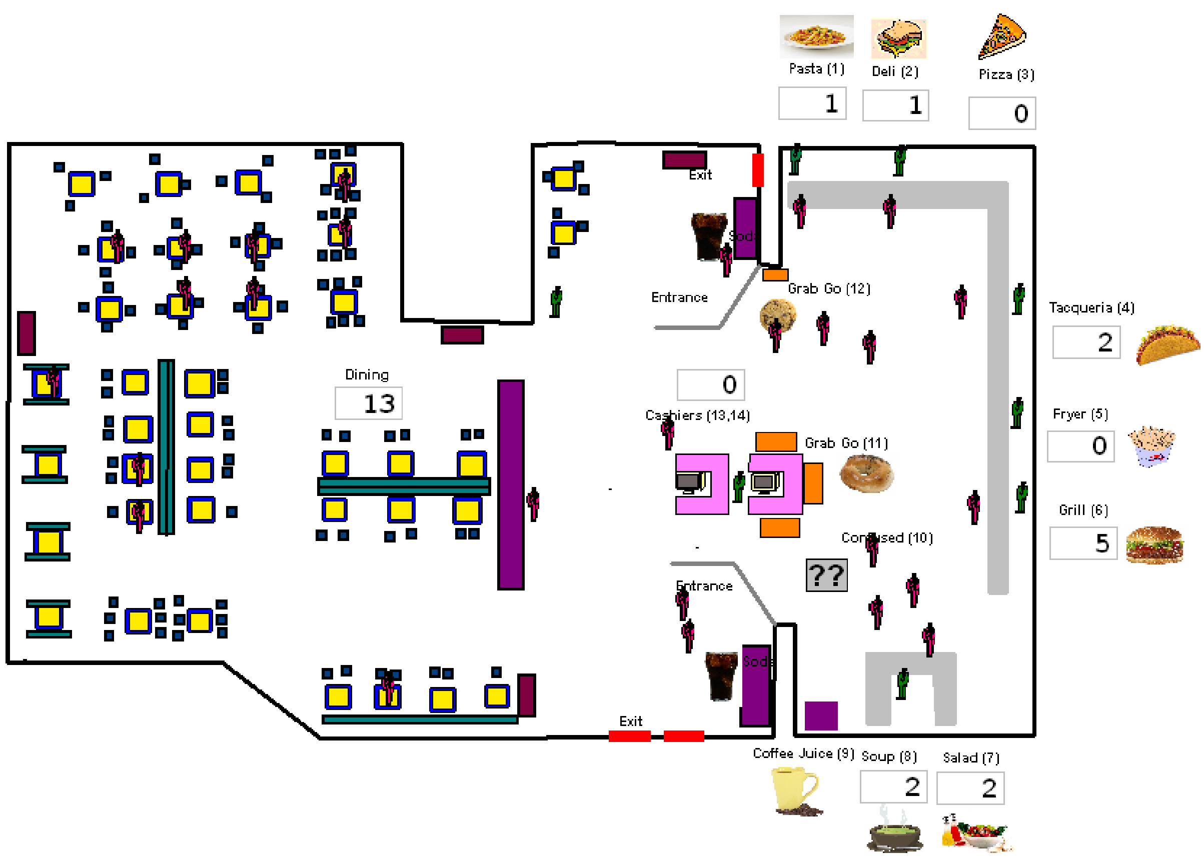


Figure 19: Running Model

Macro Definition and Assignment

To allow the user quick modification of the number of parameters at each location a graphical user interface was created that would allow user to change the number of resources and the capacity at a food station. This was done through the use of macros. Variables were created for the following parameters:

- Number of pasta chefs – *mPasta_Chef*
- Capacity at the pasta station (this should match the number of chefs since each can work on one order at a time) – *mPasta_Cap*
- Number of deli chefs – *mDeli_Chef*
- Capacity at the deli station (this is the number sandwiches that can be fit onto the oven trays) – *mDeli_OvenSpots*
- Number of Taqueria chefs (just the number of chefs that take the order) – *mTaco_Chef*
- Capacity at the Tacqueria station (this is the number of spots available on the Tacqueria grill) – *mTaco_GrillingSpots*
- Number of fryer chefs – *mFryer_Chef*
- Number of grill chefs (just the number of chefs that take the order) – *mGrill_Chef*
- Capacity at the grill (number of food items that can be managed on the grill at once) – *mGrillingSpots*
- Number of salad chefs – *mSalad_Chef*
- Capacity at the salad station (this should match the number of chefs) – *mSalad_Cap*
- Number of cashiers – *m_Cashiers*

MedModel allows for such macros to be built by defining the variable, assigning it a default value (the value of the system with no modifications), and then selecting defining its run-time interface (RTI). When the RTI is defined, a valid range of user inputs was given for each variable. Screen shots of the macro window and the RTI dialog box are provided below in Figure 20 and Figure 21 respectively.

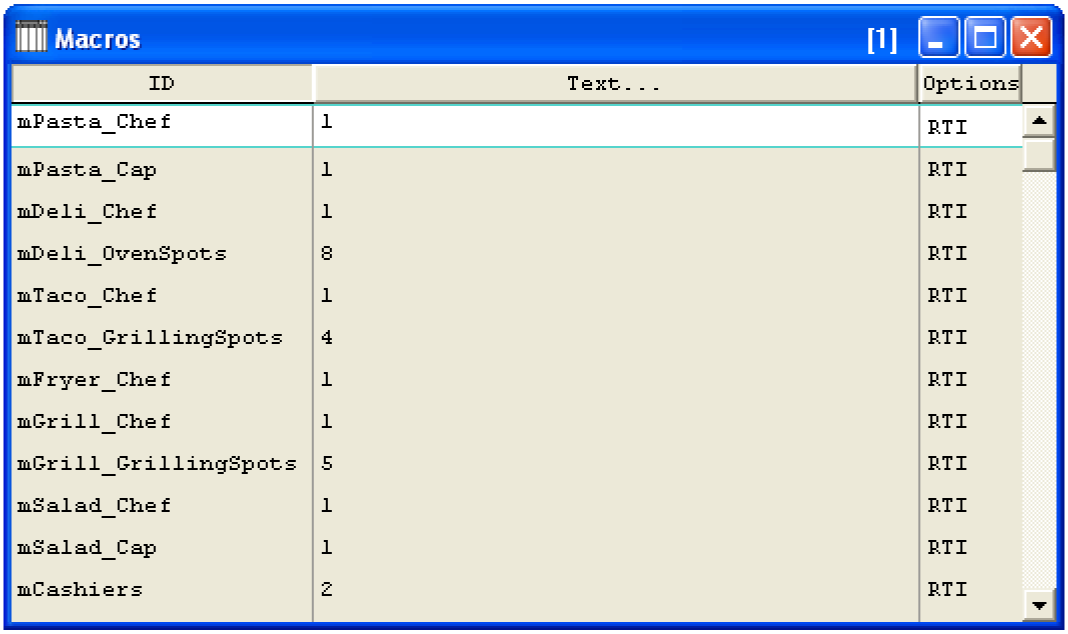


Figure 20: Macro Definition Window

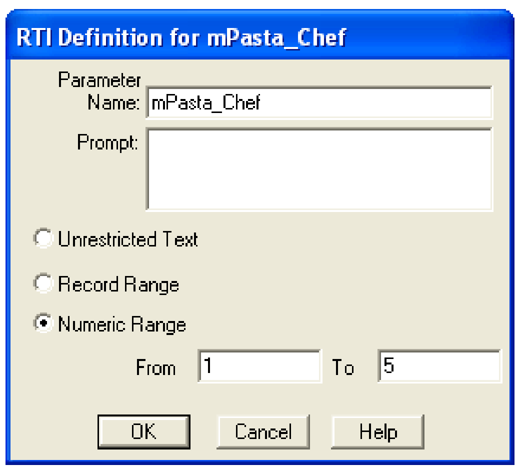


Figure 21: RTI Definition Window

The variable names were then used in lieu of the actual parameter values when defining the capacity for locations and the number of resources in their respective windows. A screen shot of the location window is given below in Figure 22 as an example.

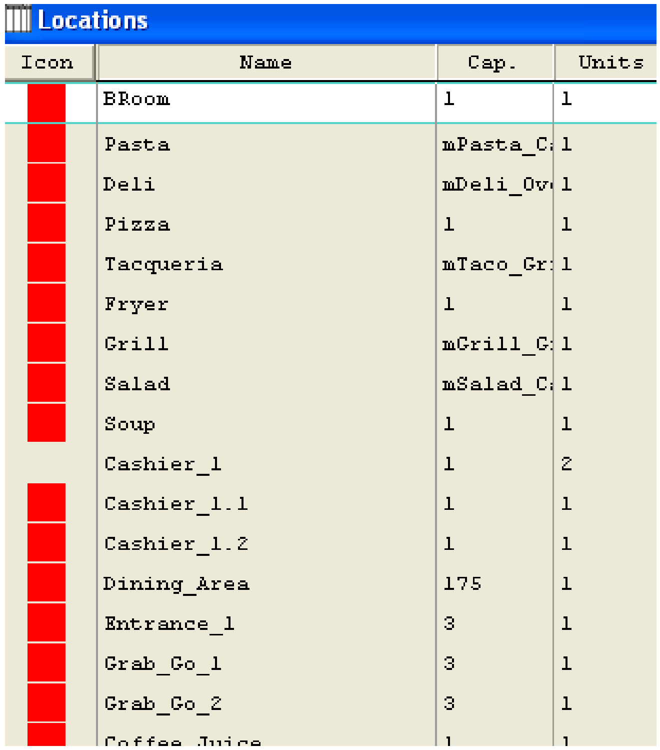


Figure 22: Location Definition Window with Variable Capacities

In this way, before the user runs the simulation in attempt to determine which changes should be to the system to improve performance, he is able to adjust the model parameters. Below in Figure 23 is the window that prompts the user to make such changes. The user is able to select a parameter to change, give it a value within the allowed range, and run the model with that change.

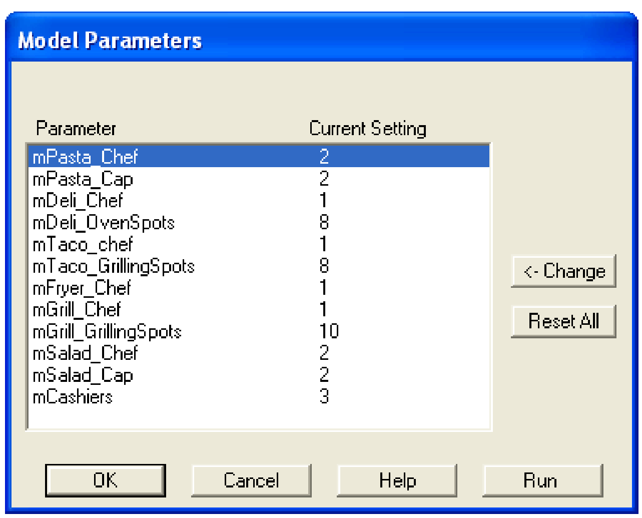


Figure 23: Interface for Adjusting Model Parameters

Results and Modification Analysis

Conducted by: Miss Davis, Miss Pitt, Miss Webelhuth
Written by: Miss Webelhuth

After the Fact: "Hidden" Assumptions

During the building of the model and the interpretation of the results, we realized that there were several assumptions that we were making that were previously unanticipated or unrealized. Some of these "hidden" assumptions include:

- Only people who bought something at Fox's Hole sit in the dining area – After observing the unusually low utilization numbers for the dining area, we realized that we only sent people who actually bought something from Fox's Hole over to the dining area. However realistically, this is not the case. People meet their friends there, people buy food from a nearby mini-supermarket and eat their purchases there, and some people just seem to show up there with no apparent reason. Therefore, by not including these other diners, the model shows a far lower utilization of the dining area than what is realistic.
- Station demand is independent of time of day and length of line – During the building of the model, we realized that we were assuming that the distribution of the station demands remains constant. This, however, does not take into account the more dynamic demands that fluctuate over time. When lines are long at one station, people who originally wanted that type of food might decide that they would rather stand in a shorter line and get a different type of food. Also, as the evening progresses, people seem to demand different types of food. For example, people seem to get salads more often earlier in the evening rather than later on.
- Arrival rate depends more on variables like weather, as opposed to day of week. Originally, we had anticipated that each day of the week would have a unique arrival rate distribution; however, after interpretation of arrival rate data, we concluded this to be false. Instead, things such as weather, big evening exams, or special events were more influential than day of the week. We then had to make the assumption that each day of the week had the same arrival distributions and the daily arrivals varied due to running the model several iterations with different seed values.

Metrics: Defining what means "Good" and "Bad"

After the program is run, the results need to be analyzed. Pages of numbers are only beneficial when you have a system of interpreting the results and judging the level of achievement. Metrics need to be determined that define "failure" and "success" of the system. Different groups of people have different needs and may want the system to accomplish different things. The goal of the cafeteria management may be to minimize cost while maintaining a level of acceptable service. The students, on the other hand, may only be interested in minimizing wait time in the cafeteria while making sure prices don't get out of control. Thus, defining a metric brings an aspect of subjectivity into the model. The goal of our project was to increase the efficiency of the cafeteria while attempting to reduce overall cost for the cafeteria management. Therefore, we defined the following metrics:

- *Utilization of Stations* – If a food station has a particularly high utilization, the possibility of having long wait-times increases. High utilization may indicate a bottleneck. At the same time, particularly low utilization levels may indicate that a resource could have the possibility of being utilized in a more effective manner, or eliminated altogether.
- *Average Time Spent at Station* – This metric is the sum of two parts, the time spent waiting in the queue and the service time. While service times are difficult to improve (who wants an undercooked hamburger), the time spent waiting in the queue can often be improved. The longer a customer has to wait in line to be served, the unhappier they become. Increasing the efficiency of the system to reduce time spent in the queue, and thus at the station, will lead to happier, more satisfied customers.
- *Percentage of Failed Arrivals* – The percentage of failed arrivals is calculated by dividing the number of failed arrivals by the number of successful arrivals. This percentage indicates a certain level of stability of the system. If the system cannot handle another customer, that customer "balks" and is considered a failed arrival, meaning they wanted to enter the system but the line at their destination station was too long and could not accommodate them. By reducing the percentage of failed arrivals, you assure that the cafeteria can accommodate most of the people who want to eat there.

Using these metrics, we defined our overall goal to be to reduce wait time, reduce extreme station utilizations, and reduce failed arrivals, all while keeping costs in mind. Now that we have the tools to interpret the data, we can use our metrics to get meaningful results.

Interpreting the Original System

<u>Scenario 1.0: Initial Results</u>

The model was first run in its original state before any modifications were made to the model. This run would serve as a sort of baseline control to which models with modifications could be compared, thus making the results of any change evident. The model was run for 10 iterations representing 10 separate days' worth of data. Then the Utilization, Average Queue Time, Average Service Time, and Total Average Station Time (see Figure 24, 25, 26, and 27 respectively) were graphed for all the stations with service times that have statistical distributions. The graph of percent utilization gives a quick visualization of which stations might be problem areas where long wait times are an issue. This wait time, as mentioned earlier, is the sum of the wait in the queue and the service time. Therefore, we then looked at both the wait in the queue and the service time to see where the problem was arising. Was it just the service time? - that cannot be changed easily. Or was it the time spent in the queue? - a metric which management could possibly implement a change to improve.

As seen in Figure 28, both Pasta and the Grill have utilizations that are hovering around 80%, and are therefore likely candidates to have the longest wait times. Our suspicions are confirmed when looking at the total station times. Because Pasta's long total station time is mainly due to the long wait in line, this is where our first improvement will be made. An improvement to the pasta station will also hopefully improve the percent of failed arrivals, which at times gets above 15%.

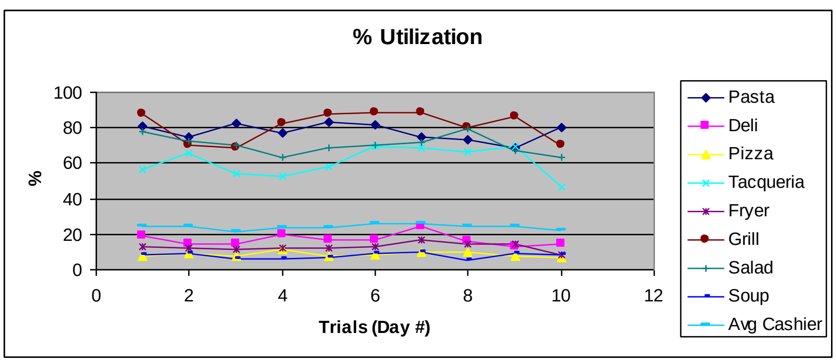


Figure 24: Utilization of Fox's Hole Stations

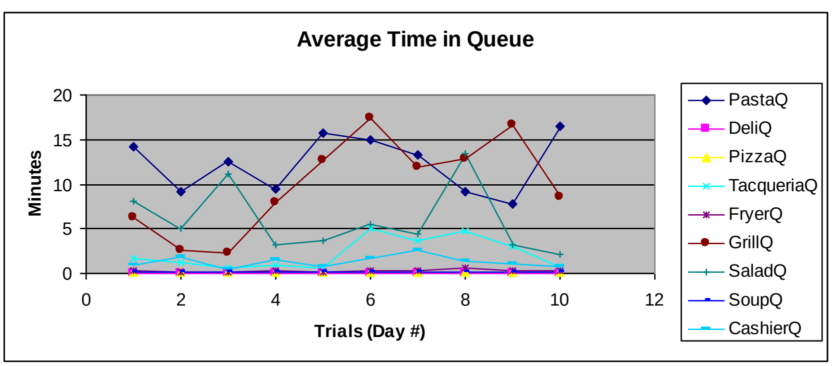


Figure 25: Average Time Waiting at Fox's Hole Stations

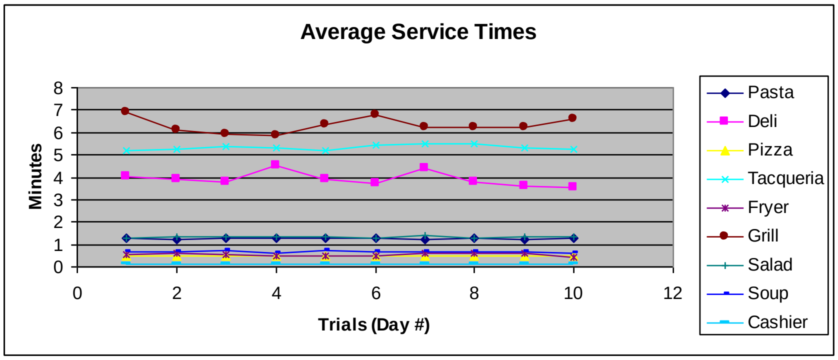


Figure 26: Average Service Times at Fox's Hole Stations

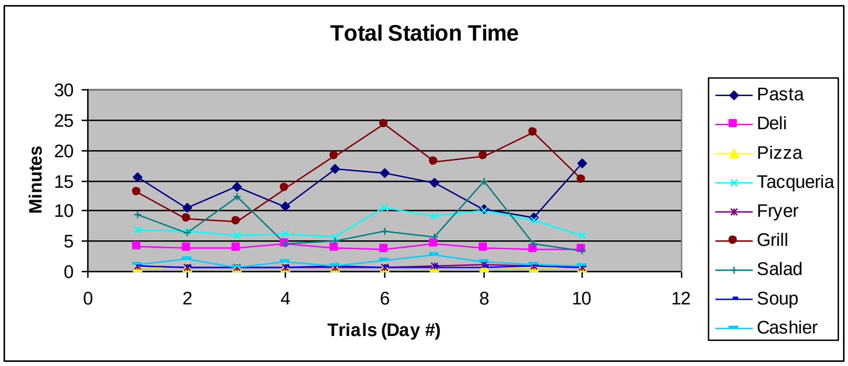


Figure 27: Total Time Spent by Customers at Fox's Hole Stations

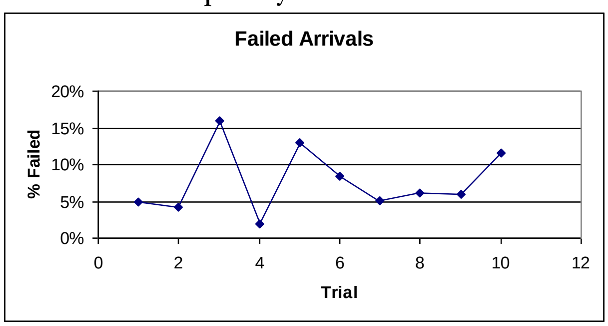


Figure 28: Failed Arrivals to Fox's Hole

Scenario 1.1: Additional Pasta Chef

Next we ran the model after adding a second resource (chef) at the Pasta station, allowing two people to be served simultaneously. This was done in hopes of reducing the wait time in the queue and also to reduce the number of failed arrivals. Looking at the results shown in Figures 30 and 33, we see that both the time spent waiting for pasta and the percentage of failed arrivals improved significantly. To decide upon our next place for improvement, we look again at the original station times (Figure 25). Grill is also very high so we target it next.

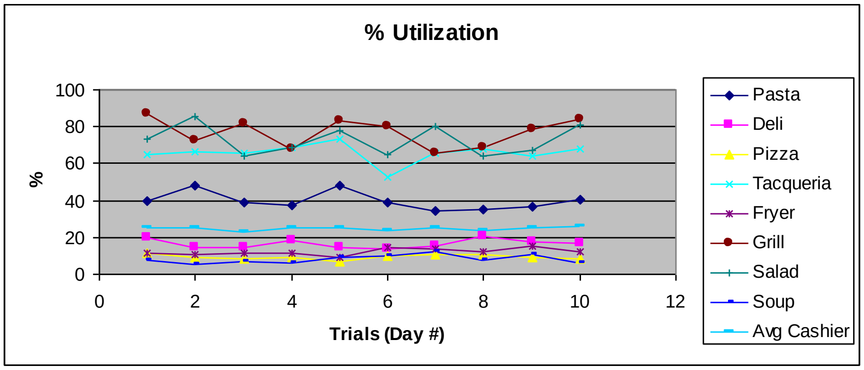


Figure 29: Scenario 1.1 Utilization of Fox's Hole Stations

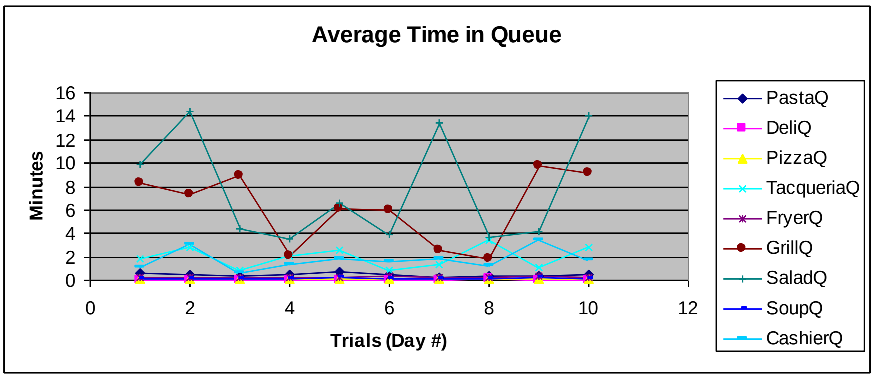


Figure 30: Scenario 1.1 Average Time Waiting at Fox's Hole Stations

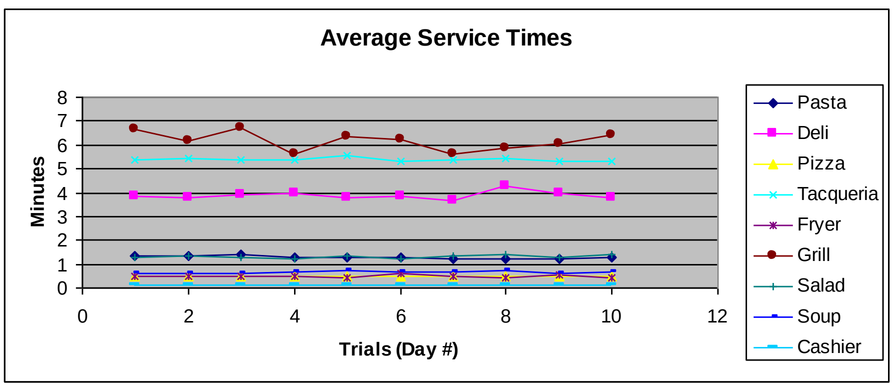


Figure 31: Scenario 1.1 Average Service Times at Fox's Hole Stations

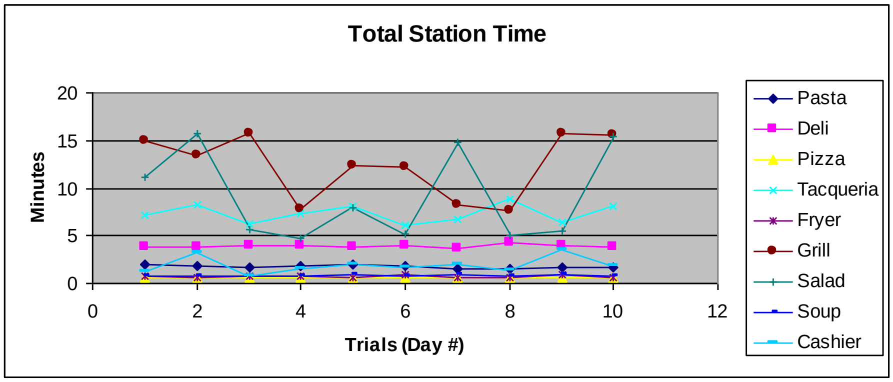


Figure 32: Scenario 1.1 Total Time Spent by Customers at Fox's Hole Stations

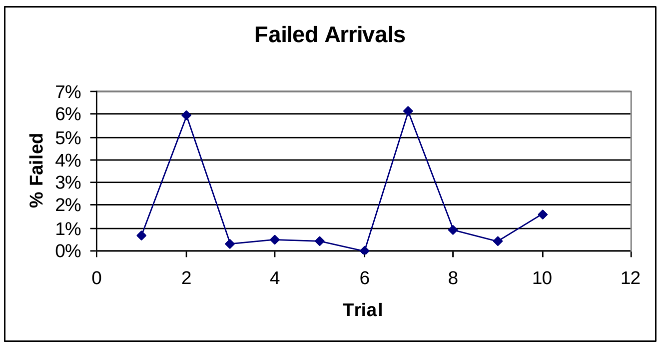


Figure 33: Scenario 1.1 Failed Arrivals to Fox's Hole

<u>Scenario 1.2: Additional Grill Space</u>

Unlike at the Pasta station where adding an additional chef would improve the wait time, the Grill needs something else. Most of the service time is spent waiting for the customer's food to cook, and while waiting, these customers are taking up one of the five location capacity spots. The chef is not being overworked, he just needs to wait for there to be room on the grill before serving the next customer. Therefore, to reduce wait time, another grill could be installed, improving the service capacity at that location. To simulate this addition, we changed the capacity at the Grill station from 5 spots to 10 spots, thus giving the grill the opportunity to serve more customers at a time. Remembering that we set the Pasta station back to one resource, examining the results below in Figures 34 through 38 shows that an independent Grill change is not as effective as the Pasta change. Even though the wait in the queue for the Grill is down to almost nothing, the percent of failed arrivals did not improve as much as when we added a Pasta resource.

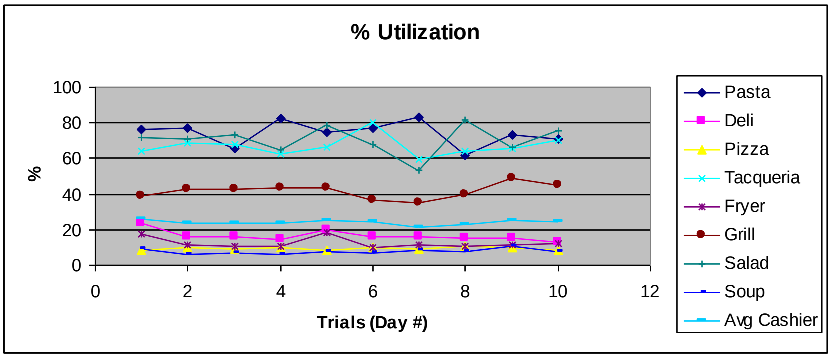


Figure 34: Scenario 1.2 Utilization of Fox's Hole Stations

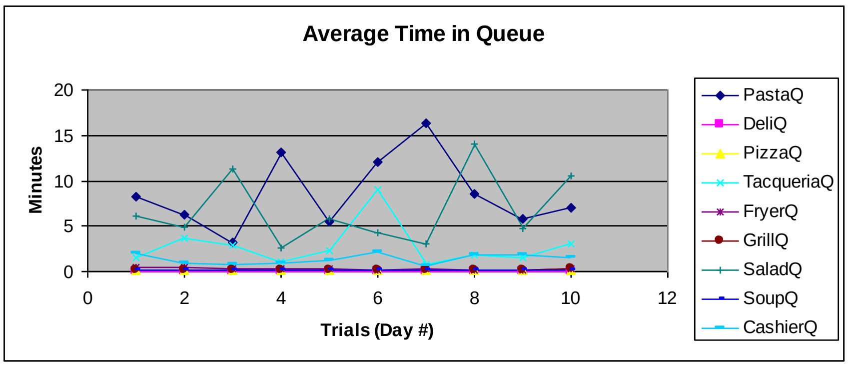


Figure 35: Scenario 1.2 Average Time Waiting at Fox's Hole Stations

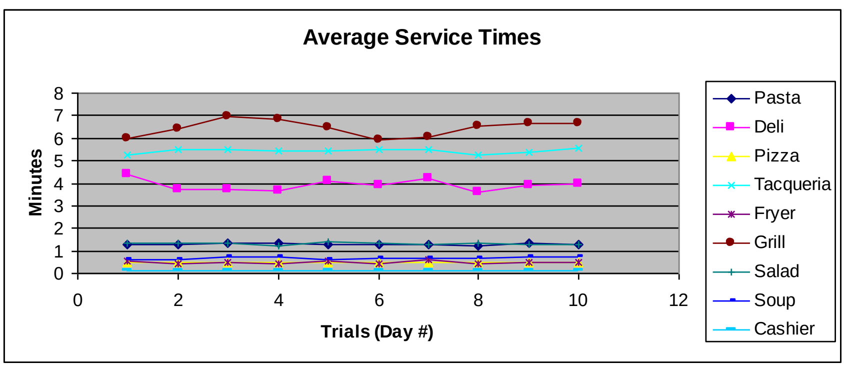


Figure 36: Scenario 1.2 Average Service Times at Fox's Hole Stations

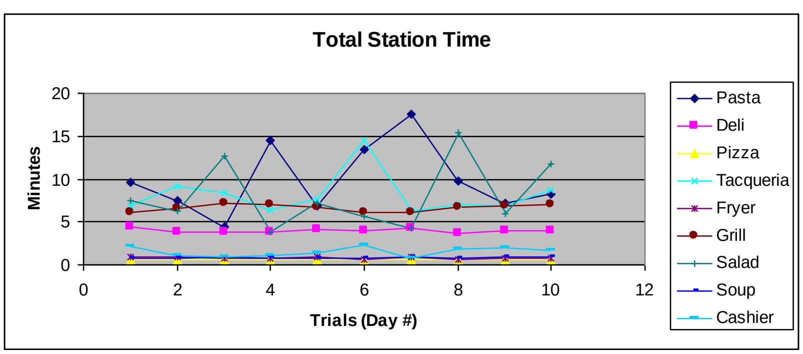


Figure 37: Scenario 1.2 Total Time Spent by Customers at Fox's Hole Stations

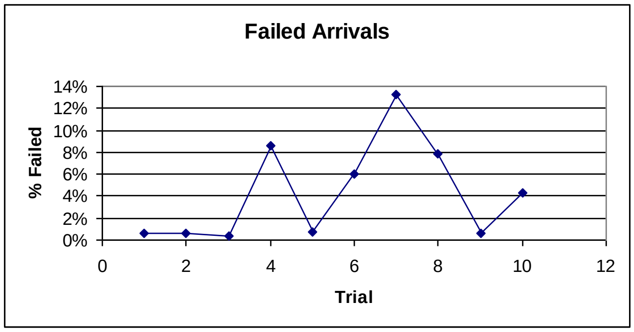


Figure 38: Scenario 1.2 Failed Arrivals to Fox's Hole

Scenario 1.3: Additional Salad Chef

The next independent change we made was adding an additional Salad resource (chef) to see if this change is more effective than an independent Pasta or Grill change. The below results in Figures 39 through 43 tell us that adding an addition Salad chef dramatically reduces the time spend waiting in the salad line, but overall, it does not have a big impact on reducing the percentage of failed arrivals. So we can conclude that a significant amount of the failed arrivals had intended to get pasta.

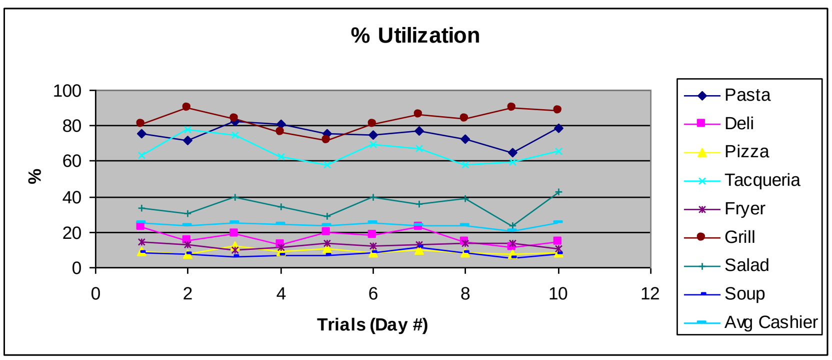


Figure 39: Scenario 1.3 Utilization of Fox's Hole Stations

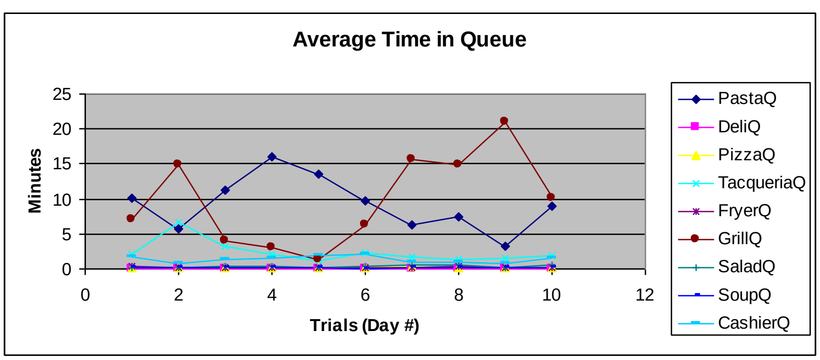


Figure 40: Scenario 1.3 Average Time Waiting at Fox's Hole Stations

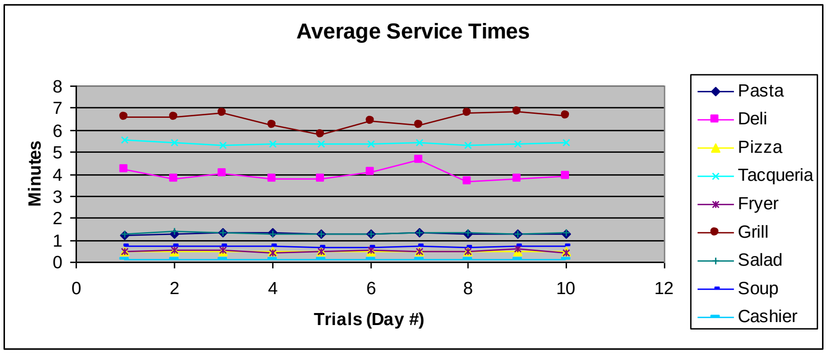


Figure 41: Scenario 1.3 Average Service Times at Fox's Hole Stations

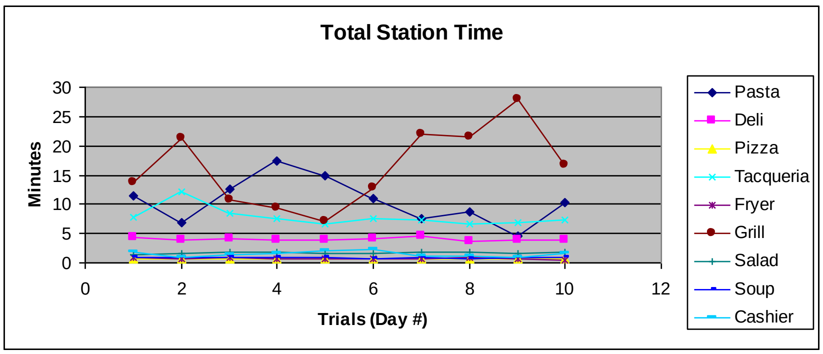


Figure 42: Scenario 1.3 Total Time Spent by Customers at Fox's Hole Stations

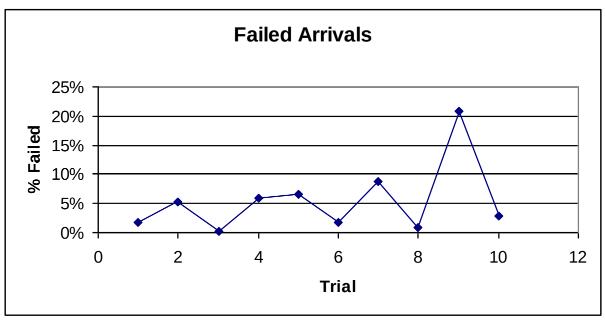


Figure 43: Scenario 1.3 Failed Arrivals to Fox's Hole

<u>Scenario 1.4: Additional Pasta and Salad Chefs, Grill Space</u>

Since the three changes we made so far were effective at reducing wait time independently, we next wanted to see how things change if they were enacted together. Therefore, we added an additional Pasta chef and Salad chef and increase the number of Grill spots from 5 to 10. From Figures 44 through 48 below, we see that this is a successful change, reducing the utilization of these three stations to less than 50%, bringing the wait time in the queue to insignificant levels, and greatly reducing the percent of failed arrivals. However, now we begin to see the arise of a different problem. Because more people are entering the system due to the low failed arrival levels, the wait time at the Cashier is now becoming a problem. This increased wait time is not only due to the higher number of customers who successfully enter the system, but is also due to the fact that since the wait time at some stations is reduced, there is less of a buffer when there is a high arrival rate. What this means is that often there is a period of time when a large number of people enter the system, but because they get stuck waiting in the station lines, they come to the Cashier in a more staggered fashion. However, since now the wait times at some stations are greatly reduced, there is less of a "filter" for the periodss with high arrival rates.

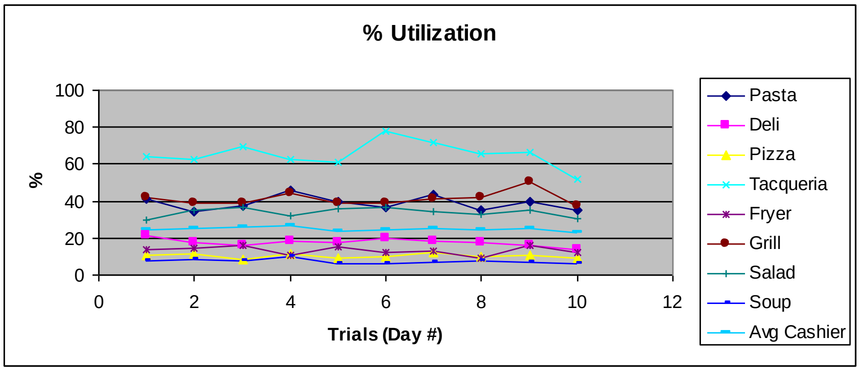


Figure 44: Scenario 1.4 Utilization of Fox's Hole Stations

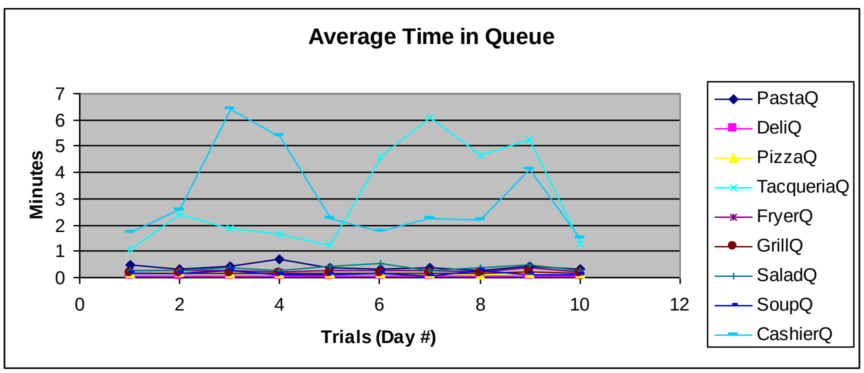


Figure 45: Scenario 1.4 Average Time Waiting at Fox's Hole Stations

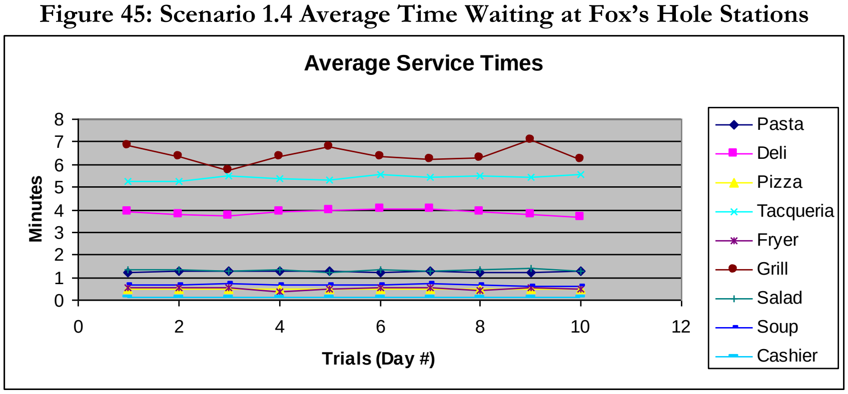


Figure 46: Scenario 1.4 Average Service Times at Fox's Hole Stations

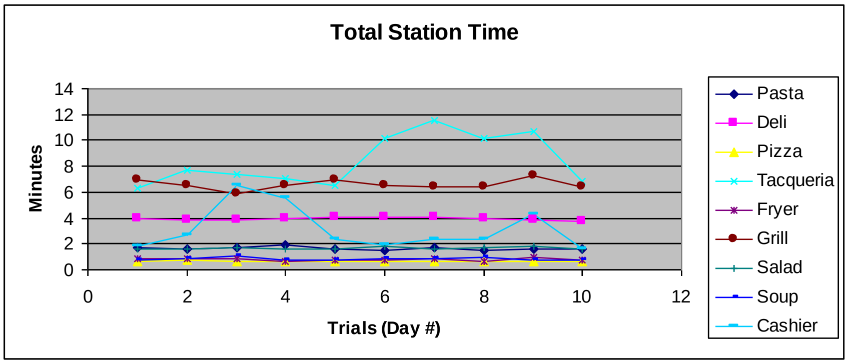


Figure 47: Scenario 1.4 Total Time Spent by Customers at Fox's Hole Stations

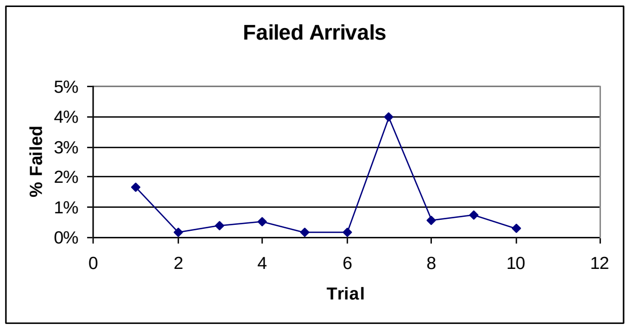


Figure 48: Scenario 1.4 Failed Arrivals to Fox's Hole

<u>Scenario 1.5: Additional Pasta and Salad Chefs, Grill and Taco Spaces, Cashier</u>

Building off the previous three changes, we then targeted the remaining problem areas. The Tacqueria station had a problem similar to the Grill station. Because there is limited room on the Tacqueria griddle, people waiting for their food to cook were taking up the station capacity and the chef could not begin serving the next customers in line. Therefore, by adding an additional griddle and doubling the capacity at that station from 4 to 8, we could reduce the time spent waiting in the queue. On the other hand, the Cashier problem is more similar to the Pasta and Salad problem. The addition of another resource (cashier) would thus be an effective strategy for reducing the time spent waiting to get checked-out. Figures 49 through 53 below show that the combination of these five changes has an astonishing result. The wait-times at all of the stations is now below 1 minute, making the graph of the total station time an almost exact replica of the graph of service times. Also, failed arrivals is now around 1%, meaning that Fox's Hole now has the ability to serve nearly everyone who wants to come.

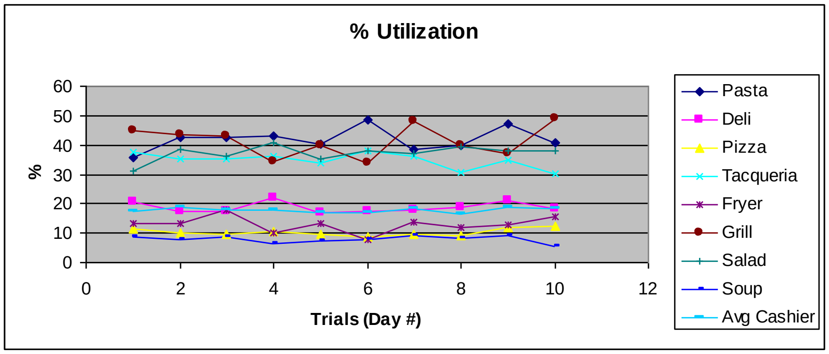


Figure 49: Scenario 1.5 Utilization of Fox's Hole Stations

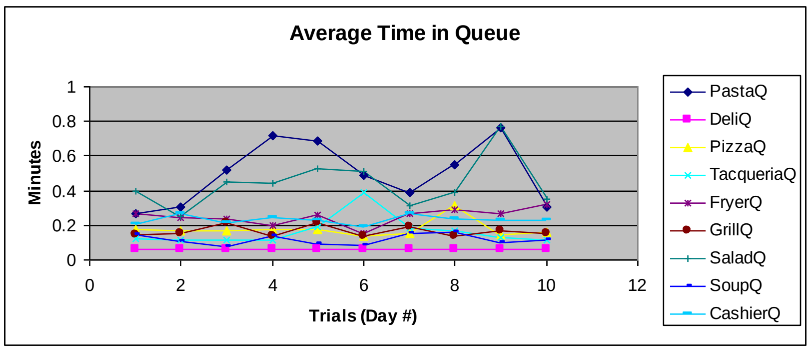


Figure 50: Scenario 1.5 Average Time Waiting at Fox's Hole Stations

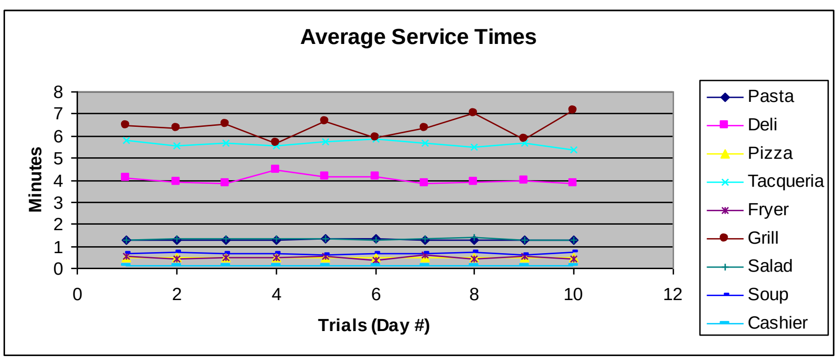


Figure 51: Scenario 1.5 Average Service Times at Fox's Hole Stations

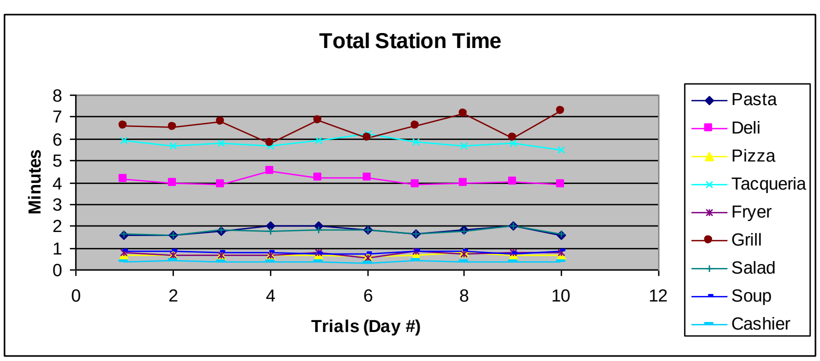


Figure 52: Scenario 1.5 Total Time Spent by Customers at Fox's Hole Stations

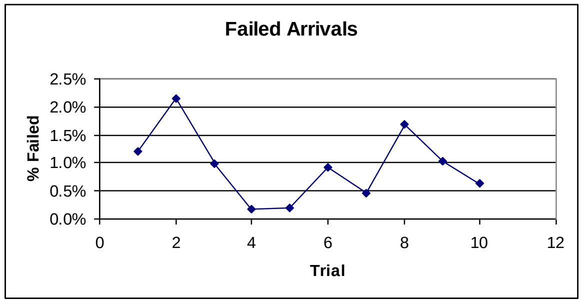


Figure 53: Scenario 1.5 Failed Arrivals to Fox's Hole

Median Market Analysis

<u>Scenario 2.0: The Closing of Median Market</u>

The goal of this project is to not only improve the efficiency of Fox's Hole, but also to save the company money. We had suggested eliminating the wasteful Median Market to significantly reduce costs, and now we have the ability to test whether Fox's Hole is equipped to handle this closure. In this scenario, we are assuming that everyone who would have eaten at Median Market will now choose to eat at Fox's Hole. We realize that this may be an extreme assumption to make, but we want to make sure that Fox's Hole could handle the worst case scenario. The following results came from running the model with the original arrival distributions plus the Median Market arrival distributions. As you can see from the result below in Figures 54 through 58, if Median Market were to be shut down and no alterations made to Fox's Hole, the system would become unstable. Not only are the wait times at near epic levels, but failed arrivals are around 20%. This is unacceptable. So how do we fix this situation?

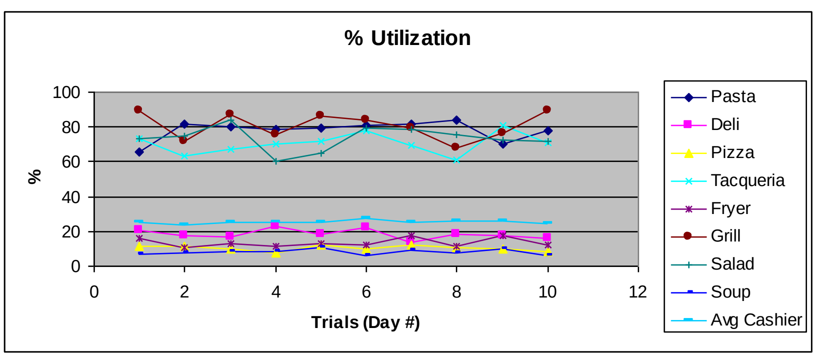


Figure 54: Utilization of Fox's Hole (Including Median Market) Stations

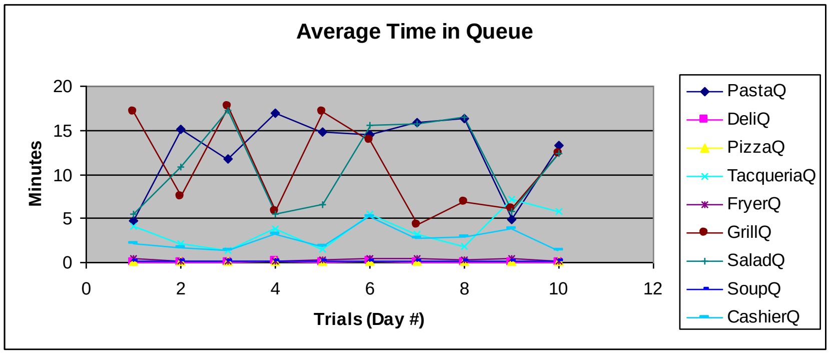


Figure 55: Average Time Waiting at Fox's Hole (Including Median Market) Stations

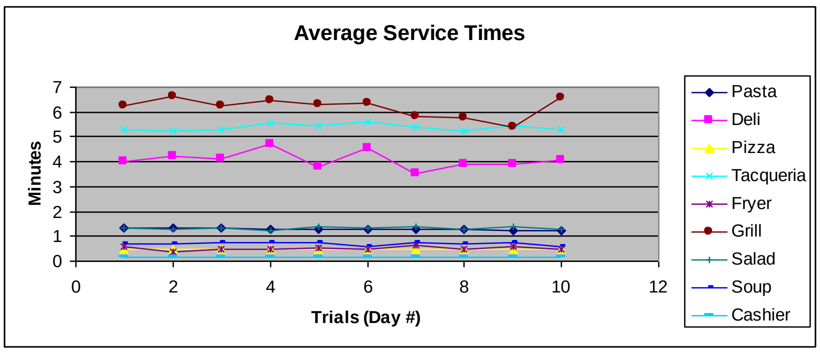


Figure 56: Average Service Times at Fox's Hole (Including Median Market) Stations

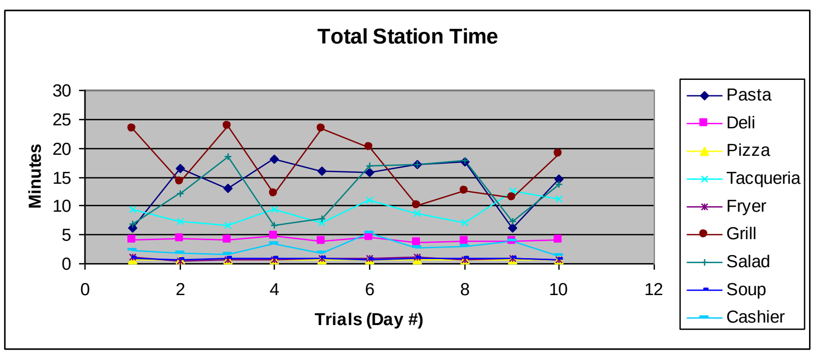


Figure 57: Total Time Spent by Customers at Fox's Hole (Including Median Market) Stations

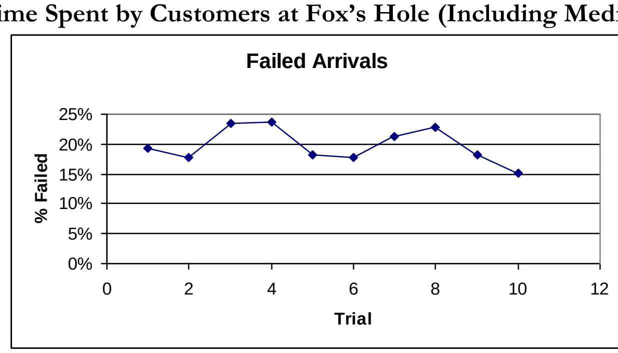


Figure 58: Failed Arrivals to Fox's Hole (Including Median Market)

Scenario 2.1: Additional Pasta Chef and Grill Space

Using the lessons learned from optimizing the original Fox's Hole model, we first worked on the problematic pasta and grill stations. We added a pasta resource (chef) and added another grill, doubling the capacity of the grill station from 5 to 10 spots. After running this improved model, the below results in Figures 59 through 63 indicate that Fox's Hole can now handle more people since the failed arrivals were reduced from 20% to around 15%. However, this is still too high of a level to be acceptable, so something else needed to be done.

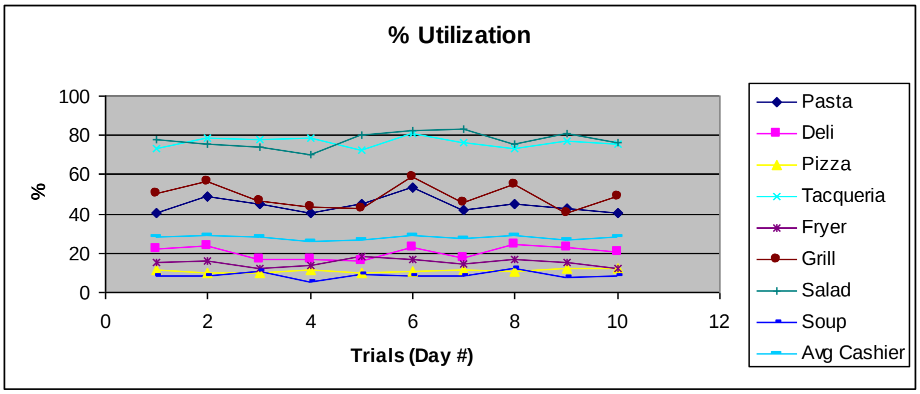


Figure 59: Scenario 2.1 Utilization of Fox's Hole (Including Median Market) Stations

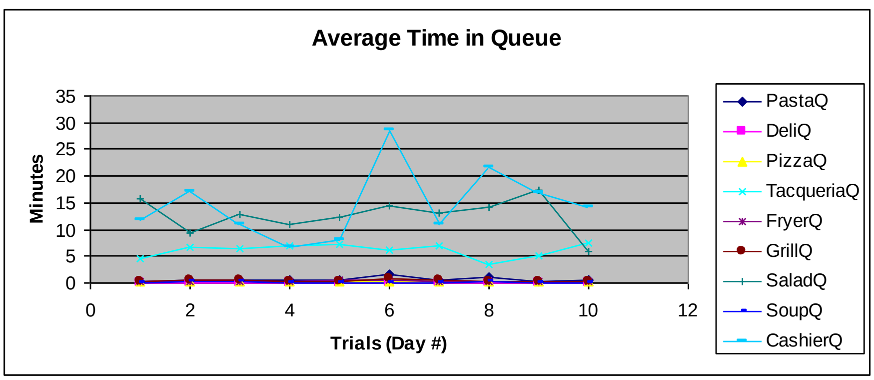


Figure 60: Scenario 2.1 Average Time Waiting at Fox's Hole (Including Median Market) Stations

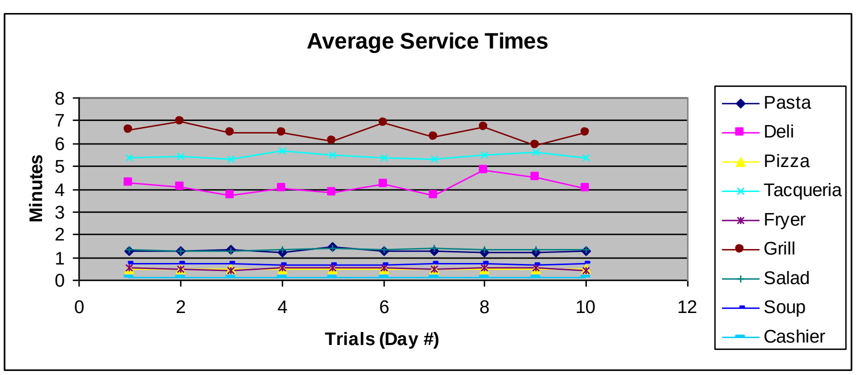


Figure 61: Scenario 2.1 Average Service Times at Fox's Hole (Including Median Market) Stations

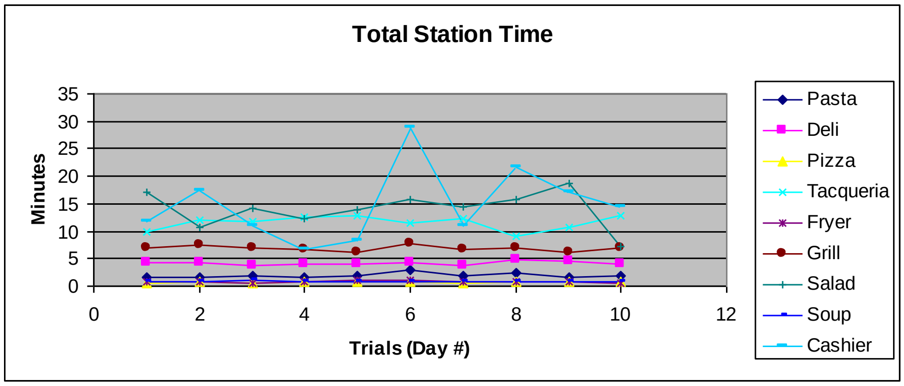


Figure 62: Scenario 2.1 Total Time Spent by Customers at Fox's Hole (Including Median Market) Stations

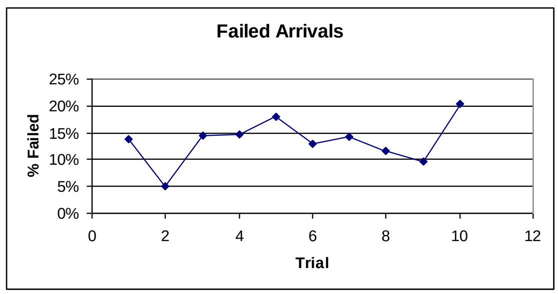


Figure 63: Scenario 2.1 Failed Arrivals to Fox's Hole (Including Median Market)

<u>Scenario 2.2: Additional Pasta and Salad Chefs, Grill Space</u>

The next step was to reduce the notorious salad line by adding another resource (chef). The results below in Figures 64 through 68 indicate that this change not only reduced the wait time in the queue for salad, but it also decreased the number of failed arrival from around 15% to a much better average of 7-8%. But looking at Figure 65, we see that another problem was worsened, the queue time for the cashiers is now out of control!

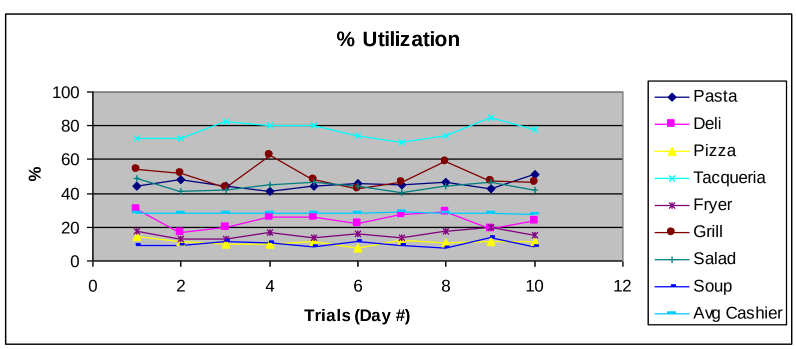


Figure 64: Scenario 2.2 Utilization of Fox's Hole (Including Median Market) Stations

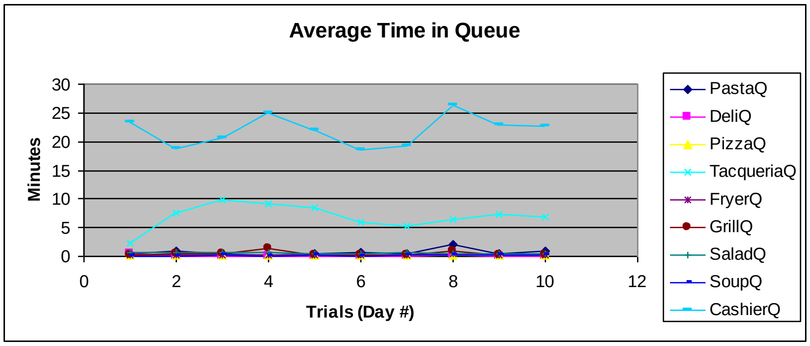


Figure 65: Scenario 2.2 Average Time Waiting at Fox's Hole (Including Median Market) Stations

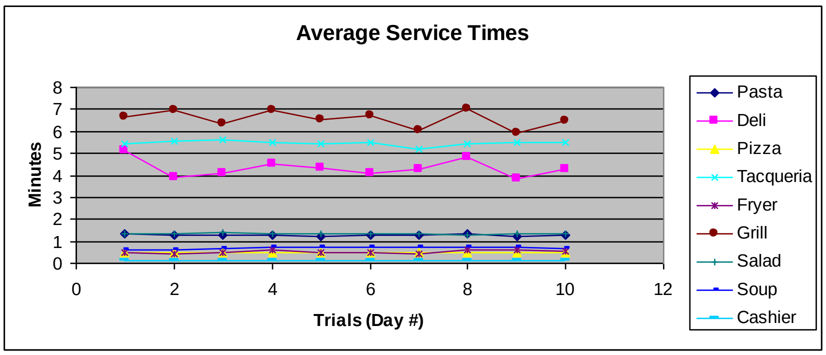


Figure 66: Scenario 2.2 Average Service Times at Fox's Hole (Including Median Market) Stations

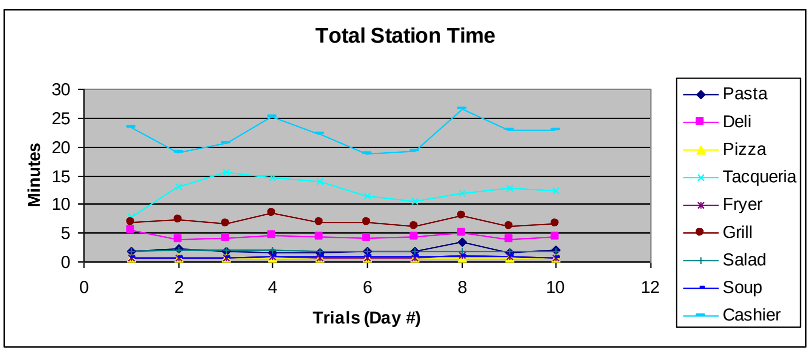


Figure 67: Scenario 2.2 Total Time Spent by Customers at Fox's Hole (Including Median Market) Stations

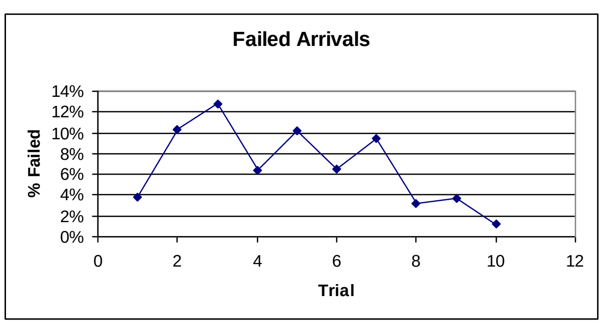


Figure 68: Scenario 2.2 Failed Arrivals to Fox's Hole (Including Median Market)

Scenario 2.3: Additional Pasta and Salad Chefs, Grill Space and Cashier

From the results of scenario 2.2, we realized that we were dealing with a so called "wandering bottleneck", meaning that a station that once was not a big problem now is after other stations have been improved. An average wait time of 20-25 minutes in the cashier queue is clearly a problem because customers will become quite upset if their food gets cold while they are waiting to get checked-out. Another cashier was added to the system and the model was run to produce the following results in Figures 69 through 73. In scenario 2.2, people were entering the system because the lines at the stations were not too long, but then they got stuck at the cashier. Therefore, when we added another cashier, this did not improve the percentage of failed arrival over last time because the cashier is a secondary process, but it did reduce the time customers had to spend in the queue. The only station now that seems to be a problem is the Tacqueria, so we move on to scenario 2.4.

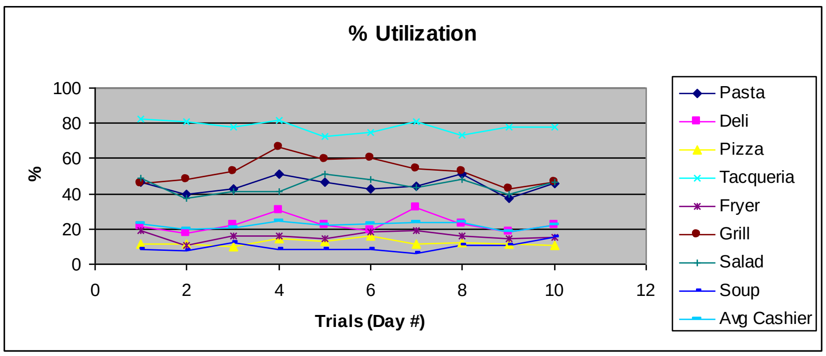


Figure 69: Scenario 2.3 Utilization of Fox's Hole (Including Median Market) Stations

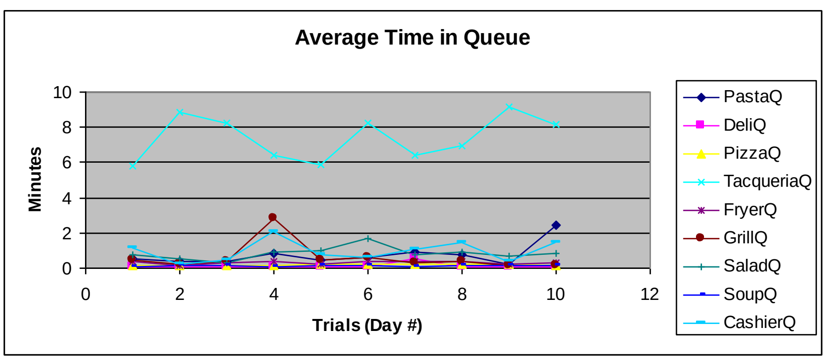


Figure 70: Scenario 2.3 Average Time Waiting at Fox's Hole (Including Median Market) Stations

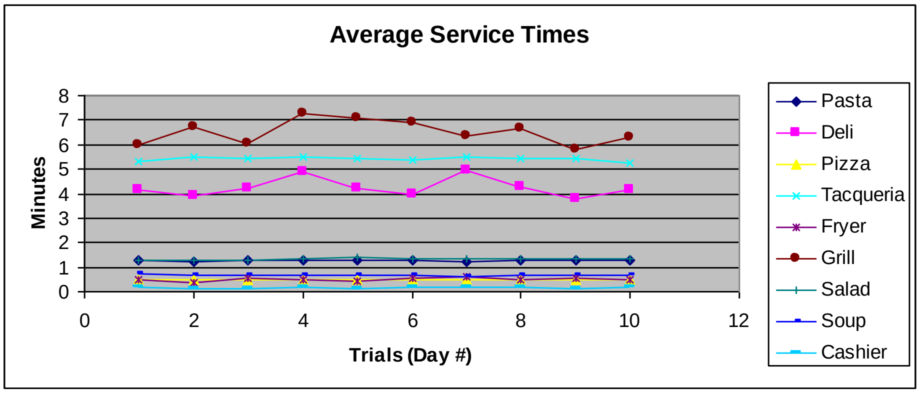


Figure 71: Scenario 2.3 Average Service Times at Fox's Hole Stations

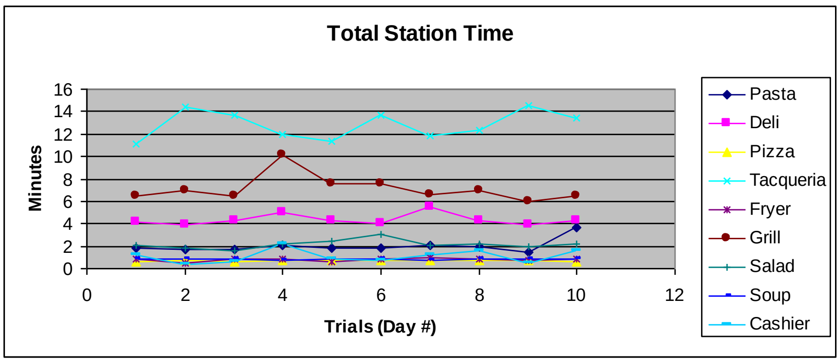


Figure 72: Scenario 2.3 Total Time Spent by Customers at Fox's Hole (Including Median Market) Stations

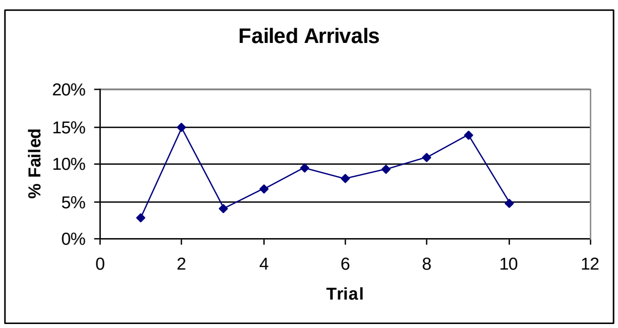


Figure 73: Scenario 2.3 Failed Arrivals to Fox's Hole (Including Median Market)

Scenario 2.4: Additional Pasta and Salad Chefs, Grill and Taco Spaces, Cashier

The Tacqueria station was improved the same way as before by adding an additional griddle to double the location capacity from 4 to 8 spots. From Figure 78 below, we see that failed arrivals have been reduced to an acceptable 4%, but from Figure 75, we notice another problem. The average queue time for the cashier has jumped up again, even though we now have 3 cashiers. This can be explained like it was in scenario 1.4; now that the wait-time in the queues of several stations has been reduced, there is less of a buffer when there are periods of high arrival rates. Therefore the cashiers have to deal with these rushes instead of having them be more spread out over time.

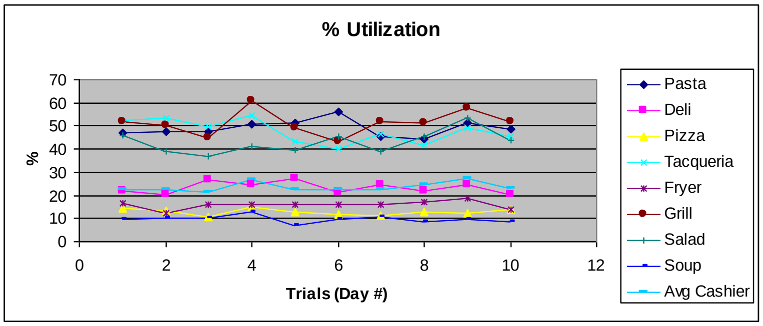


Figure 74: Scenario 2.4 Utilization of Fox's Hole (Including Median Market) Stations

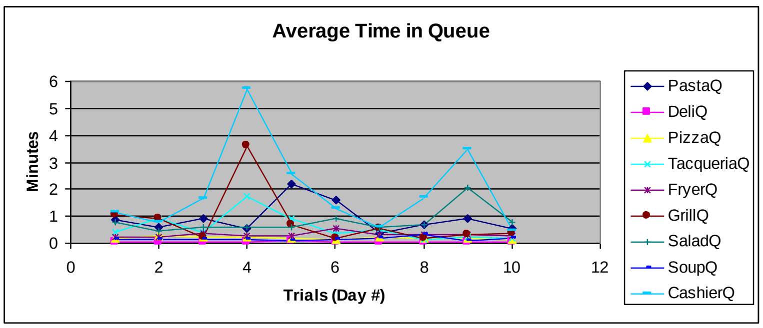


Figure 75: Scenario 2.4 Average Time Waiting at Fox's Hole (Including Median Market) Stations

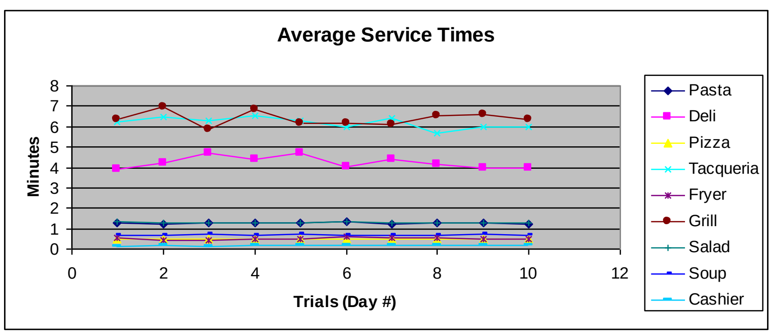


Figure 76: Scenario 2.4 Average Service Times at Fox's Hole Stations

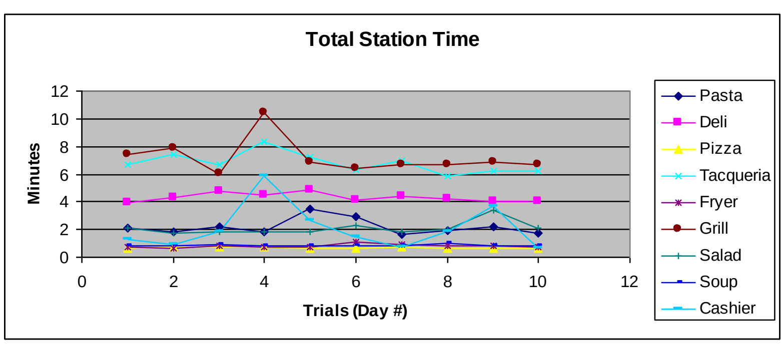


Figure 77: Scenario 2.4 Total Time Spent by Customers at Fox's Hole (Including Median Market) Stations

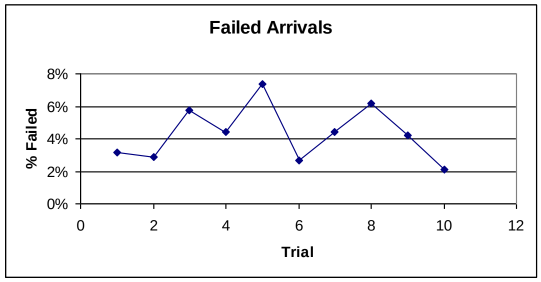


Figure 78: Scenario 2.4 Failed Arrivals to Fox's Hole (Including Median Market)

<u>Scenario 2.5: Additional Pasta and Salad Chefs, Grill and Taco Spaces, 2 Cashiers</u>

In this final scenario, the number of cashiers was increased to 4. Figure 80 shows that the average time spent in the queue for any given station is not significant enough to cause any major problems. After adding the fourth cashier, the failed arrivals stays around 4%, the level it was in scenario 2.4 because again, once people are in the system, the cashier is only a secondary station. With these changes, Fox's Hole is now equipped to handle the additional customers who normally would have eaten at Median Market. For more information, please reference Figures 79 through 83 below.

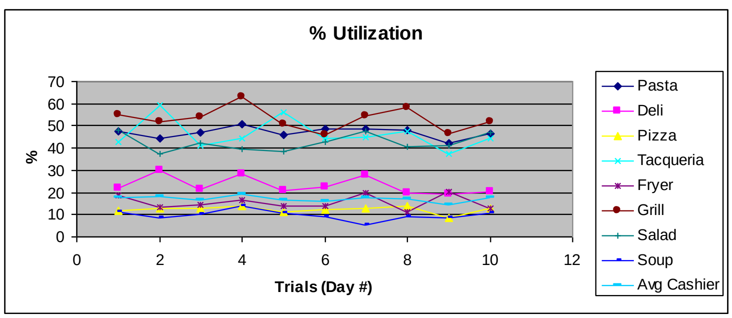


Figure 79: Scenario 2.5 Utilization of Fox's Hole (Including Median Market) Stations

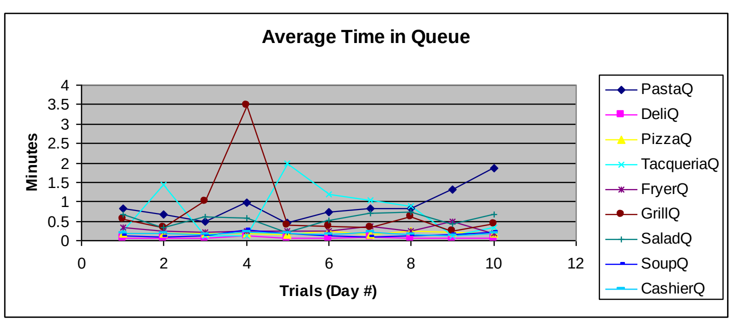


Figure 80: Scenario 2.5 Average Time Waiting at Fox's Hole (Including Median Market) Stations

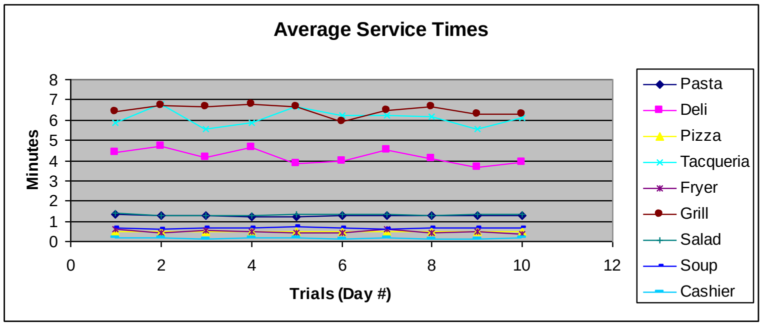


Figure 81: Scenario 2.5 Average Service Times at Fox's Hole Stations

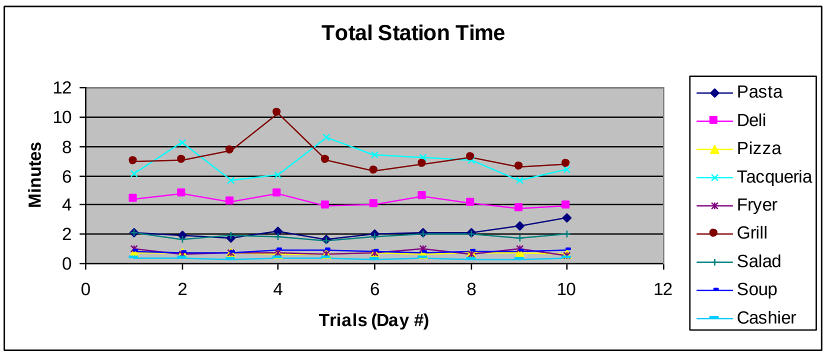


Figure 82: Scenario 2.5 Total Time Spent by Customers at Fox's Hole (Including Median Market) Stations

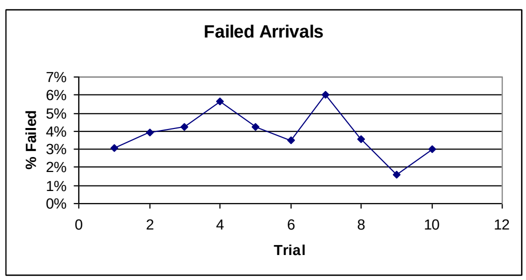


Figure 83: Scenario 2.5 Failed Arrivals to Fox's Hole (Including Median Market)

Conclusions

Written by: Miss Webelhuth

With our model, we were able to somewhat accurately see the results of changes to Fox's Hole without having to change the actual system. We were able to accomplish our goals of improving efficiency of Fox's Hole while at the same time reducing costs by closing Median Market. Even though we had to implement changes to make sure Fox's Hole was able to cope with the closure of Median Market, the cost of these improvements do not outweigh the savings generated by closing Median Market. Perhaps workers who normally are stationed at Median Market can be the ones who are added to Fox's Hole. There would be minimal, if any additional training needed and fewer jobs would be lost. Overall, the changes we recommend create a win-win situation for both management and students.

Appendix: Statistical Distributions

Conducted by: Miss Webelhuth
Written by: Miss Webelhuth

LogLogistic

The log-logistic distribution is often used in survival analysis and economics, and it is the probability distribution of a random variable whose logarithm has a logistic distribution. The log-logistic distribution was used to describe the service time at the Pasta, Deli, Fryer, and Dining stations in our model.

pdf: f(x) = $\dfrac{(\beta/\alpha)(x/\alpha)^{\beta-1}}{\left[1+(x/\alpha)^{\beta}\right]^{2}} \quad \alpha, \beta > 0$

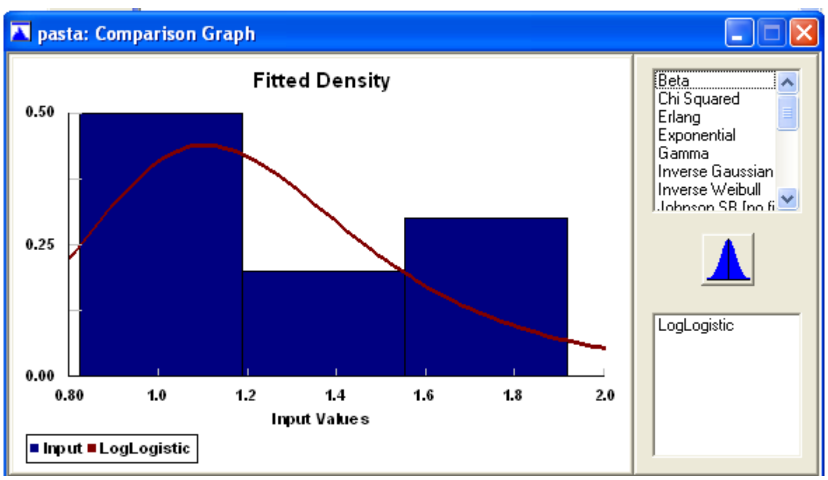


Figure 84: LogLogistic distribution fit to Pasta service times

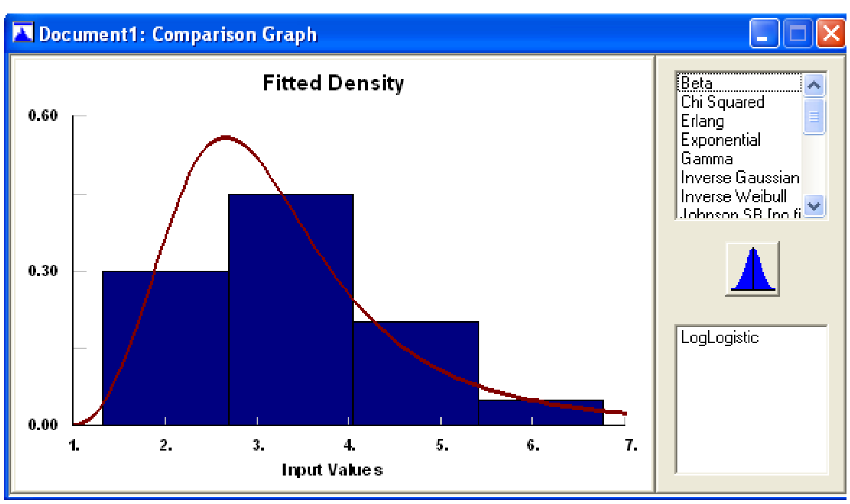


Figure 85: LogLogistic distribution fit to Deli service times

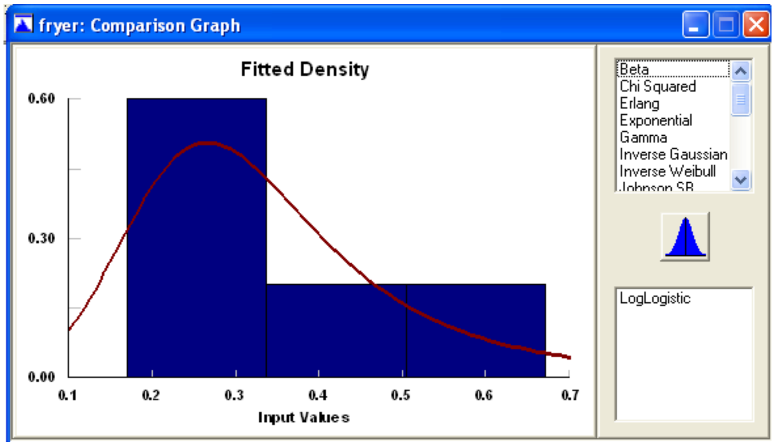


Figure 86: LogLogistic distribution fit to Fryer service times

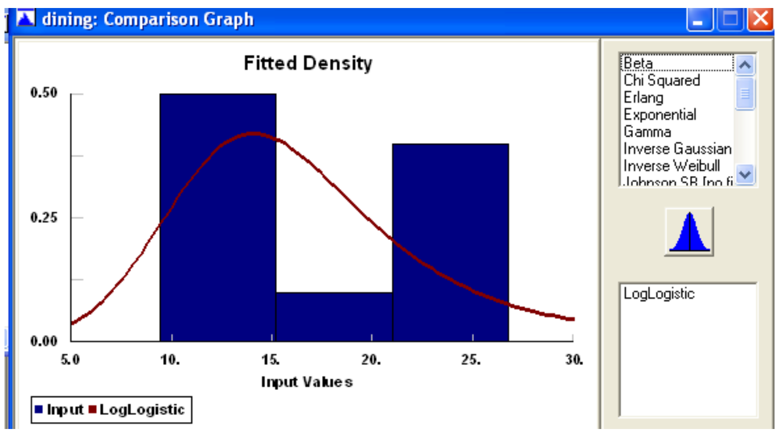


Figure 87: LogLogistic distribution fit to Dining times

Weibull

The weibull distribution is often used in the study of reliability, but in our model it is used to describe the Tacqueria and Grill stations.

pdf: f(x) = $(\beta/\alpha)(x/\alpha)^{(\beta-1)}e^{-(x/\alpha)^{\beta}} \qquad \alpha,\beta > 0$

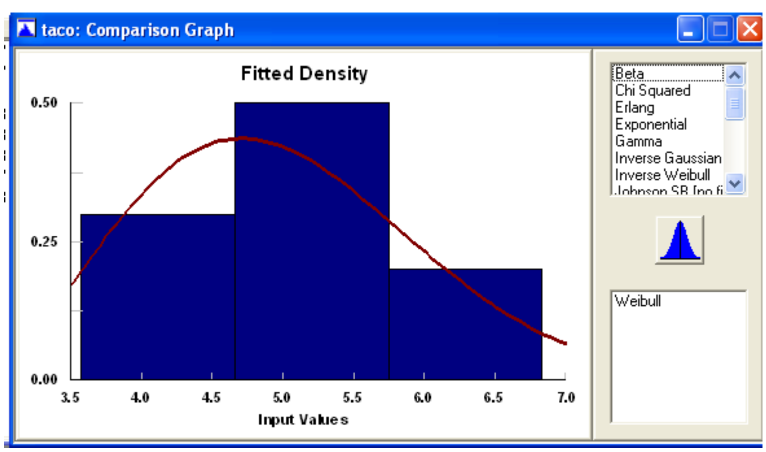


Figure 88: Weibull distribution fit to Tacqueria service times

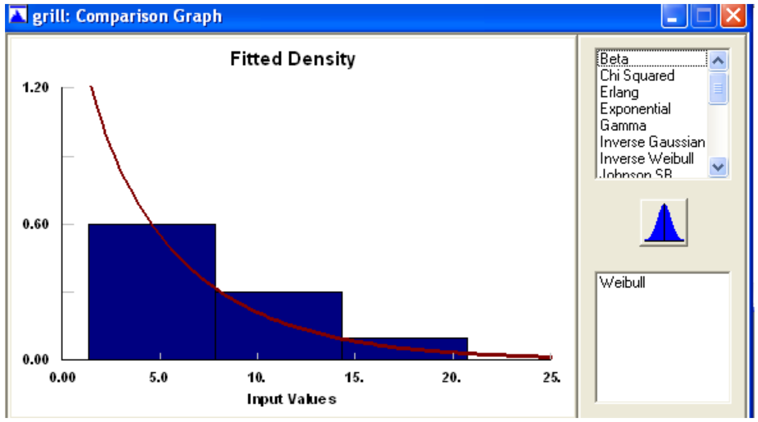


Figure 89: Weibull distribution fit to Grill service times

Inverse Weibull

The inverse weibull distribution was used to describe the Salad and Soda stations.

pdf: f(x) = $(\beta / \alpha)(\alpha / x)^{(\beta+1)} e^{-(\alpha / x)^{\beta}} \qquad \alpha, \beta > 0$

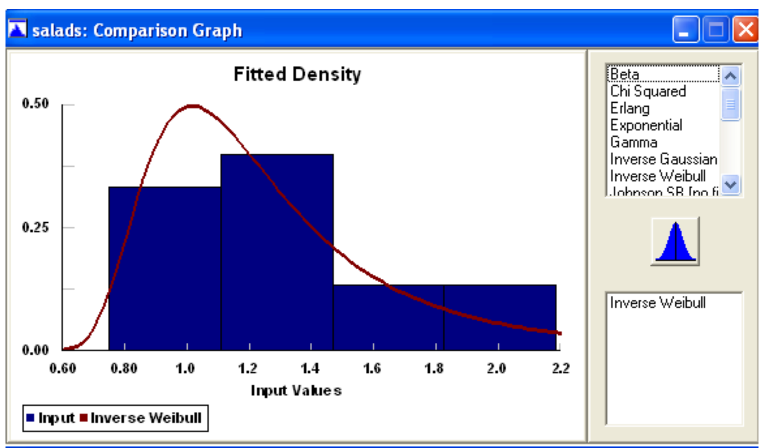


Figure 90: Inverse Weibull distribution fit to Salad service times

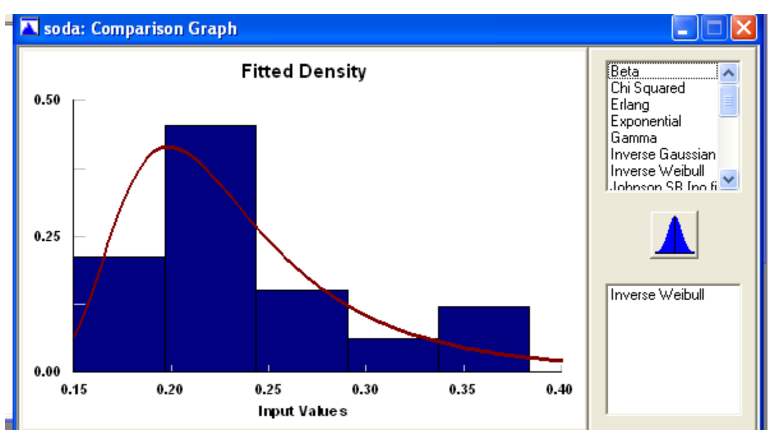


Figure 91: Inverse Weibull distribution fit to Soda service times

Gamma

The gamma distribution is an important distribution because it defines the exponential and chi-squared random variables. In our model, it defines the soup service time.

pdf: f(x) = $\frac{1}{\Gamma(\alpha)\beta^{\alpha}} x^{\alpha-1} e^{-x/\beta} \qquad \alpha, \beta > 0$

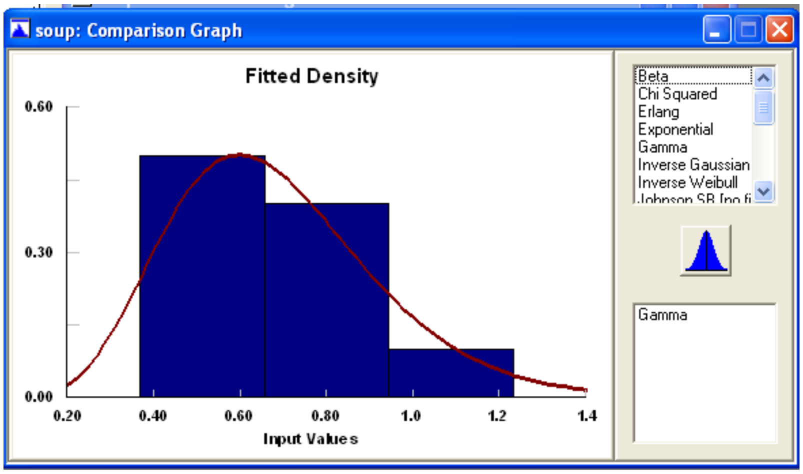


Figure 92: Gamma distribution fit to Soup service times

Pearson V

The Pearson V distribution is often used in biostatistics. We used it to describe the check-out time length at the cashiers.

pdf: f(x) = $\frac{\left(x^{(\alpha+1)} e^{-\beta/x}\right)}{\left(\beta^{-\alpha}\Gamma(\alpha)\right)} \qquad \alpha, \beta > 0$

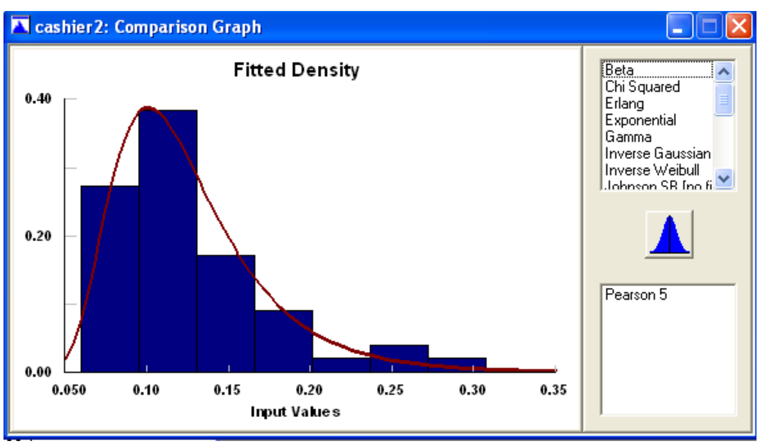


Figure 93: Pearson 5 distribution fit to Cashier service times

Beta

The beta distribution is widely used in Bayesian statistics and in the critical path method used for project management. We use the beta distribution in our model to describe the arrival of customers into our system.

pdf: f(x) = $\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \qquad \alpha, \beta > 0$

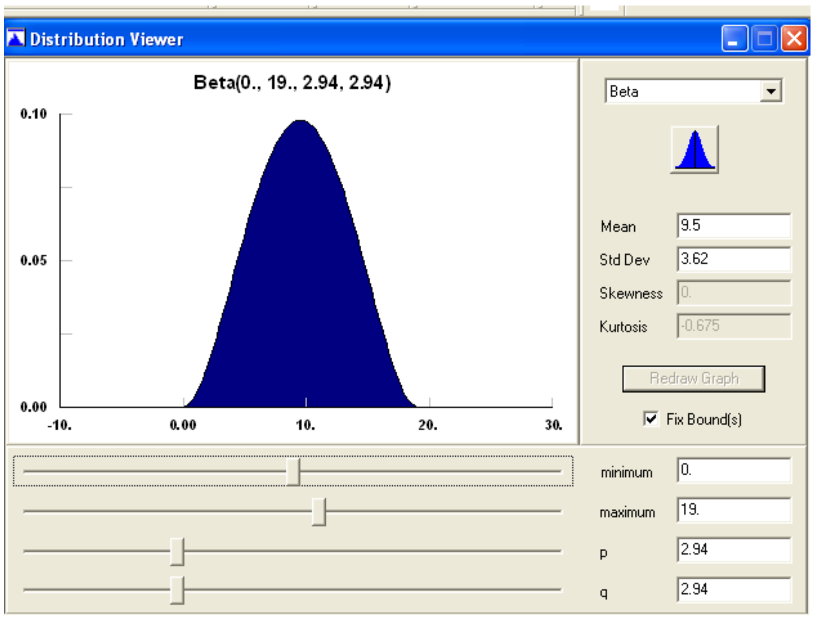


Figure 94: Beta distribution fit to 5 minute Arrival Periods

References

http://www.promodel.com/

http://www.geerms.com/

http://en.wikipedia.org/

Milton, Susan & Arnold, Jesse; "Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences"; 4th Edition

Zar, Jerrold; "Biostatistical Analysis"; 4th Edition