

**Call Center Scheduling Problem**

**Federal Reserve Bank**

**United States Treasury**

**Senior Design Final Report**

**May 6, 2009**

**Brittany McCluskey & Morgan Brunz**

Table of Contents

**Management Summary2**

**Background of Problem Situation3**

**Analysis of the Situation6**

**Technical Description of the Model8**

Waiting Line System8

Service Rate8

Arrival Rate8

Integer Programming Model9

Variables9

Constraints10

Right Hand Side10

Objective11

**Analysis and Managerial Interpretation12**

Solution Model 1 and Analysis12

Solution Model 2 and Analysis14

Solution Model 3 and Analysis15

Solution Model 4 and Analysis17

Solution Model 5 and Analysis19

**Conclusion and Critique21**

Project Summary21

Recommendations21

Self- Critique24

**Appendix25**

FedAmpl Text File – Model 125

**Management Summary**

We were given a call center scheduling problem by the Federal Reserve Bank of Dallas. They have a call center that answers calls to transfer customers from receiving checks in the mail to direct deposit. The center’s call volume varies throughout the month and day. They have hired 23 permanent employees, and on a daily basis, they hire a different amount of temporary employees. Our goal was to determine the number of temporary employees they should hire each day and those employees’ schedules. We wanted to minimize cost and maximize customer service.

We decided to use an integer-programming model to find the number of temporary employees and their schedules. The variables were the possible schedules. The equations stated that the number of employees working per 15-minute time interval was equal to or greater than the number of employees needed. The number needed was obtained by using a waiting line system, which included the service and arrival rates of calls per 15-minute time interval.

We used the mathematical programming language AMPL/CPLEX to model this integer-programming model. We entered a matrix that models the possible schedules, a right hand side indicating the number of employees needed per 15-minute time interval, a constraint that employees working must be equal to or great than employees needed, and an objective to minimize the number of employees. We ran the model 5 different times as we made conclusions about the solutions we were obtaining. In conclusion, we found that when the 23 permanent employee schedules are not changed, then 9 temporary employees are needed. If we can change the schedules of the permanent employees, then we were able to make the greatest improvement with only 25 total employees needed.  **Background of Problem Situation**

Every month the United States Treasury sends out 11 millions checks for social security and other government payments. It costs one dollar to send each check every month. In order to eliminate that cost, they are trying to get people to use direct deposit. This is an environmentally friendly and safer method.

In the checks that people receive is an insert encouraging people to switch from mail check to direct deposit. There are two methods of contacting the FED to have them change your account to direct deposit. You can call or use their website to set it up. Seventy percent of people use the call center and thirty percent use the website.

Depending on the day, the call center can receive between 9,000 and 1,000 calls. Therefore they need between 125 and 15 people depending on the call volume. The FED needed a system to determine how many temporary employees should be hired per day, in addition to their 23 permanent employees that are always there. Below is a graph that shows the fluctuations in call volume over the day for a high and low volume day.

Nine million checks are delivered on the first and third of every month. If the first or third is on the weekend, checks are sent the Friday before. The other two million checks are delivered on the second, third, and fourth Wednesdays of the month. So on those delivery days and the day after, the call center receives a high call volume. The first week of each month has a much greater call volume than the rest of the weeks.

The call center is open from 7 am until 7 pm on Monday through Friday, and the website is open all the time. The call center employees have to answer calls and verify web transactions as well. Priority is given to calls. Employees arrive in four different shifts, and they take two 15-minute breaks and a 45-minute lunch. The shifts begin at 7am, 8am, 9am, and 10:15am. The breaks and lunches are staggered. Lunches can be between 10:30am and 1pm, since that is when the cafeteria is open. On heavy days, they take only 30-minute lunches. There are 23 permanent employees, and each day the FED determines how many temporary workers to hire depending on the call volume.

Our objective for this project was to create algorithms that use daily call forecasting to determine how many temporary employees are needed each day for the second, third, and forth weeks of the month. We were to focus on only those weeks since the volume is steady then. We also needed to determine the agents’ schedules, including start, break, lunch and finish times. We optimized cost functionalities to make sure they are not wasting money on employees they don’t need. So we wanted to minimize cost. At the same time we tried to maximize customer service by minimizing the number of dropped calls.

We used the data given to us including forecasting, monthly projections, employee shift schedules, and call records. Their current methodology includes the program Erlang C. They also use self-assessment and past experience to determine employee levels.

**Analysis of the Situation**

In the first week, the call volume was much higher than the rest of the month. Thus, we determined we would not include the first week in our analysis since it would skew the results if we added it in. So we would only use the call volumes during the 2nd, 3rd, and 4th weeks of the month.

Next, we determined that in any model we used, we would use 15-minute time intervals. We came to this conclusion because of the way the agents’ schedules were set up. They worked 8 hours and 45 minutes per day, and took two 15-minute breaks and a 45-minute lunch.

Our clients gave us a list of the possible schedules they assign the temporary employees to. The list included 24 possible schedules that all temporary agents are arbitrarily assigned to at the beginning of each day. We thought it might be helpful to add some of our own schedules when we developed our model. In addition, we were given the permanent schedules that the 23 permanent employees are assigned. We considered changing their schedules as well if different schedules seemed to be more efficient.

We came up with an initial plan that used two different models. The first was a waiting line system. We would determine the arrival rate, which was the average number of calls that arrived in a 15-minute period. Also, we would determine the service rate, which was the average number of arrivals serviced in a 15-minute interval for an individual agent. The important thing from this first model was the number of agents needed for each 15-minute time interval. We would use these numbers for our second model, an integer-programming model.

Our second model would be an integer-programming model. Each variable would be a possible schedule for an agent. The variables would include the agents’ arrival, break, lunch, and leaving times. Then we would have a series of equations. There would be an equation for each 15-minute period of the workday. Each equation would represent the agents who had already been working, those that were leaving, and those that were beginning work at that time. The total number of agents working at that 15-minute interval would need to be equal to or greater than the number of agents that are needed for that period based on the call volume expected.

**Technical Description of the Model**

Waiting Line System

Service Rate

The first part of our analysis included using a waiting line system. We needed to determine the service rate, which is the average number of arrivals serviced per 15-minute interval for an individual agent. Our clients told us that an average call takes 4 minutes and 20 seconds. We divided 15 minutes by 4 minutes and 20 seconds. Thus, an agent could handle 3.46 calls per 15-minute interval.

Arrival Rate

Next we used historical data to determine the arrival rate, which is the average number of arrivals per 15-minute interval. We wanted to determine the arrival rate for each 15-minute interval between 7am and 7pm for the 2nd, 3rd, and 4th weeks of a month. We did not consider the 1st week of a month because of the extremely high call volume due to the fact that most checks are received on the 1st and 3rd of the month.

Our clients gave us the number of calls received by 30-minute period for every day of the months of December 2008, January 2009, and February 2009. We entered the data only for the 2nd, 3rd, and 4th weeks into a spreadsheet. For each 30-minute time interval, we summed the number of calls received across all the days. Thus, we got the average number of calls per 30-minute time interval of the workday. Next, we divided those numbers by 2 since we would be working with 15-minute intervals in our model. The numbers we generated were our average arrival rates per 15-minute time interval. Next, we found the total number of agents needed per 15-minute time interval by dividing the average arrival rates by the service rate of 3.46. Finally, we rounded those numbers up because the number of agents needed must be whole numbers since they refer to people. That concluded our analysis using a waiting line system.

Integer Programming Model

We determined that we would use the programming language AMPL/CPLEX to solve this integer-programming model. AMPL stands for "A Mathematical Programming Language". We used it to hold our integer-programming model. AMPL uses syntax similar to mathematical optimization models. It uses solvers such as CPLEX to optimize the objective function of the integer-programming model.

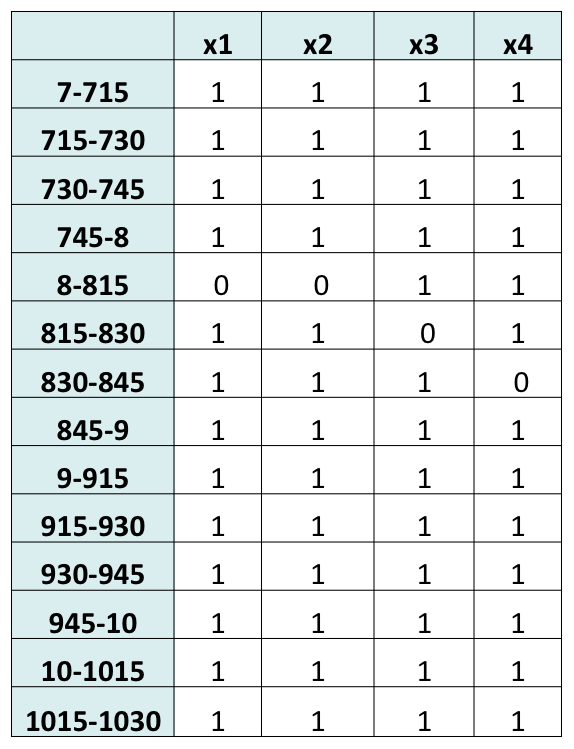
We created 2 text files to hold our AMPL code. The first was named FedFile, and in it we parameterized s and t, and entered our matrix A and the right hand side. The second was FedAmpl, which initialized all our variables, contains our objective and constraints, opens FedFile so the data can go into the constraints, and solves and displays.

Variables

Our second model would be an integer-programming model. We had 24 different variables. Each one represented a possible schedule for the temporary employees. The client gave those possible schedules to us, and they are the schedules currently being used.

To enter our variables in AMPL, we created a matrix A with t number of rows and s number of columns. T stands for time measured in 15-minute intervals, and s stands for possible employee schedules. The matrix is filled with ones and zeros. A one means the employee with a specific schedule is working during a specific 15-minute time interval. A zero means the employee has not arrived, is on a break, is at lunch, or has left for the day.

In order to enter the matrix A into AMPL we first initialized the matrix to be filled with zeros. Then we entered the ones by using the format ‘t, s, 1’. This means that for a certain time period and schedule, an agent is present. Below is a small portion of our matrix A.

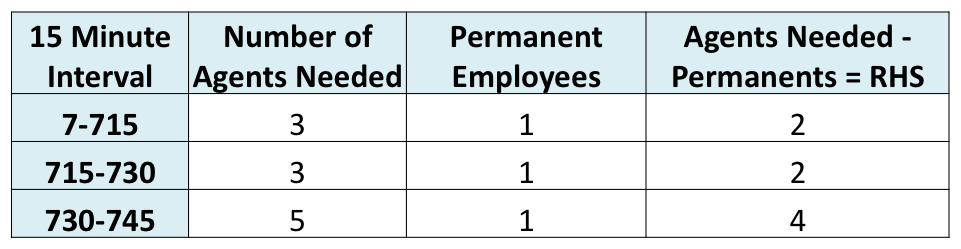


Constraints

We had one constraint that would generate a constraint for each time period of our matrix A. This meant there would be 48 constraints since there are 48 15-minute periods between 7am and 7pm, the workday. Each constraint said that the number of agents working during that time period must be greater than or equal to the number of agents that were needed, which we entered in as our right hand side. For each constraint, the number of ones in the matrix represents the number of agents working for each time period multiple by the agents working on that schedule.

Right Hand Side

Next, we had to enter the right hand side into AMPL. This represents the number of temporary agents needed for each 15-minute time interval. There are 48 entries for the right hand side since there is one for each constraint, or time period. First, we took the total number of agents needed per 15-minute time interval that we got from the waiting line system. We determined how many permanent employees were working during each 15-minute interval by using their schedules. The client provided us with the schedules of the permanent employees. Next, we subtracted the permanent employees working from the total number of people needed for each time period. This gave us the number of temporary agents needed for each time period. We entered that data into FedFile as the right hand side. Below is an example of how we obtained our right hand side.



Objective

Our objective was to minimize the number of variables, or possible schedules, used. This would mean minimizing the number of temporary agents hired each day.

**Analysis and Managerial Interpretation**

Solution Model 1 and Analysis

CPLEX 11.2.0: optimal integer solution; objective 9

6 MIP simplex iterations

0 branch-and-bound nodes

x [\*] :=

1 0 4 1 7 0 10 0 13 0 16 0 19 0 22 0

2 0 5 0 8 3 11 0 14 0 17 0 20 0 23 0

3 0 6 3 9 0 12 0 15 0 18 1 21 0 24 1;

Above is our output from running the model the first time. We were looking for the amount of temporary employees to hire and what their schedules should be. In the right hand side, we took out the permanent employees. We took into consideration the fact that the permanent employees only answer phone calls 80% of their day. For the remaining 20% they are verifying web applications. Our client gave this information to us.

The objective function was 9, which means that 9 temporary employees should be hired. It took 6 iterations for AMPL to come up with this solution. The matrix for x indicated the number of employees and which schedules they will have. For this solution we have the following:

* Schedule 4: need 1 temporary employee to come in at 7:00 to 3:45, breaks 8:30 and 1:45, and lunch at 10:30
* Schedule 6: need 3 temporary employees to come in at 7:00 to 3:45, breaks 8:45 and 1:45, and lunch at 11:00
* Schedule 8: need 3 temporary employees to come in at 8:00 to 4:45pm, breaks 9:15 and 2:15, and lunch at 11:00
* Schedule 18: need 1 temporary employee to come in at 9:00 to 5:45 breaks 10:15 and 4:00, and lunch at 12:00
* Schedule 24: need 1 temporary employee to come in at 10:15 to 7:00, breaks at 3:45 and 5:15, and lunch at 1:00

We noticed that 4 of the 9 temporary agents come in at 7am, which shows that there is a greater need for employees starting in the earlier shifts. In addition, 3 of the 9 start at 8am, 1 starts at 9am, and 1 starts at 10:15am. We will try next to change the permanent employees schedules to start earlier and see if we can get a smaller objective function, which means we will be hiring less temporary employees.

After seeing the AMPL results, we have concluded that changing the permanent employees’ schedules could lead to needing less temporary agents. When we subtracted the permanent employees from the agents needed we saw there was a zero or negative need from 3pm to 7pm. This means there were more permanent employees than we actually need. The greatest need of employees was between 12:15 to 1:15. At 12:45, we show a need of 9 temporary employees.

We believe that breaks should be minimized between 12:15pm to 1:30pm because this is the highest demand time of the day. Also, we want to minimize the amount of people leaving for lunch at 12:15 through 12:45. Having employee begin earlier in the day can solve this problem. We want to minimize the amount of people coming in on the 10:15 shift. In addition, we want to change the temporary or permanent employees to earlier shifts such as the 7am and 8am shifts. Priority to the permanent employees for the later shift will be given.

Solution Model 2 and Analysis

CPLEX 11.2.0: optimal integer solution; objective 21

16 MIP simplex iterations

0 branch-and-bound nodes

x [\*] :=

1 0 4 0 7 2 10 0 13 0 16 4 19 0 22 0

2 2 5 0 8 0 11 0 14 0 17 0 20 2 23 1

3 2 6 1 9 3 12 0 15 0 18 0 21 0 24 4

Above is the output for the second time we ran AMPL. Here we changed the right hand side by taking out the permanent employees so the solution would model both the permanent and temporary employees. This model does not take in account the fact that permanent employees are only working 80% of the day answering the phones.

We obtained an objective function of 21, which includes both permanent and temporary employees. It took 16 iterations to come up with this solution. It means the following:

* Schedule 2: need 2 employees to come in at 7:00 to 3:45, breaks 8:00 and 1:15, and lunch at 10:30
* Schedule 3: need 2 employees to come in at 7:00 to 3:45, breaks 8:15 and 1:30, and lunch at 10:30
* Schedule 6: need 1 employee to come in at 7:00 to 3:45, breaks 8:45 and 1:45, and lunch at 11:00
* Schedule 7: need 2 employees to come in at 8:00 to 4:45, breaks 9:00 and 2:00, and lunch at 11:00
* Schedule 9: need 3 employees to come in at 8:00 to 4:45, breaks 9:15 and 2:15, and lunch at 11:30
* Schedule 16: need 4 employees to come in at 9:00 to 5:45, breaks 10:15 and 3:30, and lunch at 12:00
* Schedule 20: need 2 employees to come in at 10:15 to 7:00, break 10:45 and 4:15, and lunch at 12:30
* Schedule 23: need 1 employee to come in at 10:15 to 7:00, breaks 3:30 and 5:00, and lunch at 1:00
* Schedule 24: need 4 employees to come in at 10:15 to 7:00, breaks 3:45 and 5:15, and lunch at 1:00

In conclusion, there are only 21 employees total including permanent and temporary employees. This is assuming that the permanent employees are answering the phones all the time. The client gave us the statistic that permanent employees are working to answer phone calls 80% of the time, so in our next model we will take this into consideration.

Solution Model 3 and Analysis

CPLEX 11.2.0: optimal integer solution; objective 25

18 MIP simplex iterations

0 branch-and-bound nodes

1 Gomory cut

1 zero-half cut

x [\*] :=

1 0 4 0 7 2 10 1 13 0 16 4 19 0 22 0

2 3 5 0 8 0 11 0 14 0 17 0 20 3 23 1

3 2 6 2 9 3 12 0 15 0 18 0 21 0 24 4

In this model, we factored in the fact that the permanent employees only answer the phone 80% of their day. We did this by increasing the right hand side by multiplying by 1.2 and rounding up. This solution also takes in account permanent and temporary employees.

* Schedule 2: need 3 employees to come in at 7:00 to 3:45, breaks 8:00 and 1:15, and lunch at 1030
* Schedule 3: need 2 employees to come in at 7:00 to 3:45, breaks 8:15 and 1:30, and lunch at 10:30
* Schedule 6: need 2 employees to come in at 7:00 to 3:45, breaks 8:45 and 1:45, and lunch at 11:00
* Schedule 7: need 2 employees to come in at 8:00 to 4:45, breaks 9:00 and 2:00, and lunch at 11:00
* Schedule 9: need 3 employees to come in at 8:00 to 4:45, breaks at 9:15 and 2:15, and lunch at 11:30
* Schedule 10: need 1 employee to come in at 8:00 to 4:45, breaks at 9:30 and 2:30, and lunch at 11:30
* Schedule 16: need 4 employees to come in at 9:00 to 5:45, breaks at 10:15 and 3:30, and lunch at 12:00
* Schedule 20: need 3 employees to come in at 10:15 to 7:00, breaks at 10:45 and 4:15, and lunch at 12:30
* Schedule 23: need 1 employee to come in at 10:15 to 7:00, breaks 3:30 and 5:00, and lunch at 1:00
* Schedule 24: need 4 employees to come in at 10:15 to 7:00, breaks 3:45 and 5:15, and lunch at 1:00

In conclusion, we got four more people, which is about what we were expecting after we factored in the 80% work time for permanent employees for using the phone instead of the website. Our next step will be to determine whether we can change any of the start, break, lunch or finish times for the schedules we are using to see if we can make any improvements.

Solution Model 4 and Analysis

CPLEX 11.2.0: optimal integer solution; objective 24

34 MIP simplex iterations

0 branch-and-bound nodes

x [\*] :=

1 0 5 1 9 2 13 0 17 0 21 3 25 0 29 0 33 0

2 0 6 2 10 1 14 2 18 0 22 0 26 2 30 2 34 2

3 1 7 0 11 0 15 0 19 0 23 2 27 0 31 0

4 1 8 0 12 0 16 0 20 0 24 0 28 2 32 1

We added in 10 more possible schedules that were systematically arranged throughout the day. This includes temporary employees and permanent. We also changed S to 34 because we added additional schedules.

First of all we added in ten more possible schedules that the model could choose from. We staggered the start times to begin every 15 minutes instead of every hour, we also staggered the breaks and lunches more systematically. We found that objective function was 24 instead of 25 people. And the model used 9 people in our new schedules for the model. It used 5 of the possible schedules we made.

* Schedule 3: need 1 employee to come in at 7:00 to 3:45, breaks 8:15 and 1:30, and lunch at 10:30
* Schedule 4: need 1 employee to come in at 7:00 to 3:45, breaks 8:30 and 1:45, and lunch at 10:30
* Schedule 5: need 1 employee to come in at 7:00 to 3:45, breaks 8:45 and 1:45, and lunch at 10:30
* Schedule 6: need 2 employees to come in at 7:00 to 3:45, breaks 8:45 and 1:45, and lunch at 11:00
* Schedule 9: need 2 employees to come in at 8:00 to 4:45, breaks 9:15 and 2:15, and lunch at 11:30
* Schedule 10: need 1 employee to come in at 8:00 to 4:45, breaks 9:30 and 2:30, and lunch at 11:30
* Schedule 14: need 2 employee to come in at 9:00 to 5:45, breaks 10:00 and 3:00, and lunch at 12:00
* Schedule 21: need 3 employees to come in at 10:15 to 7:00, breaks 3:00 and 4:30, and lunch at 12:30
* Schedule 23: need 2 employees to come in at 10:15 to 7:00, breaks 3:30 and 5:00, and lunch at 1:00
* Schedule 26: need 2 employees to come in at 7:30 to 4:15, breaks 9:00 and 1:30, and lunch at 10:45
* Schedule 28: need 2 employees to come in at 8:15 to 5:00, breaks 9:45 and 2:15, and lunch at 11:15
* Schedule 30: need 2 employees to come in at 8:45 to 5:30, breaks 10:15 and 2:45, and lunch at 11:45
* Schedule 32: need 1 employee to come in at 9:30 to 6:15, breaks 10:45 and 3:30, and lunch at 12:15
* Schedule 34: need 2 employees to come in at 10:00 to 6:45, breaks 11:15 and 4:00, and lunch at 12:45

Our next step would be to use the model with only the temporary agents in them but we will use our new schedules in the model.

Solution Model 5 and Analysis

CPLEX 11.2.0: optimal integer solution; objective 9

7 MIP simplex iterations

0 branch-and-bound nodes

x [\*] :=

1 0 5 0 9 0 13 0 17 0 21 0 25 0 29 0 33 0

2 0 6 3 10 0 14 0 18 1 22 0 26 1 30 0 34 0

3 0 7 0 11 0 15 0 19 0 23 1 27 3 31 0

4 0 8 0 12 0 16 0 20 0 24 0 28 0 32 0

We adjusted the RHS back to original that only takes into account the need for temporary employees. This model assumes that the permanent employees has the original schedules we were given. It determines the temporary employee schedules using the 34 possible schedules.

The first time we ran the program with the temporary employee only we got a objective function of 9. So added the additional possible schedules did not help minimize the number of temporary employees needed.

* Schedule 6: need 3 temporary employees to come in at 7:00 to 3:45, breaks 8:45 and 1:45, and lunch at 11:00
* Schedule 18: need 1 temporary employee to come in at 9:00 to 5:45 breaks 10:15 and 4:00, and lunch at 12:00
* Schedule 23: need 1 temporary employee to come in at 10:15 to 7:00, breaks 3:30 and 5:00, and lunch at 1:00
* Schedule 26: need 1 temporary employee to come in at 7:30 to 4:15, breaks 9:00 and 1:30, and lunch at 10:45
* Schedule 27: need 3 temporary employees to come in at 7:45 to 4:30, breaks 9:15 and 1:45, and lunch at 11:00

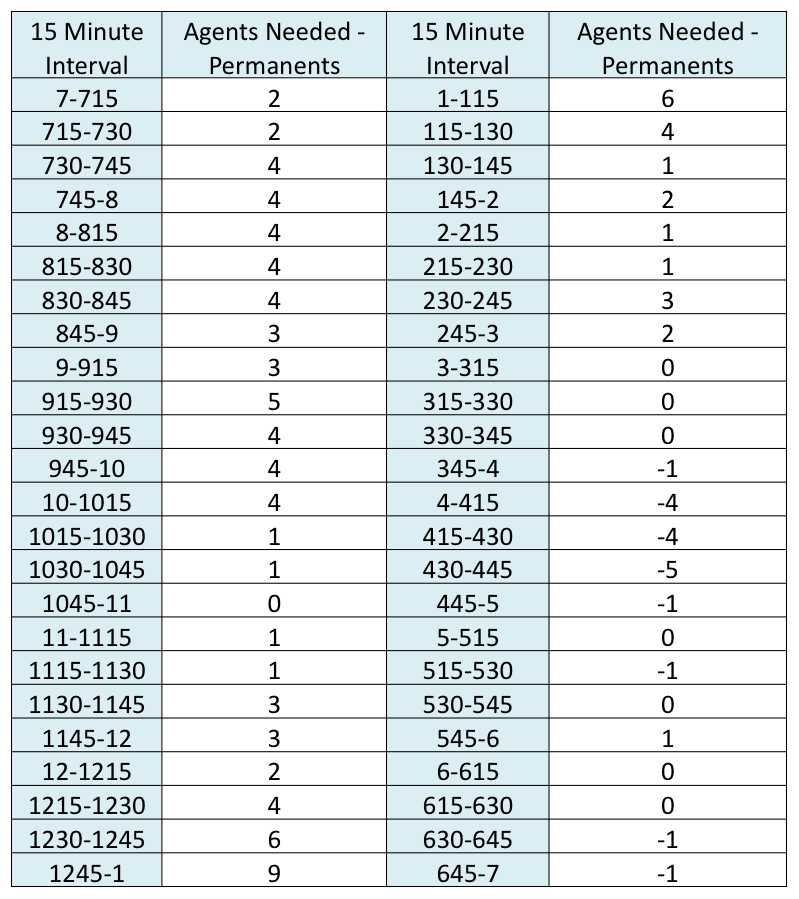
**Conclusions and Critique**

Project Summary

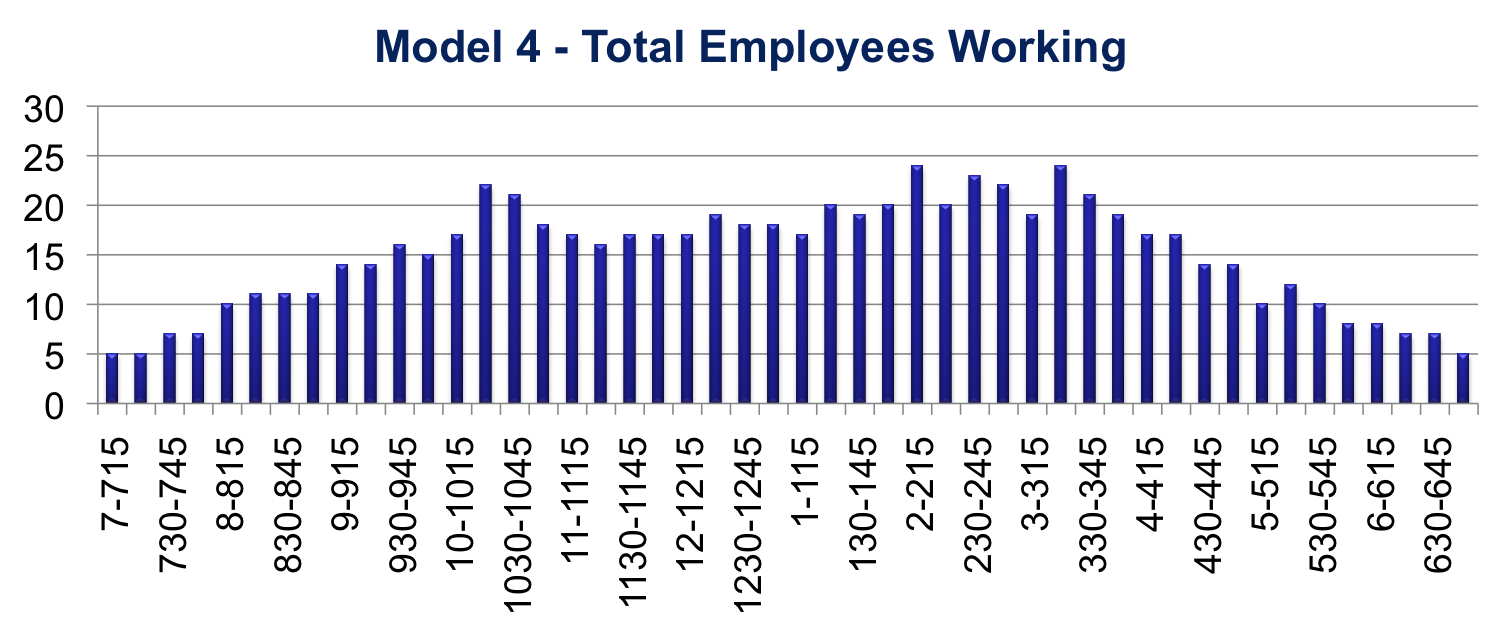
This semester, we worked to develop a call center scheduling model for the Federal Reserve Bank of Dallas. We needed to determine the number of temporary employees to hire per day and what their schedules should be. Our goals were to minimize cost and maximize customer service by hiring the correct number of agents per day. We developed an integer programming model to determine how many employees would be needed per day for the 2nd, 3rd, and 4th weeks of the month. Overall, we concluded that the schedules of the permanent employees need to be changed to earlier shifts for the best results. Through our model and analysis, we were able to reduce their total employees by approximately 50%.

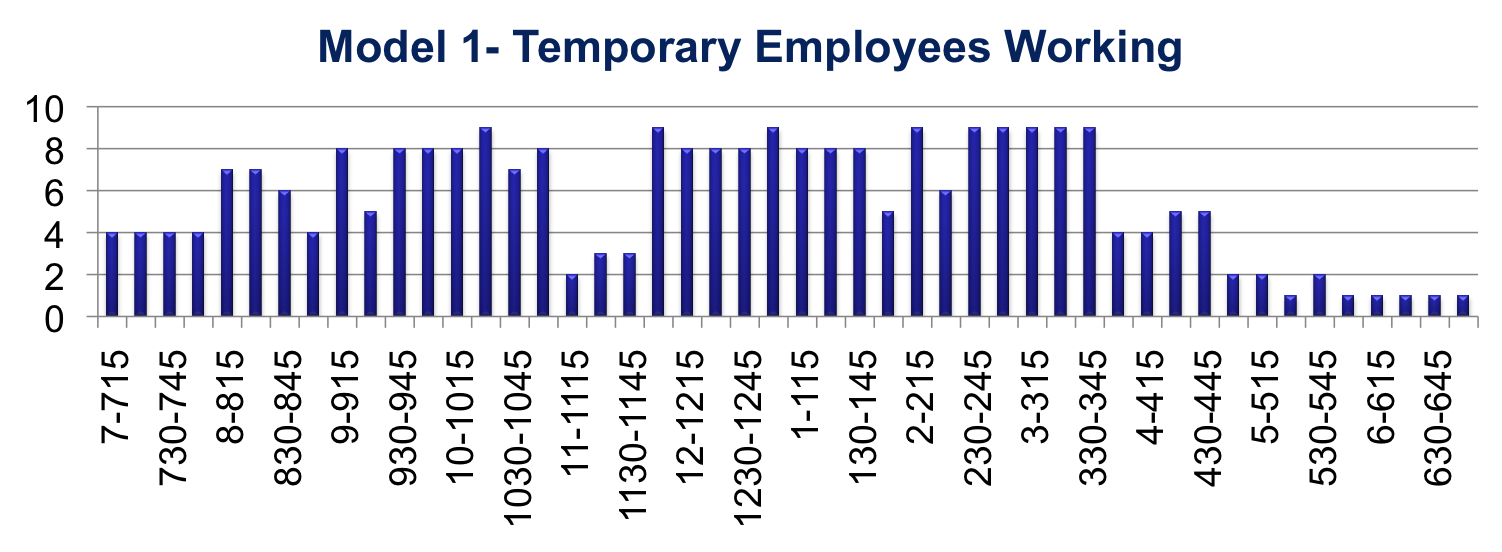
Recommendations

We have seen that changing the schedules of the permanent employees has the greatest impact on minimizing the amount of temporary employees needed. Overall the amount of temporary employees being hired by the FED exceeds the amount needed for the 2nd, 3rd, and 4th weeks of the month. The set schedules of the permanent employees need to be changed because right now too many permanent employees are starting too late in the day. Below is a chart that shows the number of temporary agents needed throughout the day. The negative need in the afternoon shows that employees need to have more early shifts.



With our solution model 4, the FED can approximately cut their number of temporary employees in half. If permanent schedules cannot be changed the FED can use our recommendations about the temporary employee schedules made in solution model 1. Below is a graph of the employees working throughout the day in model 4 and 1. In model 4 there are 24 total employees working and in model 1 there are 23 permanent and 9 temporary employees.





Our model assumes that employees are working very diligently and do not take any personal breaks. For more of a realistic recommendation we would suggest about 2 or 3 extra employees than what the model recommended because that will lead to better customer service rate and less dropped calls.

In general, we would recommend that the client limits the amount of employees taking lunch between 12:15 and 1:15 due to the high call volume demand at this time. Also, shift schedules could be adjusted to beginning at more intervals, such as every 15 minutes instead of every hour. Thus employee schedules could be more staggered.

Another suggestion is to create part-time employee schedules. Employees could come in during the high call volume times. Most calls are received around lunch times, between 10 and 2. This would be suggested for low volume days in order to cover the middle of the day. Also, schedules could be created for employees to come in only 4 days a week and work 10 hours a day. This would help employees with long commutes, and it would lower pollution from cars. These ideas could be explored in the future.

Self-Critique

The way we modeled our problem limited us in several ways. The biggest limitation was the fact that we had to enter in possible schedules by hand. Thus, it was impossible to have an infinite amount of possible schedules. This might have given us a slightly better solution. Another limitation is that 23 permanent employees had to be used. We might have come up with a better solution if we could hire only temporary employees whose schedules could be changed on a daily basis. In addition, the temporary employees were cheaper than the permanent ones.

We were also limited by some of our constraints. We were required to use 8 hour 45 minute shifts with 45-minute lunches and two 15-minute breaks. If we could lengthen or shorten the length of the day and or the lunches and breaks, we might have been able to make improvements on the model.

Overall, we believe we were successful in minimizing the cost and maximizing the customer service at the Federal Reserve call center. We took into account all of the information and constraints that were given to us by the client. We were able to offer our client multiple suggestions about how to improve their scheduling process. If they are able to accept our best suggestions, they can eliminate 50% of the temporary employees they currently hire.

**Appendix**

FedAmpl Text File – Model 1

reset;

option solver cplex;

param S;

param T;

set Schedules:= 1..S;

set Time:= 1..T;

param A{Time,Schedules}default 0;

param RHS{Time};

data C:\SeniorDesign\Fedfile.txt;

display S,T,Schedules,Time,A,RHS;

var x{Schedules}>=0,integer;

minimize People:sum{s in Schedules}x[s];

subject to Const{t in Time}:sum{s in Schedules}A[t,s]\*x[s]>=RHS[t];

solve;

display x;