Problem: Bank Service Problem

The bank manager is trying to improve customer satisfaction by offering better service. Management wants the average customer to wait less than 2 minutes for service and the average length of the queue (length of the waiting line) to be 2 persons or fewer. The bank estimates it serves about 150 customers per day. The existing arrival and service times are given in the tables below.

Time between arrival (min.)	Probability
0	0.10
1	0.15
2	0.10
3	0.35
4	0.25
5	0.05

Table 1: Arrival times

Service Time (min.)	Probability
1	0.25
2	0.20
3	0.40
4	0.15

Table 2: Service times

(1) Build a mathematical model of the system.

(2) Determine if the current customer service is satisfactory according to the manager guidelines. If not, determine, through modeling, the minimal changes for servers required to accomplish the manager's goal.

(3) In addition to the contest's format, prepare a short 1-2 page non-technical letter to the bank's management with your final recommendations.

Businesses are always looking for ways to improve customer satisfaction so that they can attract new customers and retain old ones. In order to accomplish this, a specific bank manager would like to reduce the average time customers spend waiting for services to less than 2 minutes and the average length of the waiting line to less than 2 people. We developed a two part model capable of determining the minimal changes necessary to meet the manager's requirements.

The first part of our model was a purely theoretical approach. We derived a discrete-time equivalent of Lindley's equation, which is typically used to simulate continuous time queues, and created a recurrence relation that allowed us to find the probability distribution of wait times for any given customer. We then used these distributions to provide an exact value for the average waiting time for customers. This approach, however, is not capable of testing data with multiple servers and also does not directly yield the average queue length.

The second part of our model was a computational approach, which we used to test more complex scenarios and find the average queue length. We created an algorithm to simulate the bank's day-to-day operations and then tested our simulation by running multiple trials and comparing the resulting frequency distributions with the theoretical probability distributions. We found that the average waiting times derived from the two approaches agreed to within 0.164 percent. This indicated that our computer simulation could approximate the theoretical values with high accuracy, allowing us to extend our simulation to test the impact of adding new servers, as well as the addition of "emergency" servers who only serve customers when the queue length exceeds a predetermined limit.

Using our model, we determined that the bank's current system limits the average queue size to a relatively small 1.8 customers, but the average customer waits about 5 minutes for service, and some customers wait as long as 8 minutes. We tested two ways to reduce the mean wait time, choosing to also measure server idle time, the amount of time a server spends not helping a customer, in order to determine which method would be more efficient. By modifying the bank's system to use two servers simultaneously, we were able to decrease the average wait time to about 6 seconds and reduce the average queue length to 0.04 customers, but we also greatly increased the time servers spent doing nothing from 37 minutes to 430 minutes (more than 7 hours). By adding an emergency server who would only begin serving customers when the queue reached 3 customers, however, we were able to reduce the wait time to 1.46 minutes and the queue length to 0.55 customers while keeping the idle time to a more reasonable 62 minutes. Furthermore, this change would only require the emergency server to work for about 40 minutes each day, a relatively minimal change. Our model shows that adding a second "emergency" server is the most efficient method to reduce average customer wait times and average queue length to within the requirements.

Solution Paper (Problem B)

Introduction

Customer satisfaction is of vital importance for companies whose customers frequently interact with company employees, especially when many other companies in the area offer competitive services. In the banking industry, the waiting time for a given customer before they are served and the length of the line are two factors that can greatly affect whether or not a customer has a pleasant experience. Unfortunately, due to the unpredictable nature of customer arrivals and the varying time required to serve each customer, it can be difficult to determine whether a current system is satisfactory. In this paper we provide two methods to model these factors and propose a strategy for a bank to raise customer satisfaction with minimal changes to its current system.

Problem Restatement

A bank is attempting to improve customer satisfaction by offering better service to its customers. Specifically, the management wants to ensure that on average, customers wait no longer than 2 minutes before receiving service and the waiting line is no more than 2 people long. We are provided the probability distribution of the difference between customer arrival times (ranging from 0-5 minutes) and the probability distribution of the time it takes for the bank to serve a customer (ranging from 1-4 minutes). Using these probabilities and assuming that 150 customers arrive at the bank each day to receive service from only one server (teller), we are tasked with establishing whether or not the bank's current system is satisfactory. If necessary, we can then determine the minimal changes for servers required to accomplish the management's goal.

Assumptions

- Customers only arrive at the bank and are served at exact minute intervals.
  Justification: The data given to us is only applicable by the minute, so estimating data and probabilities in between minutes is impossible.

- Customers are served in the order they arrive at the bank.
  Justification: Most lines (queues in general) work this way. This is necessary in order to properly count the waiting time of customers.

- Servers work continuously until all 150 customers have been served (no breaks), and the time difference between serving customers is negligible.

Justification: As soon as one customer is done being served, the next customer should immediately begin receiving service in order to keep times on the minute.

- The service time data provided corresponds to the rate of service of a single server, and this single server serves all the customers in the original service system.
  Justification: The provided data implies that there is only one line and one server, and the rate of service for a single server needs to be consistent for us to create a model.

- Multiple servers work at the same rate.
  Justification: Servers need to all work at the same rate given to us in order for us to be able to predict the outcome of waiting times and the length of the queue.

- The time for an "emergency" (back-up) server to begin servicing customers from when he or she is called is two minutes.
  Justification: It is not practical for someone to immediately begin working when they are called, so we added a two minute delay period during which the worker would be transitioning.

For the purposes of our model, we will also define the following:

- Customers are numbered from 1 to 150 in the order that they arrive.
- The queue is the line in which the customers stand waiting to be served.
- A server is a person who is capable of providing service to customers
- All times are measured in minutes unless otherwise specified.
- An emergency server is a server that only begins working whenever the queue length exceeds a predetermined limit and stops working whenever the queue is empty.
- q_{enter} is the length of the queue before an emergency server begins to provide service to customers.
- A probability mass function, or PMF, is the discrete-time equivalent of a probability density function. A PMF gives the probability that a random variate is exactly equal to any given integer value [2]. For example, if *F* is a PMF, then $F(x) = P(f = x)$.

Designing the Mathematical Model

We approach the problem from two different approaches: one using purely mathematical methods which yields exact theoretical results and one using a computer simulation that yields approximate results for more complex situations.

Purely Theoretical Approach

The problem can be interpreted as a discrete-time version of a G/G/1 queue (a queue with two separate non-exponential probability distributions that determine when people and leave the queue, and with a single server). In general, a continuous-time G/G/1 queue can be modeled using Lindley's integral equation [1], given by

$$W(x) = \int_{0^-}^{\infty} U(x-y)dW(y),\ \ x \geq 0$$

where

- *W(x)* is the probability that the nth customer waits for no more than x minutes as n tends to infinity
- *U(x)* is the probability that the difference between the previous customer's service time and the nth customer's arrival time is less than or equal to x minutes as n tends to infinity
- *dW(y)* is the infinitesimal probability that the nth customer waits for exactly y minutes as n tends to infinity

We derive a discrete-time equivalent of this equation to find the theoretical waiting time distribution of any given customer. We decided to compute the discrete-time version using probability mass functions instead of cumulative density functions, as we are given tables that match discrete time intervals with probabilities. Also, as we were given an explicit estimate of number of consumers, we decided not to take the limit as customer number -> infinity, but instead calculate each customer's wait-time distribution separately. We found that the distribution of waiting times for the nth customer can be found solely on the basis of the distribution of waiting times for the previous customer and the data provided in Tables 1 and 2. The following formula summarizes our relation:

$$W_n(y) = \begin{cases} \displaystyle\sum_{i=0}^{\max\{w_{n-1}\}} W_{n-1}(i)U(y-i) & \text{if } y > 0 \\ \displaystyle\sum_{i=0}^{\max\{w_{n-1}\}} \left(W_{n-1}(i) \sum_{j=0}^{j_{max}} U(-i-j) \right) & \text{if } y = 0 \end{cases} \qquad \text{[Eqn. 2]}$$

where

- *max{w$_{n-1}$}* is the maximum possible wait time of the (n-1)th customer
- *W$_n$(y)* is the probability that the nth customer waits exactly y minutes
- *U(x)* is a probability mass function that gives the probability that $s_{n-1} - t_n = x$, where s_{n-1} is the service time of the (n-1)th customer and t_n is the time interval between arrivals of

the (n-1)th and nth customers. It can be directly constructed from the provided probability distributions in Tables 1 and 2.

- j_{max} is a constant indicating the maximum time in minutes between the end of the nth customer's service and the arrival of the (n+1)th customer. Given the provided data, we can set $j_{max} = 4$.

The full derivation of this recursive relation from the given data can be found in Appendix A.

Given the initial condition that $W_1(0) = 1$ (the first customer has a 100% chance of having no wait time), we can then use these two recurrence relations to generate $W_n(x)$ for each n > 1. This in essence allows us to construct probability distributions of the wait times for any customer arriving at the bank. Once we have computed $W_1(x)$, … $W_{150}(x)$, we can compute the average waiting time of the nth customer by treating W_n as the weights for a weighted average of the integer waiting times, as shown below.

$$\overline{w}_n = \sum_{i=0}^{\max\{\mathrm{w}_n\}} iW_n(i)$$

We can then find the average waiting time of all customers by taking the mean of $\overline{w}_1, \overline{w}_2 \dots, \overline{w}_{150}$.

This purely theoretical approach provides an exact mathematical formulation for the average waiting time of customers on a given day and can be extended to accommodate for cases with an arbitrary number of people or different probability distributions. However, the complexity of evaluating $W_n(x)$ quickly escalates as n increases, making it relatively unfeasible to evaluate for large n, and this approach also cannot easily incorporate multiple servers working simultaneously. Additionally, this purely mathematical method only finds the average waiting time for each customer in the queue, which cannot easily be converted into the mean queue length.

Computational Approach

In order to simulate more complex circumstances (specifically the effects of adding more servers) and estimate the mean queue length, it is much more feasible to analyze the results of multiple computer simulation trials. We designed a computer model based on state transitions at each discrete time step: at any given minute, customers arrive at the bank, customers finish being served, other customers begin being served, and emergency servers transition between serving as tellers and performing other tasks.

We differentiate between two types of servers: regular servers, who are ready to accept customers at all times, and *emergency servers*, who usually perform other, non-customer-service tasks but can act as regular servers if needed, but only after some constant transition period.

We simulate the bank queue using the following algorithm:

1. All 150 customers are initialized, each with randomly determined *t* (time between arrival of previous customer and arrival of current customer) and *s* (service time).
2. Using t_1, t_2, … t_{150}, the arrival time *a* is computed for all the customers using cumulative summation.
3. An internal clock variable *time* is set to 0 minutes. All customers are placed into a customer queue, and a list of servers is created.
4. Perform the following procedure until all consumers are served, the waiting queue is empty and no customers are in service:
    a. All customers who were calculated to arrive at this time step (*a* = *time*) are removed from the customer queue and added to the waiting queue.
    b. All servers decrease their service counters by one. If the counter reaches 0, remove the currently served customer and mark the server as inactive (not serving a customer).
    c. For every emergency server in the server list, increment the emergency-server-time-spent counter.
    d. For every server that is currently inactive:
        i. If this server is an emergency server and the waiting queue is smaller than the emergency server's exit queue length, mark the emergency server for removal from the server list.
        ii. Otherwise, if there is anyone waiting in the waiting queue, remove the next customer from the waiting queue and record their waiting time (the difference between *time* and *a*). Set this server as active and set their service counter to *s* (the time this customer will be served for)
        iii. If there is no one waiting in the queue and this server is not an emergency server (in other words, if this server remains inactive), increment the idle-time counter.
    e. If there are emergency servers not on duty and the queue is larger than or equal to the emergency enter queue length, add an emergency server to the server list and decrement the number of not-on-duty emergency servers. Give this emergency server a service time equal to the transition time, as this server will be unable to accept a customer until after the transition.
    f. Record current values of queue length.

At the end of every run, the average queue length for that "day" can be found by summing up all the recorded queue lengths and dividing by the final value of *time*. Similarly, the average wait time for each customer can be found by averaging the w's for all of the customers. In this way, our simulation accurately models the proceedings of a random day at the bank, and its versatility allows it to be easily extended to fit new circumstances, such as the addition of new servers or changes in server efficiency. However, because our algorithm simply produces a single random outcome during each run, it can only approximate the true average wait time. Multiple trials thus become essential to increasing the confidence of our results.

By comparing our experimental distribution with our theoretical distribution for cases involving only one server, we can determine the veracity and accuracy of our experimental results, allowing us to proceed in cases that the theoretical approach cannot handle with greater confidence. The full python script used to generate our data is included in Appendix C.

Model Data and Testing

Using Wolfram Mathematica, we iteratively evaluated the recurrence relations in Equations 2a and 2b from the starting point $W_1(0) = 1$, and found the average wait time $\overline{w}$ to be 4.92761 minutes. We also computed the probability mass distribution $\overline{W}(x)$ (average probability that any customer waits x minutes) taking the average of each customer's wait time distributions. The theoretical values of $\overline{W}(x)$ can be found in Table 5 of Appendix B. Interestingly, with the given probabilities for differences in arrival time and service time, there is a $\overline{W}(0)$ = 25.08% chance that any given customer is served immediately, indicating that over one quarter of customers should not have to wait even if only one server is present.

Experimentally, we ran our computer simulation 10000 times, recording the wait times for every customer and the average queue length for each trial. Plotting the experimental wait times against the theoretical wait times ($\overline{W}(x)$), we can see that the two distributions match each other almost exactly - the data points for the theoretical values are hardly even visible (Figure 1). A more detailed comparison of the theoretical and experimental values can be found in Table 5 of Appendix B.

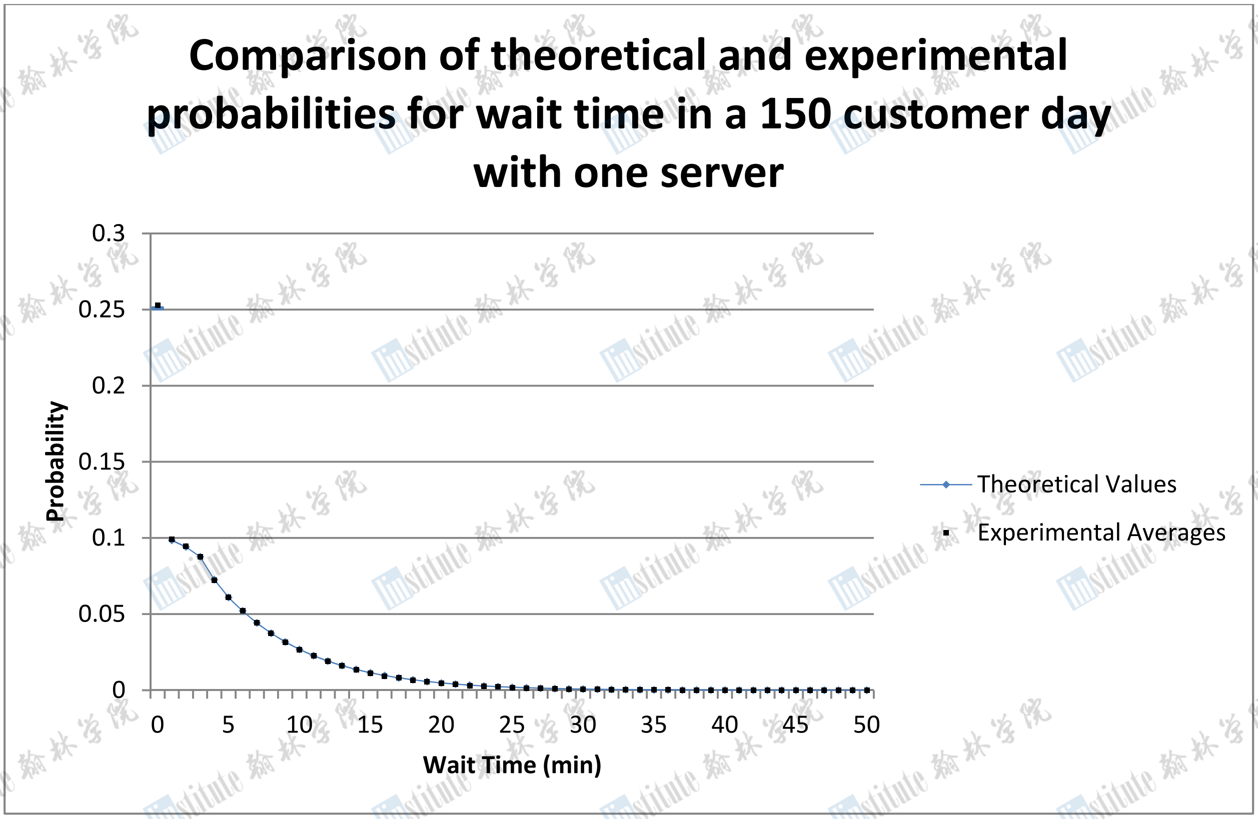

Figure 1: A comparison of the theoretical and experimental probability distributions for average customer wait time on a 150 customer day with only one server available. Notice that the two graphs match each other almost exactly, indicating that our computational simulation has a high accuracy when run over 10,000 trials. The only significantly visible error is the slightly higher experimental value for wait time 0.

We computed the average wait time for the theoretical distribution to be 4.92761 min., whereas the average wait time for the experimental distribution over 10000 trials evaluated to 4.9195 min., a difference of only 0.164%. With this level of accuracy, we can safely extend our computer simulation into cases with multiple servers without having to worry about insufficient confidence levels.

Because our theoretical model is incapable of evaluating average queue times, we assumed that our simulation's accuracy in average wait times is indicative of overall accuracy, especially accuracy in average queue length, as the two quantities are closely related. Though this assumption is not entirely justified, strong correlation between the theoretical and experimental wait times still helps to support the validity of our computation model.

With only one server available, the experimental average queue length evaluated to 1.84773, a value already less than the 2 persons or fewer goal desired by the manager. Because our average wait time of 4.92761 minutes per customer is considerably over 2 minutes, however, a strategy must be adopted to help reduce this average wait time. Furthermore, the standard deviation of average wait times on each given day is 3.30591 minutes, indicating that the wait times vary greatly between days.

Potential Changes

We tested two different changes in server structure in order to reduce the average customer wait time:

1. Increase the number of servers
2. Have additional "emergency" servers that only work when queue length exceeds a certain number

Strategy 1: Increase the number of servers

By introducing only one additional server to the bank, we can drastically reduce both the average queue length and average wait time of customers. Using our computer simulation, we obtained an average wait time of 0.11285 minutes (about 6.77 seconds) and an average queue length of 0.043093, well within the 2 minute and 2 customer bounds specified. Furthermore, the standard deviation of the daily average wait times is a negligible 0.06283 minutes (about 3.8 seconds), indicating that the wait time is very consistently small. However, note that when two servers are assigned there is a large period of time for which one or both of the servers are idle (on average 429.8945 server-minutes, or 7.165 server-hours), resulting in a greatly reduced work efficiency. This data is summarized in Table 1.

	Mean wait time	Mean queue length	Mean server idle time
Full-time server	0.112851	0.043093	429.8945

Table 1: A summary of the mean statistics for the addition of a full-time server. A full-time server lowers both the wait time and queue length well below the manager's goals, but significantly increases the amount of time servers spend idling, reducing in greatly reduced worker efficiency.

From the perspective of a manager, adding another dedicated (full-time) server would definitely reduce the average queue length and wait time to values below his/her desired goals, but at the same time this strategy would waste money on hiring a full-time server who would only work for a small percentage of the total time.

Strategy 2: Add an emergency server

Unlike a dedicated server, an emergency server only begins working when the queue length exceeds a predetermined limit (q_{enter}) and continues to work until no customers remain in the queue. Because it would be unreasonable for an emergency server to switch between tasks instantaneously, we added a two minute transition delay before the emergency server could begin his/her duties as server into our simulation.

We tested various values of q_{enter} to minimize the amount of time the emergency server spends serving customers while still keeping the average wait time below two minutes. Using our computer simulation, we obtained average wait times and average queue lengths for the two minute and two customer bounds specified. While adding an emergency server also creates time for which the dedicated server is idle, this idle time is much less than that in strategy one. By setting q_{enter} to 3 customers, we can minimize the average server downtime to only 61.7 minutes while still satisfying the manager's desired conditions. Our results are summarized in Table 2:

q_{enter}	Mean emergency server use time	Mean customer wait time	Mean queue length	Average number of times emergency server is called	Mean server idle time
2	64.7958	0.927046667	0.350106628	10.7512	75.2787
3	38.5827	1.464951	0.552342	4.6354	61.7095
4	25.0496	2.034833	0.7677	2.3729	53.0709

Table 2: A comparison of three values of q_{enter} for an emergency server. The emergency server with q_{enter} = 3 meets the established requirements of the manager while minimizing the mean server idle time and time spent by the emergency server providing services.

Comparison of Strategies

Overall, adding an emergency server with q_{enter} = 3 keeps the average wait time under two minutes and the average queue length to at most two customers while limiting the time that additional server provides service to an average of only 38.5 minutes each day and keeping the server idle time to a reasonable 61.7 minutes. This makes adding an emergency server the minimal change for servers required to accomplish the manager's goal. A comparison of the approaches for an emergency server, the simple additional server, and the original single server are shown below in Table 3, and graphical representation comparing the distribution of customer wait times for these three strategies can be found in Figure 3 of Appendix B.

	Mean total server idle time	Mean additional server work time	Mean customer wait time	Mean queue length (customers)
Single server	36.8365 min	n/a	4.91953 min	1.847728
Emergency server (q_{enter} = 3)	61.7095 min	38.5827 min	1.464951 min	0.552342
Full-time second server	429.8945 min	397.6912 min	0.112851 min	0.043093

Table 3: A comparison of three different service systems. Note that the emergency server meets the goals required by the manager while minimizing the mean total server idle time.

Note that while adding a full-time server significantly reduces the mean wait time and mean queue length, adding a single emergency server with q_{enter} = 3 is sufficient to meet the manager's requirements while significantly reducing the total amount of time each server spends idle.

Sensitivity Analysis

We also attempted to determine how sensitive our model was to differing parameters, as fluctuations in the measurement of the bank's original data might make the calculated values significantly different. Additionally, on any given day, the probabilities may vary slightly due to external events.

To determine how our model responds to these fluctuations, we ran our model using sets of slightly different probabilities. Our model proved to be very sensitive to small changes in probability. For the single-server case, after increasing the rate of customer arrivals by shifting 5% of the arrival time distribution, the average customer wait time increased to 8.2 minutes, and after decreasing the rate of customer arrival the wait time decreased to 3.1 minutes. Modifying the service times in a similar fashion caused wait time to fluctuate between 6.0 minutes and 4.0 minutes. Making larger changes to the frequency distributions led to even larger changes, in one case even raising average customer waiting time to 57 minutes.

When we tested these different distributions with the presence of an emergency server, however, the time fluctuations were greatly reduced. The small changes to probability distribution barely affected the average wait time, which stayed between 1.3 and 1.6 minutes on average. Even the large change that caused the single-server case to increase to 57 minutes only increased the waiting time to 2.3 minutes when an emergency server was present. Further details of these fluctuations can be found in Appendix D.

These results reveal that the single-server system currently in place depends greatly on the distribution of customer arrival times and service durations. This makes sense, as if the single server cannot keep up with the customer arrivals, the queue quickly grows and increases the waiting times of many customers. When an emergency server is present, however, these fluctuations can be greatly reduced because the second server is ready to step in once the queue increases in length. This demonstrates that our recommended emergency server system is capable of effectively adapting to different circumstances and random variation.

Strengths of Our Model

- Our model uses both theoretical and computational methods to generate our data. The theoretical approach is based on purely mathematical methods, ensuring that its results are exact. The computational method is based on a computer program, making it easily adaptable to varying conditions such as an increased number of servers, different rates of entry and exit, and the addition of emergency servers.

- Our computational method produces results nearly identical to the theoretical method when tested on the single server situation. This strongly verifies the precision and accuracy of our data and allows us to apply our computation model to different situations with high confidence.
- We consider two possible methods of increasing customer satisfaction rather than merely one. Our model allows us to generate concrete data and easily determine which change is more satisfactory using metrics including server idle time, average queue length, and average waiting time. This ensures that our suggested changes are both practical and effective.
- Our model can easily be adapted to use different probability distributions, which makes it applicable to multiple situations, including queues at other banks or at different types of business.
- Because our model produces a probability distribution for average customer wait times as a by-product, businesses can extract other meaningful information about wait times from it, including, for instance, the probability that a customer will have to wait for more than 10 minutes.

Weaknesses of Our Model

- We do not have any way to find exact theoretical values for the queue length distribution, so we have to approximate these values through our computer simulation, which can only approximate the true values.
- Due to the discrete data that the bank gives us, our model cannot account for events that occur in between minutes. This space of time is considerable, especially in comparison to the two minute restriction imposed by the manager, as many different events that significantly affect the inputs of our model can occur within these time windows.
- Our model is unable to account for different rates of customer influx, such as more frequent arrivals during rush hour and less frequent arrivals during lunch time.
- We do not consider other potential methods of increasing customer satisfaction, such as training workers to reduce service times.

Conclusion

We model the bank service queue using two different approaches: purely theoretical mathematics and computational simulation. By utilizing a series of recurrence relations on several probability mass distributions derived from the given data, we can compute the exact average waiting time for queues with a single server. However, because this method is incapable of taking into consideration multiple servers and cannot compute the average queue length, we also utilize a computer program which is capable of stochastically simulating the proceedings of any one day. By comparing the theoretical and computational results for a queue with a single server, we are able to verify the reliability and accuracy of our computational model from the nearly identical wait time distributions and average wait times. With this increased confidence, we can extend our computational model to incorporate cases that our theoretical approach cannot handle, such as the addition of new full-time servers or emergency servers, while also estimating the average queue length by averaging over multiple trials.

Our computational model suggests that the best method for management to satisfy the given requirements (at most an average queue length of 2 and at most an average wait time per customer of 2 minutes) would be to implement a single emergency server who would only begin working when the queue length exceeded 3 customers. If this is not feasible, adding an additional server would also reduce the average queue length and average wait time significantly below the manager's requirements, though this approach would result in a significant amount of server idle time and thus reduced worker efficiency. The emergency server would only need to provide service for customers for an average of about 40 minutes per day, leaving plenty of time to accomplish other tasks that a full-time server would not be capable of completing. This makes using a single emergency server more profitable for the bank. Additionally, during the time when the emergency server is not interacting with customers, he/she could be performing other tasks such as organizing files, returning messages, or answering the phone. Through our sensitivity analysis, we also determined that, when an emergency server was present, wait times fluctuated only slightly given different arrival and service rates, as the emergency server would always be ready to provide services to customers once the queue increased in length.

In the future, we could apply our computational method to larger scale banks with more servers and more customers. This would make our model more applicable to practical situations. It would be interesting to consider different probability distributions of customer arrivals depending on the time of day, as customers are generally more inclined to arrive at the bank at certain times of day, such as rush hour. We could also look at the economic impacts of adding extra servers on the management. Additionally, if we had time, we could attempt to construct a computational method with continuous intervals between arrivals and continuous service lengths, using integration and probability density functions to model the situation with greater accuracy.

Works cited

1. Liu, John C.S. "G/G/1 Queueing Systems." *Computer System Performance Evaluation (CSC5420)*. N.p., n.d. Web. 17 Nov. 2013. <http://www.cse.cuhk.edu.hk/~cslui/CSC5420/GG1.pdf>.
2. Stewart, William J. *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*. Princeton, NJ: Princeton UP, 2009. Print.

Appendix A: Derivation of our Theoretical Model

For the purposes of this derivation, we define the following:

- t_n is the time in minutes between the arrival of the (n-1)th customer and the nth customer.
- s_n is the time in minutes it takes for the nth customer to be served.
- w_n is the difference in minutes between the nth customer's arrival time and the time when he/she begins getting served, equivalent to the nth customer's waiting time.
- Probability mass functions T and S represent the probability tables 1 and 2, such that, for example, $T(1) = P(t_n = 1) = 0.15$.
- In general, lowercase symbols such as t, s, w, and u refer to individual random variates (particular outcomes of a random process), whereas uppercase symbols such as W and U represent PMFs that give the probability distribution of all possible values for their corresponding lowercase symbol.
- j_{max} is the maximum time in minutes between the end of the nth customer's service and the arrival of the (n+1)th customer.
- max{a, b} is defined as the largest value in the set {a, b}, whereas max{a} is defined as the maximum possible value the random variate a can hold.

We derived a discrete-time equivalent of Lindley's integral equation as follows. We decomposed the problem into a series of recurrence relations, with the waiting time of each customer depending on the waiting time of the previous customer.

The waiting period of the nth customer begins as soon as they enter the queue, which is defined as t_n minutes after the previous customer enters the queue. This waiting period ends as soon as the previous customer is done being served, which happens s_{n-1} minutes after that customer leaves the queue. Thus, in general, the nth customer's waiting time (w_n) is given by the waiting time of the customer directly preceding him or her (w_{n-1}) plus the time it takes for the previous customer to be served (s_{n-1})) minus the interval between the nth customer's arrival and the previous customer's arrival (t_n). This relationship is illustrated in Figure 2.

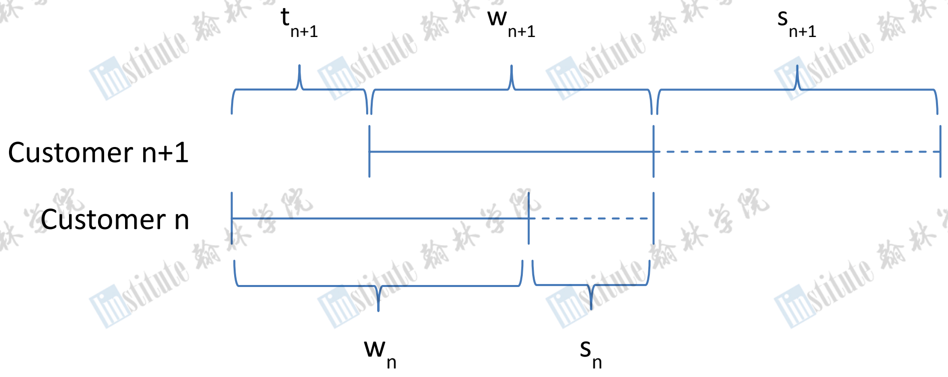

Figure 2: A diagram illustrating the derivation for the value of w_n given w_{n-1}, s_{n-1}, and t_n.

However, because the waiting time can never be negative, our formula can be revised to:

$$w_n = \max\{w_{n-1} + s_{n-1} - t_n\ , 0\} = (w_{n-1} + s_{n-1} - t_n)^+$$

For the purposes of simplification, let us define the sequence $u_2, \dots\ , u_{150}$ such that for every $2 \leq n \leq 150$, u_n is the difference between the (n - 1)th customer's service time and the nth customer's arrival time interval:

$$u_n = s_{n-1} - t_n$$

Substituting this into our prior equation gives:

$$w_n = (w_{n-1} + u_n)^+ \quad \text{[Eqn. 1]}$$

which demonstrates that if w_n is not zero, then u_n represents the difference between the nth customer's waiting time and the (n-1)th customer's waiting time.

We then define the corresponding probability mass function U such that

$$U(x) = P(u_n = x) \text{ for any } n$$

Note that U is independent of n, as the distribution of service time and time between arrivals does not depend on the identity of the customer. In fact, as the PMFs of both T and S are given in Tables 1 and 2, it is possible to compute U directly. By inspection, U(x) yields 0 for all values outside [-4, 4], as the difference between service and arrival times is at least -4 (corresponding to s_{n-1} = 1 and t_n = 5) and at most 4 (corresponding to s_{n-1} = 5, t_n= 1).

Then for each x within the interval, we can directly evaluate U(x) as the sum of all possible combinations of service and arrival times whose difference is x. For example,

$$U(2) = P(s_{n-1} = 4 \wedge t_n = 2)P + P(s_{n-1} = 3 \wedge t_n = 1) + P(s_{n-1} = 2 \wedge t_n = 0)$$
$$= (.10)(.15) + (.15)(.40) + (.10)(.20) = .095$$

Manually extending this process from t = -4 to t = 4 and taking into account the fact that P(x) = 0 for all $x \notin [-4, 4]$ yields the following table:

x	U(x)
… -6, -5	0
-4	0.0125
-3	0.0725
-2	0.1575
-1	0.2025
0	0.235
1	0.1475
2	0.095
3	0.0625
4	0.015
5, 6, …	0

Table 4: Values of U(x) determined from provided Tables 1 and 2.

With this distribution, we can now create a recurrence relation defining the probability distribution of waiting times W_n in terms of W_{n-1} by extending our previous equation $w_n = (w_{n-1} + u_n)$^+ [Eqn. 1] into distributional space.

If we let y take the value of each possible wait time for the nth customer, we can divide the possible values of this distribution into two cases: y > 0 and y = 0.

We can derive $W_n(y)$ for all y>0 by summing the probabilities of each possible combination of w_{n-1} and u_n such that $w_{n-1} + u_n = y$. Each of these probabilities is given by:

$$P(w_{n-1} = i \wedge u_n = y - i) = W_{n-1}(i)U(y - i)$$

for some integer i between 0 and max{w_{n-1}}, the maximum length of time that the previous customer could have waited. This value can be found by finding the maximum time x such that $W_{n-1}(x) \neq 0$. Summing these values gives

$$W_n(y) = \sum_{i=0}^{\max\{w_{n-1}\}} W_{n-1}(i)U(y - i) \text{ if } y > 0 \quad \text{[Eqn. 2a]}$$

Note that y-i may take values outside [-4, 4] because max{w_{n-1}} may be greater than 4. Because U(x) yields 0 for all values outside this range, however, these terms of the sum do not affect the final result.

If y=0, however, we need to take into account the fact that waiting times cannot be less than zero. Thus, if $w_{n-1} + u_n$ is negative, w_n must still be equal to zero because $w_n = \max\{w_{n-1} + u_n, 0\}$. In other words, the probabilities $P(w_{n-1} + u_n = 0)$, $P(w_{n-1} + u_n = -1)$, $P(w_{n-1} + u_n = -2)$, … - need to be summed to calculate $W_n(0)$. Thus our expression becomes:

$$\mathrm{W_n}(0) = \sum_{i=0}^{\max\{w_{n-1}\}} \left(W_{n-1}(i) \sum_{j=0}^{j_{max}} U(-i-j) \right) \text{if } y = 0 \quad \text{[Eqn. 2b]}$$

For simplicity, we chose to let $j_{max} = 4$, the minimum number required to cover all possible U values in all cases.

Combining these two equations yields our final equation:

$$W_n(y) = \begin{cases} \displaystyle\sum_{i=0}^{\max\{w_{n-1}\}} W_{n-1}(i)U(y-i) \ \text{ if } y > 0 \\ \displaystyle\sum_{i=0}^{\max\{w_{n-1}\}} \left(W_{n-1}(i) \sum_{j=0}^{j_{max}} U(-i-j) \right) \text{if } y = 0 \end{cases} \quad \text{[Eqn. 2]}$$

Appendix B

Table 5: Theoretical and experimental values of $\overline{W}(x)$ for values of *x* from 0 to 5

x	$\overline{W}(x)_{\textit{theoretical}}$	$\overline{W}(x)_{\textit{experimental}}$
0	0.250835307	0.249608994
1	0.098455397	0.098389626
2	0.094179782	0.093728065
3	0.087637054	0.087623596
4	0.072659258	0.072983742
5	0.061006037	0.061038971
6	0.052221589	0.052988052
7	0.044280849	0.044584274
8	0.037492559	0.037916183
9	0.031765768	0.031980515
10	0.026885639	0.026967049
11	0.022731497	0.022891045
12	0.019204206	0.019203186
13	0.016209968	0.016200066
14	0.013669888	0.013542175
15	0.011517118	0.011408806
16	0.009694141	0.009776115
17	0.008151753	0.008030891
18	0.006847942	0.006660461
19	0.005746819	0.005592346
20	0.00481775	0.004709244
21	0.004034607	0.003953934
22	0.003375122	0.003400803
23	0.002820332	0.002817154
24	0.002354099	0.00228405
25	0.001962701	0.002033234
26	0.001634481	0.001657486
27	0.001359544	0.001452446
28	0.0011295	0.001192093
29	0.000937239	0.00094986
30	0.000776743	0.000816345
31	0.000642923	0.000728607
32	0.000531481	0.000588417
33	0.000438788	0.000453949
34	0.000361787	0.000370026
35	0.000297904	0.000278473
36	0.000244971	0.00023365
37	0.000201169	0.000182152
38	0.000164972	0.000150681
39	0.000135099	0.000119209
40	0.00011048	9.63211E-05
41	9.02185E-05	9.15527E-05
42	7.35667E-05	6.10352E-05
43	5.9901E-05	6.96182E-05
44	4.87021E-05	4.86374E-05
45	3.95381E-05	4.3869E-05
46	3.20504E-05	3.62396E-05
47	2.59414E-05	1.81198E-05
48	2.09647E-05	2.09808E-05
49	1.69167E-05	1.33514E-05
50	1.36291E-05	6.67572E-06

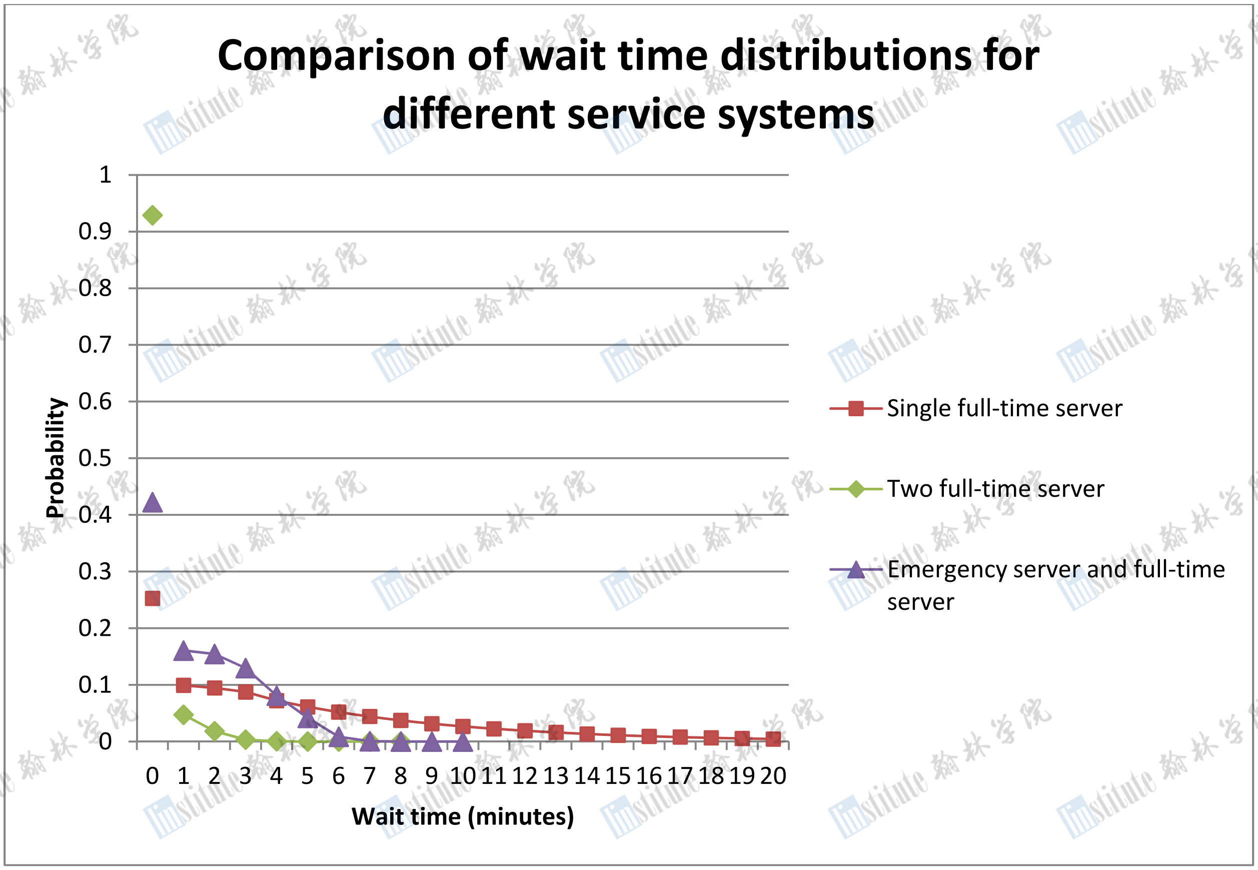

Figure 3: A comparison of the experimental wait time distribution for three different service systems: a single full-time server, two full-time servers, and an emergency server with a full-time server. The setup with two full-time servers clearly minimizes the amount of time customers have to wait in line, but its lower efficiency compared to the emergency server setup makes it a worse strategy for management to adopt.

Appendix C

Below is the complete python script used to simulate the proceedings of the bank.

```
# -*- coding: utf-8 -*-
from random import choice
import numpy as np
import collections
import csv
import os
from itertools import izip_longest


(nservers,  nemergency,  eenterqsize,  etranstime,  eexitqsize)  =
(1,0,0,0,0)

def runday():
    arrivals=  [0]*10  +  [1]*15  +  [2]*10  +  [3]*35  +  [4]*25  +
[5]*05
    services= [1]*25 + [2]*20 + [3]*40 + [4]*15
    customerdata=  [(0,choice(services))]  +  [(choice(arrivals),
choice(services)) for i in range(149)]
    arrivaltimes= [0] + np.cumsum([c[0] for c in customerdata])

    customers  =  collections.deque([(c[0],c[1],arrivaltimes[i])
for i, c in enumerate(customerdata)])

    waiting=collections.deque()


    servers=[(False,0,False)]*nservers
    emergency=nemergency

    emergencyTransitionTime=etranstime
    emergencyEnterQueueSize=eenterqsize
    emergencyExitQueueSize=eexitqsize

    time=0


    dataWaits=[]
    dataQueueLen=[]
    dataTimeEmergency=[]
    dataEmergencyAdded=[]
    dataServerIdle=[]
    while True:
        while len(customers)>0 and customers[0][2]<=time:
            waiting.append(customers.popleft())
        servers=[(active,lasts-1,isE) for active,lasts,isE in
servers]
        servers=[(active,s,isE) if s!=0 else (False,0,isE) for
active,s,isE in servers]

        curDataTimeEmercency=0
        curDataServerIdle=0
        curDataEmergencyAdded=0

        toRemove=[]
        for i,zz in enumerate(servers):
            (active,s,isEmergency) = zz
            if isEmergency:
                curDataTimeEmercency+=1
            if not active:
                if isEmergency and
len(waiting)<=emergencyExitQueueSize:
                    toRemove+=[i]
                    continue
                if len(waiting)>0:
                    (a,nexts,entrytime)=waiting.popleft()
                    dataWaits+= [time-entrytime]
                    servers[i]=(True,nexts,isEmergency)
                else:
                    curDataServerIdle+=1
        for i in reversed(toRemove):
            emergency+=1
            del servers[i]
        if emergency>0 and
len(waiting)>=emergencyEnterQueueSize:
            emergency-=1
            curDataEmergencyAdded+=1
            servers.append((True,emergencyTransitionTime,True))

        dataQueueLen+= [len(waiting)]
        dataTimeEmergency+= [curDataTimeEmercency]
        dataEmergencyAdded+=[curDataEmergencyAdded]
        dataServerIdle+= [curDataServerIdle]
        #print([len(customers),len(waiting),servers])
        if len(waiting)==0 and len(customers)==0 and all([not
active for active,t,isE in servers]):
            break
        time+=1
    return ((dataWaits), mean(dataQueueLen),
sum(dataServerIdle), sum(dataEmergencyAdded),
sum(dataTimeEmergency),time)
```

```
datas=[];
for n in range(10000):
    #print(n)
    if mod(n,1000) == 0:
        print(n/1000)
    datas+=[runday()]

# datas now contains all data from these trials.
```

Appendix D: Sensitivity Analysis for Data and Fluctuations

In order to test how sensitive our model is to changes in data, we changed the probabilities given to us by small and large amounts. The following tables show the results of either slight or great changes to either the times of arrivals or service times.

Greatly increased arrival rate:

Time between arrival (min.)	0	1	2	3	4	5
Probability	0.17	0.33	0.29	0.10	0.06	0.05

	Mean customer wait time (min.)	Mean queue length (customers)	Emergency Server Work Time (min.)	Server idle time (min.)
Single server:	57.05992	11.40478	n/a	2.11110
Single + emergency:	2.67963	1.56315	141.31050	16.41240

Greatly decreased arrival rate:

Time between arrival (min.)	0	1	2	3	4	5
Probability	0.04	0.04	0.09	0.14	0.42	0.27

	Mean customer wait time (min.)	Mean queue length (customers)	Emergency Server Work Time (min.)	Server idle time (min.)
Single server:	0.47679	0.13087	n/a	183.55140
Single + emergency:	0.44185	0.12114	1.76800	184.76850

Slightly increased arrival rate:

Time between arrival (min.)	0	1	2	3	4	5
Probability	0.11	0.16	0.11	0.36	0.26	0

	Mean customer wait time (min.)	Mean queue length (customers)	Emergency Server Work Time (min.)	Server idle time (min.)
Single server:	8.24167	3.20595	n/a	20.56580
Single + emergency:	1.61710	0.64559	48.28230	87.72130

Slightly decreased arrival rate:

Time between arrival (min.)	0	1	2	3	4	5
Probability	0.09	0.14	0.09	0.34	0.24	0.10

	Mean customer wait time (min.)	Mean queue length (customers)	Emergency Server Work Time (min.)	Server idle time (min.)
Single server:	3.13214	1.12310	n/a	56.78220
Single + emergency:	1.31758	0.47106	29.31460	76.39490

Greatly increased service rate:

Service Time (min.)	1	2	3	4
Probability	0.28	0.43	0.10	0.19

	Mean customer wait time (min.)	Mean queue length (customers)	Emergency Server Work Time (min.)	Server idle time (min.)
Single server:	2.36801	0.89845	n/a	70.89160
Single + emergency:	1.13919	0.43058	25.19650	87.72130

Greatly decreased service rate:

Service Time (min.)	1	2	3	4
Probability	0.28	0.43	0.10	0.19

	Mean customer wait time (min.)	Mean queue length (customers)	Emergency Server Work Time (min.)	Server idle time (min.)
Single server:	36.26800	11.56639	n/a	2.97420
Single + emergency:	2.43217	0.91092	102.77920	21.63090

Slightly increased service rate:

Service Time (min.)	1	2	3	4
Probability	0.28	0.43	0.10	0.19

	Mean customer wait time (min.)	Mean queue length (customers)	Emergency Server Work Time (min.)	Server idle time (min.)
Single server:	3.99738	1.50759	n/a	44.77870
Single + emergency:	1.38542	0.52295	34.94580	67.43670

Slightly decreased service rate:

Service Time (min.)	1	2	3	4
Probability	0.28	0.43	0.10	0.19

	Mean customer wait time (min.)	Mean queue length (customers)	Emergency Server Work Time (min.)	Server idle time (min.)
Single server:	5.99814	2.23645	n/a	29.99890
Single + emergency:	1.55318	0.58562	42.60380	56.26230

DV
MD Consulting & Analysis

Control Number: 4100
14 Maniidon Dr.
Middletown, CA 17163
Phone:(555) 555-3141
Fax:(555) 555-5926

November 17, 2013

Mr. Kevin Banks
3141 Banker Ave.
Middletown, CA 17109

Dear Mr. Kevin Banks,

We have completed the statistical analysis you requested on the current service system implemented by your bank. Our results show that the present setup already meets the desired upper limit on the average number of customers waiting to be served at any given time by keeping the size of the line to an average of 1.8 customers, slightly less than the desired average of 2. However, in regards to the average time a customer must wait before being served, the current bank service system proves unsatisfactory. The desired average wait time is 2 minutes or less, but the actual average wait time is around 4.9 minutes. In order to improve on your current system, we tested two possible solutions. The first involves hiring another new full-time dedicated teller (bringing the total number of tellers to two), while the second involves keeping the current full-time teller and adding an “emergency” teller to provide services to customers when three or more people are waiting in line. Both options will bring the desired average customer waiting time down to the target value, but if the second solution can be implemented, it will also maintain a low idle time for workers and higher worker productivity.

Using two tellers instead of one would greatly improve customer satisfaction. Our simulations indicate that hiring another teller would reduce the average customer wait time to about 6 seconds and the average number of waiting customers to 0.04, both of which are significant improvements to the current system and will meet the outlined goals. However, with this system, each worker would be idle a staggering 54% of the time. While this method would be successful in greatly improving customer satisfaction, it may be undesirable due to the immense decrease in worker efficiency.

A more efficient method would be to add a secondary teller only when too many people are waiting in line. We suggest that you designate an “emergency” teller and, when there are 3 or more people waiting, instruct them to set aside their current work and begin serving customers. Once the line of customers is empty, this employee can leave once again to fulfill other obligations. Our results show that, assuming it takes the employee 2 minutes to transition between these two tasks, this method reduces the average customer wait time to only 1.5 minutes and the average line length to 0.6 customers, both well within the desired ranges. Additionally, the total time tellers will spend idle in a given day would only be 60 minutes on average, considerably less than the total idle time when using two full-time tellers. This system would not only improve customer service quality but also minimize the amount of wasted time by idle tellers. Furthermore, we determined that this method has the capability to deal with unpredicted increases in arrivals and service times without greatly increasing customer waiting times or queue size.

After testing both methods, our final recommendation is to add an emergency teller to your current system. This will maximize the efficiency of your business as well as increase service flow. Thank you for choosing DVMD to provide an accurate statistical analysis of your situation. If you need further assistance in implementing this new system or would like us to provide a more detailed analysis of your current system, we would be happy to assist at the previously agreed-upon rates. We hope that after applying this new bank service system, you will see your desired improvements in overall customer satisfaction.

Sincerely,

David Davis
DVMD Consulting