Risk Mitigation at Call Centers

by

Viviana Nieto Valencia
Bachelor of Science, Industrial Engineering, Icesi University, 2014

and

Jin Li
Master of Science, Operations Management, Cranfield University, 2013

SUBMITTED TO THE PROGRAM IN SUPPLY CHAIN MANAGEMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE IN SUPPLY CHAIN MANAGEMENT

AT THE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2018

Signature of Author..
Viviana Nieto Valencia
Department of Supply Chain Management
May 11, 2018

Signature of Author..
Jin Li
Department of Supply Chain Management
May 11, 2018

Certified by..
Dr. Josué Velázquez-Martínez
Executive Director, Supply Chain Management Blended Program
Capstone Advisor

Certified by..
Dr. Cansu Tayaksi
Postdoctoral Associate – Center for Transportation and Logistics
Capstone Co-Advisor

Accepted by...
Dr. Yossi Sheffi
Director, Center for Transportation and Logistics
Elisha Gray II Professor of Engineering Systems
Professor, Civil and Environmental Engineering

Risk Mitigation at Call Centers

by

Viviana Nieto Valencia

and

Jin Li

Submitted to the Program in Supply Chain Management
on May 11, 2018 in Partial Fulfillment of the
Requirements for the Degree of Master of Applied Science in Supply Chain Management

ABSTRACT

Climate catastrophes (i.e. tornadoes, hail, hurricanes, etc.) have a significant economic and operational impact on the operation of call centers. It was found that catastrophe events such as hurricanes critically impact the operation of the affected location for a period of two months after the hurricane has occurred. A sudden increase of demand affects the service level agreement Company X has with its customers due to a shortage of labor resources to attend the inbound calls until the process stabilizes and the location can achieve an adequate service level. Can a company leverage resources from a network of call centers to support impacted locations during a disruptive climate catastrophe event? This study focuses on the development of a call rerouting model. The problem was divided into four main parts: (i) Data preprocessing, (ii) Demand analysis with the use of exponential smoothing, (iii) Capacity analysis using queueing theory and, (iv) Determination of locations to deviate the inbound calls to with the use of a Mixed Integer Linear Programming Model (MILP). In conclusion, the project defines a framework for the company to balance resources during high pressure situations, which can be applied to different types of disruptions in the inbound calls process.

Capstone Advisor: Dr. Josué Velázquez-Martínez

Title: Executive Director, Supply Chain Management Blended Program

Capstone Co-Advisor Dr. Cansu Tayaksi
Title: Postdoctoral Associate

TABLE OF CONTENTS

List Of Figures

LIST OF TABLES

ABBREVIATIONS

Abbreviation	Explanation
CPI	Consumer Price Index
CSR	Customer Service Representative
CST	Central Time Zone
CV	Coefficient of Variation
DTMF	Dual Tone Multi Frequency
GLPK	GNU Linear Programming Kit
IQR	Interquartile Range
IVR	Interactive Voice Response
LF	Lower Fence
MILP	Mixed Integer Linear Programming
NCEI	National Centers for Environmental Information
NOAA	National Oceanic and Atmospheric Administration
RMSE	Root Mean Squared Error
s	Seconds
SL	Service Level
SLA	Service Level Agreement
UF	Upper Fence
α	Alpha
μ	Mean

1. INTRODUCTION

How can a company leverage resources from a network of call centers to accommodate during a disruption, such as a climate catastrophe event? In recent years, there has been an increase in the interest, research and study of the frequency and duration of catastrophe events, specifically those that are climate related.

Since 1980, the United States has experienced 218 weather and climate disasters. In 2017, across the U.S. there were 15 weather and climate events that resulted in material and financial losses that exceeded $1 billion each. These events were composed of droughts, flooding, severe freezing, severe storms, tropical cyclones and wildfires (NOAA-NCEI, 2017).

Between 1980 to 2016 the annual average was 5.5 events (CPI-adjusted). Although the annual average from 2012 to 2016 was 10.6 events (CPI-adjusted) with an increase of around 50% in 2017 (15 climate disasters as of October 6th 2017).

Figure 1-1 was adapted from the NOAA National Centers for Environmental Information (NCEI) and it shows areas that were critically affected by climate events during 2017, giving an understanding of the ample geographic impact and critical disruptions catastrophe events can cause to different sectors.

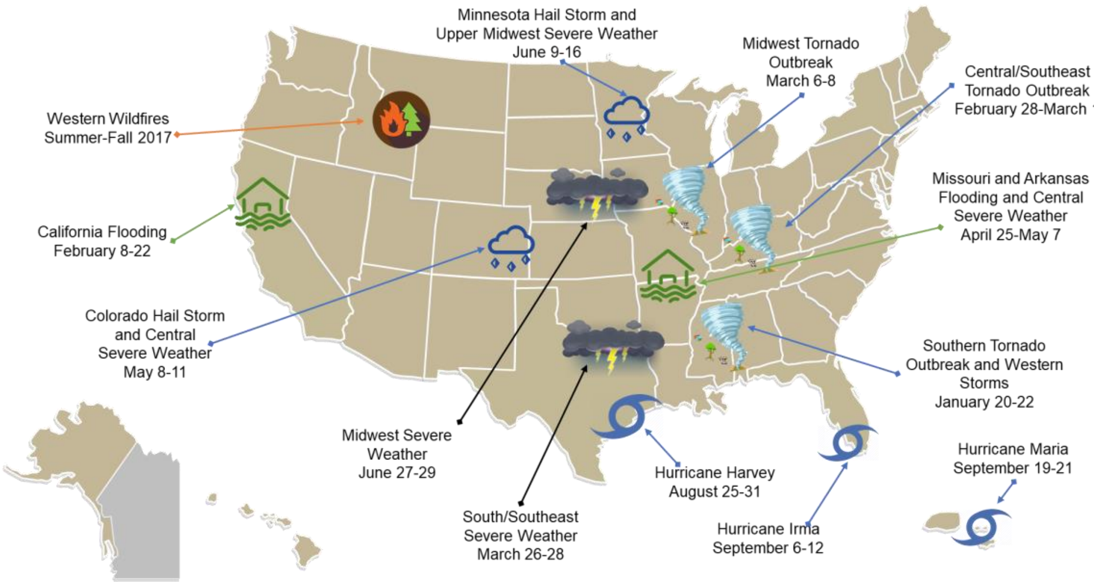

Figure 1-1. Image adapted from 2017 Weather and Climate Disasters in the US

(https://www.ncdc.noaa.gov/billions/)

Climate catastrophes (i.e., tornadoes, hail, hurricanes, etc.) have a significant social and economic impact on the operation of various businesses. These natural disasters typically cause damages that range in the millions of dollars, as well as, leaving a trail of distressed citizens and disrupted industries looking to recover as soon as possible. Unfortunately, there has not been research focusing on the direct impacts of climate catastrophe events on call centers.

Climate catastrophes can cause major disruptions to call center operations including but not limited to completely shutting them down, only allowing intermittent service, or the opposite, resulting in large increases of incoming calls. This project focuses on mitigating impacts due to sudden increases of inbound calls to the call center system that can directly influence customer decisions on conducting business with Company X.

Company X manages an operation that helps sellers connect with buyers of product A. There are over 150 physical locations (warehouses) across the US, where Company X conducts storage, distribution and call center operations relating to the transfer of product A. During a typical day, the call center operation handles inbound and outbound calls related to: (i) coordination of the pickup of product A to be delivered to one of the warehouses, and (ii) inbound calls serving the buyers that use the ecommerce site to make purchases.

Company X has a target Service Level Agreement (SLA) to respond to incoming calls in under 60 seconds, based on internal studies and widely accepted standards for example those analyzed by (Batt, Holman & Holtgrewe, 2007). However, some call centers fail to meet the SLA due to significant increases in the rate of incoming calls during climate catastrophes. This in turn leads to an abandonment rate greater than 15% and Company X's goal is to have abandonment rates below 5% at each call center. The key question that this study aims to answer is: How to temporarily utilize/borrow available staff from unaffected call centers to affected call centers during climate catastrophe events?

Currently, there is not a strategic decision-making process in place to reroute inbound calls to unaffected call centers with available capacity (man-hours). In this study, Queuing Theory and Optimization techniques are applied to develop a model that can select the appropriate call centers to divert inbound call queues to. This study is focused on the acute inbound call increases caused by destructive climate catastrophe events such as hurricanes, since there is not a risk mitigation process in place.

2. Literature Review

Most of the research regarding call centers has been done on staffing and schedule optimization without integrating the risk variable in the model. To leverage the resources in the organization for call rerouting with the use of standard best practices, the literature research was concentrated on what has been done in two different areas: (i) Call Centers and (ii) Catastrophe

Events Risk Mitigation to finalize with an optimization model proposal that mitigates risk at call centers.

2.1. Call Centers

Over the last two decades, call centers have experienced significant growth and adoption by small and medium businesses. It has become an effective method for expansion that helps in the provision of services and an integral part of a business in different industry sectors.

A call center can be dedicated to outbound calls and/or inbound calls. However, this research only studies incoming calls to Company X's owned and operated call center. Inbound calls have a higher cost impact when compared to outbound calls for Company X. The likelihood of losing a customer from not answering an inbound call is higher than from not making an outbound call. Therefore, it was determined to only focus on inbound calls for this study. Outbound calls were outside scope for this project, although it can serve as an area of future research.

Inbound call centers can serve a broad range of customers by having various service offerings and can be automated using different technologies. Examples of the latest technologies include Voicemail, dealer locator and Interactive Voice Response (IVR) Systems (Batt, Holman & Holtgrewe, 2007).

Interactive Voice Response (IVR) Systems have been developed with the aim to better serve customers, by giving them the option of interacting with an automated machine using voice or Dual Tone Multi Frequency (DTMF) keypad tones (Thirumaran M. et al., 2015). Company X uses a single level IVR system for inbound calls that greets the callers and assists them in reaching the most appropriate agent for their needs with five different options (queues) they can choose from. Company X does this by responding to the caller input via the telephone keypad. For example, a caller that selects Spanish in the IVR will be routed to the Spanish queue where it will be served by a Spanish-speaking agent. This type of service helps increase efficiency and first call resolution in the organization.

The phone provider offers the functionality of deviating the IVR queues calls from one location to other location(s) during specific time frames. For example, the calls placed to Queue 1 of location 24 8:00 AM to 10:00 AM EST can be deviated to Queue 1 at location 34 from 8:00 AM to 10:00 AM EST between May 01st and May 15th of 2018, after this period the calls will be received at the location where the calls are placed to. The service of call deviation will be used to employ the results of the risk mitigation model.

2.1.1. Call Center Performance

Short holding times are essential to accomplish client expectation fulfillment. Achievement levels are measured by wait times, accessibility of service to the caller, and call abandonment. These performance metrics can be enhanced if call center leaders hire additional operators. However, most of the maintenance costs at call centers are due to the staffing, in fact it is approximately 60 to 70 percent. Call center managers have the task of balancing between meeting performance metrics and staying within budget (Stolletz, R., 2003).

The measures of variables such as: waiting time, process time, interarrival time (time between the start time of two events) for each type of incoming call or service provided can be analyzed to develop models that achieve the call center performance goals without adding unnecessary costs to the operation of the center.

Call arrivals and call handling times have high variability at inbound call centers due to the influence of customer behavior. Therefore, performance metrics can be obtained with optimization techniques that use queueing models, for example we used queuing theory to find out the capacity of calls a location can handle for a specific queue and timeframe by taking into consideration the established performance metrics by Company X. These models differ due to internal and external features, for example, type of customers, differently trained operator groups or customers' willingness to wait.

2.1.2. CALL CENTER MODELING

Typically, the objective of a call center is to obtain a specific minimum level of service at minimum cost. The Service Level (SL) can consist of multiple criteria related to the quality of service or the time before a customer service representative (CSR) is reached. Although there is a relationship between these two types of measurements, this study focuses on a service level measurement where $\alpha\%$ of customers must have a waiting time shorter than b seconds. The industry standard for this metric is that 80% of the customers must wait less than 20 seconds in any queue, supported by research done by Whitt (2005) which shows that the distribution of the abandoned calls has a considerable impact on the performance of the inbound call center.

Different objective functions and optimization approaches can be used based on the problem that needs to be solved regarding the operation of the call center. The basic model for an inbound call center is the Erlang C or $M|M|s$ model, and it is the starting point of the staffing management. This model considers an average of λ calls arriving per time unit, average holding time β, and s agents. If $\lambda\beta < s$, then there is a probability than an arbitrary arriving call gets blocked. And if a call gets blocked, the waiting time has the attributes of an exponential distribution with parameter $s\mu - \lambda$, where $\mu = \beta^{-1}$.

When applying the Erlang C formula, different scenarios must be evaluated to accomplish the desired SL. Among the features that need careful consideration are: arrival rates of customer calls, agents' availability, distribution of the service times, number of lines of service, and abandoned calls.

Arrivals in a call center are described by a Poisson distribution with a varying rate, and the number of agents can vary over time as well. The Erlang formula is valid under stationarity which assumes non-varying parameters. Therefore, the service times were divided into various intervals, (often 15 minutes intervals are taken) that approximate the requirement of stationarity. Service times at call centers are usually assumed to be exponential and this was tested with a Chi-squared test to better define the model.

Modeling abandonments is pivotal, unless the service level is high enough that abandonments infrequently happen. Modeling abandonments likewise makes the model more vigorous: even in over-burden circumstances, (i.e., $\lambda > s$) where the SL is well characterized. Typically, the customers' willingness to hold is thought to be exponential, and the subsequent model is referred to as the Erlang A model (Koole G., 2004). Thus, queueing theory was used to model capacity based on maximum abandonment rate and acceptable waiting times based on SLA.

2.2. Catastrophe Event Risk Mitigation

Large businesses have long engaged in risk assessment and mitigation as a core business practice, nevertheless this practice has not been acquired at a large scale by small and medium organizations. Uncertainty about future conditions makes it difficult to know what the goal of mitigation efforts should be.

It is important to distinguish between risk, which characterizes situations in which probabilities of a random event are perfectly known, from the broader notion of uncertainty, which characterizes situations in which some events do not have an obvious, unanimously agreed upon probability assignment (Ghirardato et al., 2004).

A robust risk management approach to deal with the problem of catastrophic climate change that incorporates both risk and model uncertainty was proposed by Berger L. et. al. (2017). The application of recently developed tools from decision theory help them encounter the best strategies in the case of climate change with deep uncertainty.

3. Methodology

To create a generalized framework for call rerouting, we partitioned the problem into four main parts: (i) Data preprocessing, (ii) Demand analysis with the use of exponential smoothing, (iii) Capacity analysis utilizing queueing theory, and (iv) Optimization model using Mixed Integer Linear Programming (MILP) to define the locations to deviate the calls to during and after a catastrophe event, based on Company X’s requirements.

The tools that were used to analyze and provide an optimization solution were Tableau for initial data analysis and R for model definition and results. These tools were chosen because they provided a smoother integration with Company X’s databases and the possibility of creation of an interactive dashboard to run the optimization with minimum dedication of personnel for code maintenance.

3.1 DATA

The data was provided by Company X in a large .CSV file. The data included around five million records of inbound calls between November 2016 and November 2017 that were previously anonymized by Company X, formatted as shown in Figure 3-1.

	location_id	queue	dateCall	arrivalTime	serviceTime	endTime	agentCode	holdTime	durationTime	totalTime	loc_time_off
1	348	Queue 2	2017-09-26	04:35:16 PM	04:35:22 PM	04:38:22 PM	483	6	174	180	2
2	349	Queue 2	2016-11-28	03:43:07 PM	03:43:16 PM	03:45:43 PM	482	9	138	147	2
3	354	Queue 2	2017-01-12	10:58:36 AM	10:58:39 AM	11:01:35 AM	562	3	173	176	3
4	361	Queue 2	2017-10-25	08:20:39 AM	08:20:44 AM	08:22:41 AM	487	5	112	117	2
5	367	Queue 2	2016-12-09	03:47:15 PM	03:47:19 PM	03:48:06 PM	481	4	43	47	3
6	441	Queue 2	2017-09-19	11:27:39 AM	11:27:53 AM	11:29:59 AM	481	14	112	126	3

Figure 3-1. Image of Sample .CSV File in R dataframe

The columns in the input data file represented the location id, queue type, date, arrival time, start service time, end time, agent code, holding time, processing time, total time and location time zone difference. Each column is described below:

- The **location ID** is a unique number assigned to a specific call center location.
- The **queue type** is a unique identifier of the five queues Company X uses in their inbound call system. This attribute is standardized throughout the organization and the queues are independent, which means an agent assigned to Queue 2 is unable to pick up calls from any other queue (e.g., Queue 3)
- The **date** field refers to the date the call was received.
- The **arrival time** is the local time stamp when the call entered the system.
- The **service time** is the local time stamp when the caller started to be served by a Customer Service Representative (CSR).

- The **end time** is the local time stamp when the call is finalized due to a resolution to the customer's call.
- The **agent code** is the identification number for the CSR that answers the call at the call center.
- The **holding time** is the amount of time in seconds the caller waits in the system before the call is answered by a CSR.
- The **processing time** is the amount of time it takes to give a resolution to the call, in seconds. It is measured from the moment the CSR answers the call until it is finalized by the CSR or the caller.
- The **total time** is the total time the caller spends in the system. It is the sum of the holding time and processing time.
- The **location time zone difference** is the number of hours ahead of time a location is in comparison to the Pacific Time Zone (PST), that was used as reference. Therefore, if a location is in PST the value of this feature is zero and if a location is in Central Time (CST), the value of this feature is two.

In addition to the previous terms, the definitions below are used on the following sections:

- **Abandoned calls** are the calls that were initiated to the call center, however ended before any conversation occurred.
- **Abandon rate** is the percentage of inbound calls made to the call center that were abandoned by the caller before speaking to a CSR or agent.
- **Timeslot** refers to the segments of one-hour that the call arrived at the system. The segments start being counted from 8:00 AM until 5:00 PM local time, for a total of nine segments during the day. For example, if the call arrived at the call system at

9:10 AM, it is classified as part of timeslot two (2) at that specific location. Table 3-1 illustrates the relationship between timeslot k and the period of time they are associated with.

Table 3-1. Relationship between timeslot k and period of time

Timeslot (k)	Start Time*	End Time*
1	8:00 AM	9:00 AM
2	9:00 AM	10:00 AM
3	10:00 AM	11:00 AM
4	11:00 AM	12:00 PM
5	12:00 PM	1:00 PM
6	1:00 PM	2:00 PM
7	2:00 PM	3:00 PM
8	3:00 PM	4:00 PM
9	4:00 PM	5:00 PM

* Start Time is inclusive and End Time is exclusive for classification purposes

3.2 PACKAGES USED

The software that was chosen to work with was the R programming language. Because R was already being used at Company X for various models, therefore providing an easy integration with the current infrastructure in addition to a handful of free packages that were used throughout this project. At the beginning of the program, the R packages mentioned in Table 3-2 were imported. These packages were necessary to support functions such as: MILP optimization, forecasting, statistical analysis, mathematical functions, queueing theory analysis, data handling, among others.

Table 3-2. Packages Included in R Program

Package	Function
data.table	Statistical Analysis
Rglpk	Optimization Engine
forecast	Forecasting Analysis
queuecomputer	Queueing Theory Analysis
raster	Statistical Analysis
lubridate	Data Handling
ISOweek	Date manipulation and formatting
timeDate	Date manipulation and formatting

3.3 DATA PREPROCESSING

The initial data included abnormal entries or outliers that were taken out to increase reliability of the proposed solution. The abnormal data was related to problems in the phone system software that generated inconsistent values. Specifically, two types of records were removed from the initial data provided by Company X, based on business knowledge of the process and feasible time durations to be included in the model. The data entries that were removed are described below:

a) Records with negative holding or processing time. A total of four entries were found under this scenario.

b) Records with holding or processing time greater than 2 hours. A total of 741 entries were removed under this condition.

The percentage of records removed in the preprocessing stage was 0.014% of the total records that were given by Company X. Figure 3-2 shows a representation of a bar-chart of the number of incoming calls for each of the five queues used by Company X for its inbound call system during the period of November 2016 until November 2017. Queue 1 constitutes approximately 57% of the total records, therefore being the queue that demands more capacity across all locations. Although, the percentage of calls associated with each queue does not affect the results obtained in later steps since all the queues are independent from each other.

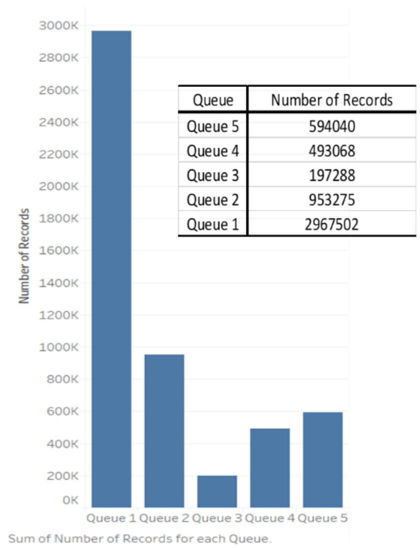

Figure 3-2. Bar chart with number of calls per queue

Table 3-3 and Figure 3-3 illustrate the dispersion of holding time. From the results, we found that more than 99% of the calls were picked up within 1,200 seconds (20 minutes). However, according to expert experience from the partner firm, it is possible that some customers wait in the queue for two hours when the rate of incoming calls is higher than the number of CSRs available to serve them. Thus, calls with holding time from 0 to 7,200 seconds (2 hours) were regarded as feasible data. Eighty-six records were removed from the initial data due to holding time values above two hours that were related to software malfunction.

Table 3-3. Analysis of holding time

Holding time bin (seconds)	Frequency	Relative Frequency	Cumulative Relative Frequency
0-600	328262	96.37%	96.35%
601-1200	9211	2.70%	99.05%
1201-1800	2004	0.59%	99.64%
1801-2400	668	0.20%	99.83%
2401-3000	220	0.06%	99.90%
3001-3600	123	0.04%	99.93%
3601-7200	140	0.04%	99.97%

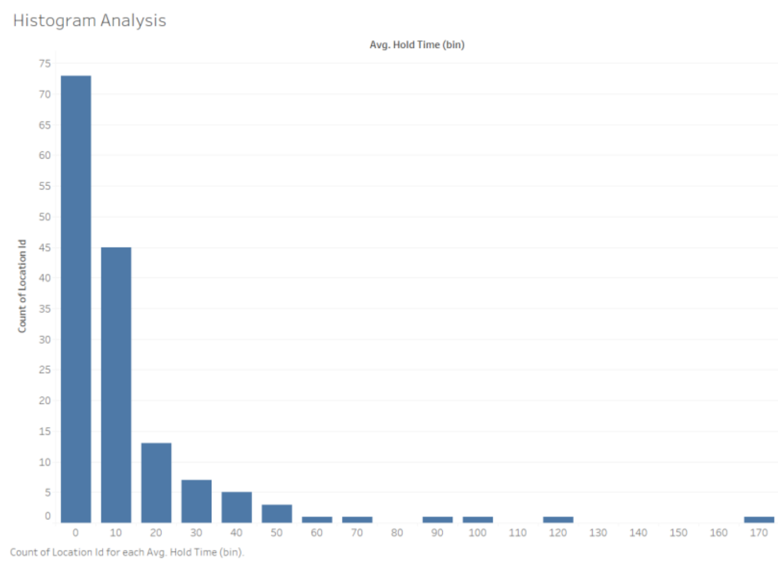

Figure 3-3. Histogram holding time across all locations

Figure 3-4 and Table 3-4 describe the dispersion of duration time. Similar to the analysis results of holding time, 99.98% of the calls had a duration time of less than 7200 seconds (2 hours). Records with duration time higher than two hours were removed due to software malfunction to capture the call duration time, a total of 655 records were removed in this case.

Table 3-4. Analysis of duration time

Duration time bin (seconds)	Frequency	Relative Frequency	Cumulative Relative Frequency
0-600	5061162	97.25%	97.23%
601-1200	123646	2.38%	99.61%
1201-1800	14830	0.28%	99.89%
1801-2400	3156	0.06%	99.95%
2401-3000	944	0.02%	99.97%
3001-3600	380	0.01%	99.98%
3601-7200	398	0.01%	99.99%

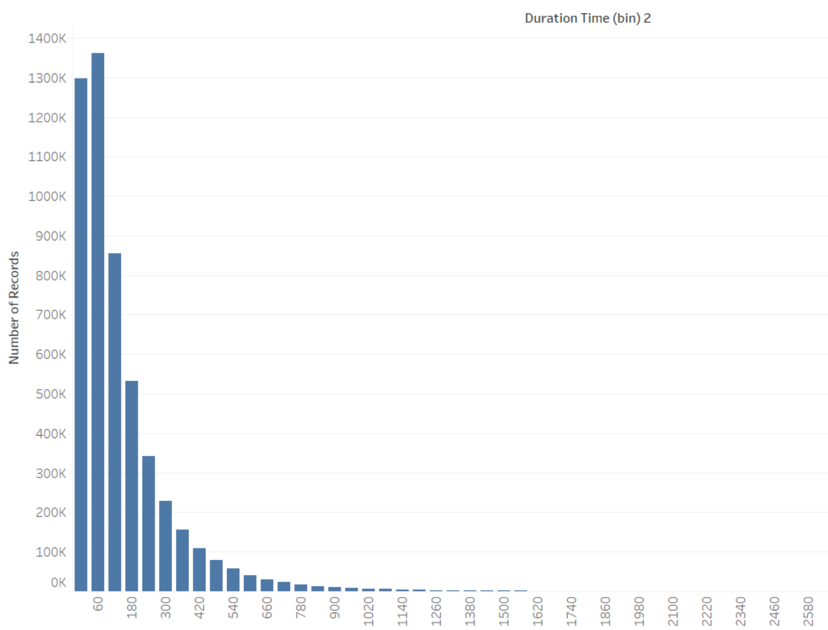

Figure 3-4. Histogram analysis results for duration time

After the data preprocessing stage, there were around five million historical call records that were considered for the development of the project.

3.4 DEMAND ANALYSIS

The demand requirement refers to the number of inbound calls arriving to location "A" in queue *j* during timeslot k, identified with the variable name D_{jk} (number of calls/hour).

Simple Exponential Smoothing (SES) was used to forecast demand for a period of two weeks, using two months of historical data that change dynamically with the model based on the date it will be used. Exponential smoothing models iteratively forecast future values of a regular time series array of values, applying specific weightings to past values of the series. The value of the parameter alpha on the SES was chosen by using the value that minimized the root mean squared error (RMSE) on the test data for the month of September 2017 across all call centers. Therefore, reducing noise in the model by having a more accurate demand forecast.

On the other hand, the likelihood of a Hurricane affecting a location in the United States is higher in coastal regions, based on historical data from Company X and research on the NOAA website. During 2017, the call center with location ID 310 was affected by a Hurricane. Thus, an analysis of the inbound calls' distribution was performed on location ID 310 to find out the demand patterns during and after a hurricane until its stabilized. (See R code in Appendix B – Demand Analysis Code)

The output of this analysis is used later in an interactive dashboard to make recommendations to the model user when defining the percentage of inbound calls demand at location *i* required to reroute.

3.5 CAPACITY ANALYSIS

The key performance indicator for Company X when measuring inbound calls' service performance is to have at least 95% of the calls being answered in less than 60 seconds, hence reducing the rate of abandoned calls by the caller.

Queueing theory was used to establish the capacity bandwidth at the combination of location, queue and timeslot. The capacity bandwidth is the difference between the maximum capacity at location *i* minus the demand at location *i* that was forecasted using simple exponential smoothing. The calculation included restrictions on holding time below 45 seconds, not the 60 seconds on the target KPI, to account for variability in the rate of call arrival. The number of agents that were available to answer calls on the previous two months were considered as the number of parallel servers for the model.

Table 3-5 references to the base nomenclature and Table 3-6 to the performance metrics used to develop the queueing theory model for all locations in Company X.

Table 3-5. Notation adopted in queueing theory model

Notation	Definition	Unit
r_a^{ijk}	Rate of call arrival at location *i* in queue *j* for timeslot *k*	calls/ time
t_a^{ijk}	Mean time between arrivals at location *i* in queue *j* for timeslot *k*	time/call
CV_a^{ijk}	Coefficient of variation of interarrivals at location *i* in queue *j* for timeslot *k*	
m^{ijk}	Number of parallel agents at location *i* in queue *j* for timeslot *k*	
r_p^{ijk}	Rate or capacity at location *i* in queue *j* for timeslot *k*	calls/time
t_p^{ijk}	Mean effective process time at location *i* in queue *j* for timeslot *k*	time/call
CV_p^{ijk}	Coefficient of variation of process time at location *i* in queue *j* for timeslot *k*	

Table 3-6. Performance metrics for queueing theory model

Notation	**Definition**	**Unit**
t_q^{ijk}	Expected waiting time at location *i* in queue *j* for timeslot *k*	time
CT^{ijk}	Expected time in system $(t_q^{ijk} + t_p^{ijk})$ for a call at location *i* in queue *j* for timeslot *k*	time
WIP^{ijk}	Average calls in process at location *i* in queue *j* for timeslot *k*	calls
WIP_q^{ijk}	Average work in process in queue at location *i* in queue *j* for timeslot *k*	calls
u^{ijk}	Utilization of the server $(r_a^{ijk} + r_p^{ijk})$ at location *i* in queue *j* for timeslot *k*	calls/time

An optimization model was developed to maximize the rate of call arrival at location *i* in queue *j* for timeslot *k* (r_a^{ijk}) that could be answered within the desired performance with the available agents *(*m^{ijk}*)* and with an expected waiting time (t_q^{ijk}) of less than 45 seconds to account for uncertainty in the forecasted demand. The package ‘queuecomputer’ and ‘raster’ in R were used to generate the capacity using queueing theory. The R code can be found in Appendix C – Capacity Analysis Code.

Kingman’s or VUT equation with interarrival and process time associated to a general distribution was used to establish the connections between the variation, utilization and processing time at location *i* in queue *j* for timeslot *k*, as it can be seen below:

$$t_q = \left(\frac{CV_a^2 + CV_p^2}{2}\right)\left(\frac{u^{\sqrt{2(m+1)}-1}}{m(1-u)}\right)t_p$$

where,

$$u = \frac{r_a * t_p}{m}$$

3.6 OPTIMIZATION MODEL

To create a base optimization case for the rerouting of calls from a chosen location *i* for each combination of queue *j* and timeslot *k*, a Mixed Integer Linear Programming model was developed using the Rglpk package in R. While the GNU Linear Programming Kit (GLPK) solver was chosen to solve the optimization, any other mixed integer linear programming package could be used to create a similar model. The Rglpk package is a free optimization solver that counts with the option of building MILP models, thus providing Company X with a low-cost solution that could easily be implemented in the dedicated servers they have for the use of R. (See Appendix D – Optimization Model Code)

When establishing the distribution of inbound calls, Company X must take into account that locations cannot process an unlimited number of incoming calls. Their capacities can be limited for a variety of reasons such as shortage of phone lines and staff. Capacity limits must therefore be considered in the modelling of inbound call rerouting. Maximum and minimum bounds were considered in order to account for operational and physical restrictions.

Based on the problem description given in previous sections, the optimization model was defined with the following sets, parameters and variables:

Sets:

- $I=\{1,\ldots,M_i\}$ potential locations,
- $J=\{1,\ldots,N_j\}$ potential queues,

- K={1,…,T_k} potential timeslots,

Parameters:

- c_{ij} fixed cost associated with the platform setup by a telecom engineer to schedule a call rerouting to location *i* in queue *j*, $i \in I$ and $j \in J$
- s_{ijk} fixed cost associated with additional supervisor requirement to manage the Customer Service Representatives (CSRs) for each activated location *i* queue *j* and timeslot *k*, $i \in I$, $j \in J$ and, $k \in K$
- C_{ijk} Maximum idle capacity at location *i* queue *j* and timeslot *k*, $i \in I$, $j \in J$ and, $k \in K$
- D_{jk} Minimum demand in queue *j* and timeslot *k* that needs to be rerouted, $i \in I$, $j \in J$ and, $k \in K$
- Q big number, in this case greater than the number of timeslots *k*

Variables:

- X_{ij} Binary variable that identifies if location *i* was use as a recommendation to reroute calls in any of the timeslots for queue *j*, $i \in I$ and $j \in J$
- Y_{ijk} Binary variable that identifies if the calls that need to be reroute, were sent to location *i* queue *j* during timeslot *k*, $i \in I$, d $j \in J$ and $k \in K$

Using the above definitions, the model was formulated as follows:

$$Min \qquad \sum_{i=1}^{M}\sum_{j=1}^{N} c_{ij} * X_{ij} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{T} s_{ijk} * Y_{ijk} \tag{1}$$

s.t.

$$Y_{ijk} \leq \frac{C_{ijk}}{D_{jk}} \qquad \forall_{ijk} \in I, J, K \tag{2}$$

$$\sum_{i=1}^{M} Y_{ijk} \geq 1 \qquad \forall_{jk} \in J, K \tag{3}$$

$$\sum_{k=1}^{T} (-Q * X_{ij} + Y_{ijk}) \leq 0 \qquad \forall_{ij} \in I, J \tag{4}$$

$$X_{ij}, Y_{ijk} \in \{0,1\} \tag{5}$$

This model minimizes the costs, Eq. (1), with the first term representing the system setup cost by a telecom engineer for location (*i*), queue (*j*). The second term refers to the supervisor costs at location (*i*), queue (*j*) and timeslot (*k*) where the calls are rerouted to. Constraint (2) refers to the idle capacity at location (*i)*, queue (*j)* for timeslot (*k)* where the calls can be rerouted to. Constraint (3) assures that demand for all queues (*j*) and timeslots (*k*) are considered. Another consideration was to link the setup cost of the system and the physical cost in the operation, Eq. (4) is a linking constraint, where Q is a big number, in this case greater than the number of timeslots (*k).* Eq. (5) is the binary constraint for the variables used in this model.

3.6.1 MODEL OUTPUT

The result of the optimization model is a table similar to the one presented in Figure 3-5 with the queues *j* for location "A" on the first column and a list of locations *i* and timeslots *k* the calls are recommended to be sent to in columns two and three respectively. In this case, there were two locations used to deviate the calls for queues one and three and queues two, four and five used one location for the deviation of the calls.

Queue Location "A"	Location to	Timeslots "A"
1	436	c("1", "2", "6", "7")
1	413	c("3", "4", "5", "8", "9")
2	321	c("1", "2", "3", "4", "5", "6", "7", "8", "9")
3	343	c("1", "2", "4", "5", "6", "7", "8")
3	368	c("3", "9")
4	364	c("1", "2", "3", "4", "5", "6", "7", "8", "9")
5	321	c("1", "2", "3", "4", "5", "6", "7", "8", "9")

Figure 3-5. Image with Sample of Optimization Results

4. RESULTS

The results of the inputs to the optimization model such as: (i) final data inputs, (ii) demand forecasting and, (iii) capacity analysis with the use of queueing theory, are addressed in the sections below. The results of the optimization model for three different locations that used the capabilities of Mixed Integer Linear Programming (MILP) as examples to define locations to deviate the calls to during and after a catastrophe event are then provided. All the results were done using Company X’s data.

4.1 COMPREHENSIVE STATISTICAL ANALYSIS

The principal continuous variables that were analyzed throughout this project were holding time, duration time and total time. Total time is the sum of the holding time and processing time. Table 4-1 illustrates the average value (μ), standard deviation (Std) and Coefficient of Variation (CV) related to the parameters described above. The average holding time for all call centers from November 2016 to November 2017 was 10.11 seconds with a standard deviation of 15.61 seconds.

The main parameters were analyzed separately for each location and divided into timeslots of one hour to be able to use the call demand and capacity as a deterministic variable in the MILP model. Due to the high variability on the holding, duration and total time when analyzing all the locations in conjunction.

Table 4-1. Overall statistics across all call centers
(time unit: seconds)

	Mean (μ)	Std (σ)	CV(σ/μ)
Holding time	10.11	15.61	1.5440
Duration time	187.7	302.2	1.6100
Total time	202.0	318.2	1.5752

4.1.1 STATISTICS FOR CALL CENTERS

In the interest of proving the Service Level Agreement the organization was congruent with the overall service in the call center operation, we developed an Interquartile Range (IQR) analysis, also technically called H-spread.

Table 4-2 describes the results of the IQR analysis across all call centers. The IQR is a measure of statistical dispersion, being equal to the difference between the 75th and 25th percentiles, or between the upper and lower quartiles. Furthermore, the Lower Fence (LF) and Upper Fence (UF) are also included in the table to identify the bounds for each parameter that can help on defining outliers.

In this case, calls with holding time greater than 24.5 seconds could be regarded as outliers and it was found these calls were directly related to outages, software malfunction and climate events during week 34 and week 40 of 2017.

Table 4-2. Interquartile Range (IQR) analysis for all call centers

	Q1 (25%)	Q2 (50%)	Q3 (75%)	IQR (Q3-Q1)	Lower Fence (Q1-1.5*IQR)	Upper Fence (Q3+1.5*IQR)
Holding time	4	7	14	7	-6.5	24.5
Duration time	60	116	221	105	97.5	378.5
Total time	75	135	247	112	-93	415

time unit: seconds

To find out the contributing factors for the sudden increase during week 34 to week 41 (around July and August), the 152 call centers were grouped into two categories based on average holding time. Centers with average holding time higher than 24.5 seconds are in group D (27 centers in all) and all others are in group S (125 centers in all).

After filtering the outliers (i.e., data in group D), the average holding time across all weeks during the year of 2017 was lower than 15 seconds as it can be seen in Figure 4-1. Meanwhile, 95% of the calls' holding time were lower than 60 seconds, being in congruence with the Service Level Agreement established by Company X.

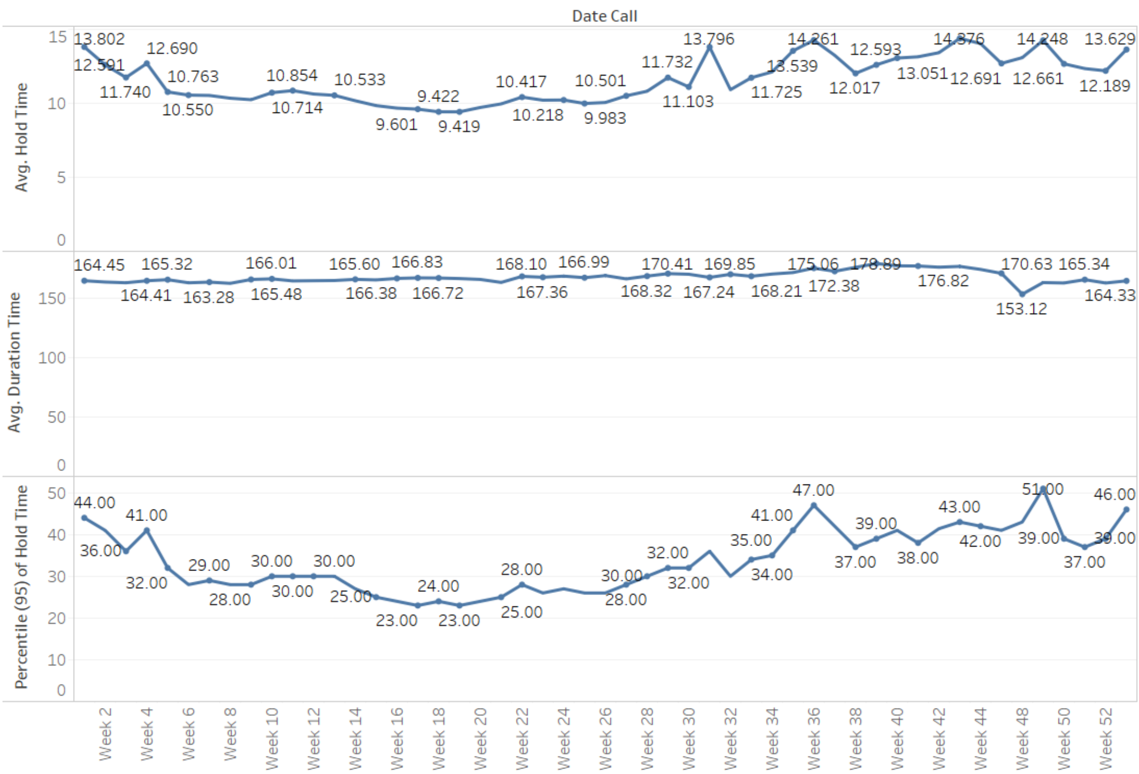

Figure 4-1. Call centers analysis with holding time lower than 24.5 seconds (Group S)

Figure 4-2 describes the average holding time, number of records and average duration time for the centers in group D. The average holding time presented a significant increase between week 34 and week 41 of 2017 (July 17 to September 13). The peak value was 206.5 seconds on week 37 due to two simultaneous hurricanes in two of the Company X's principal locations.

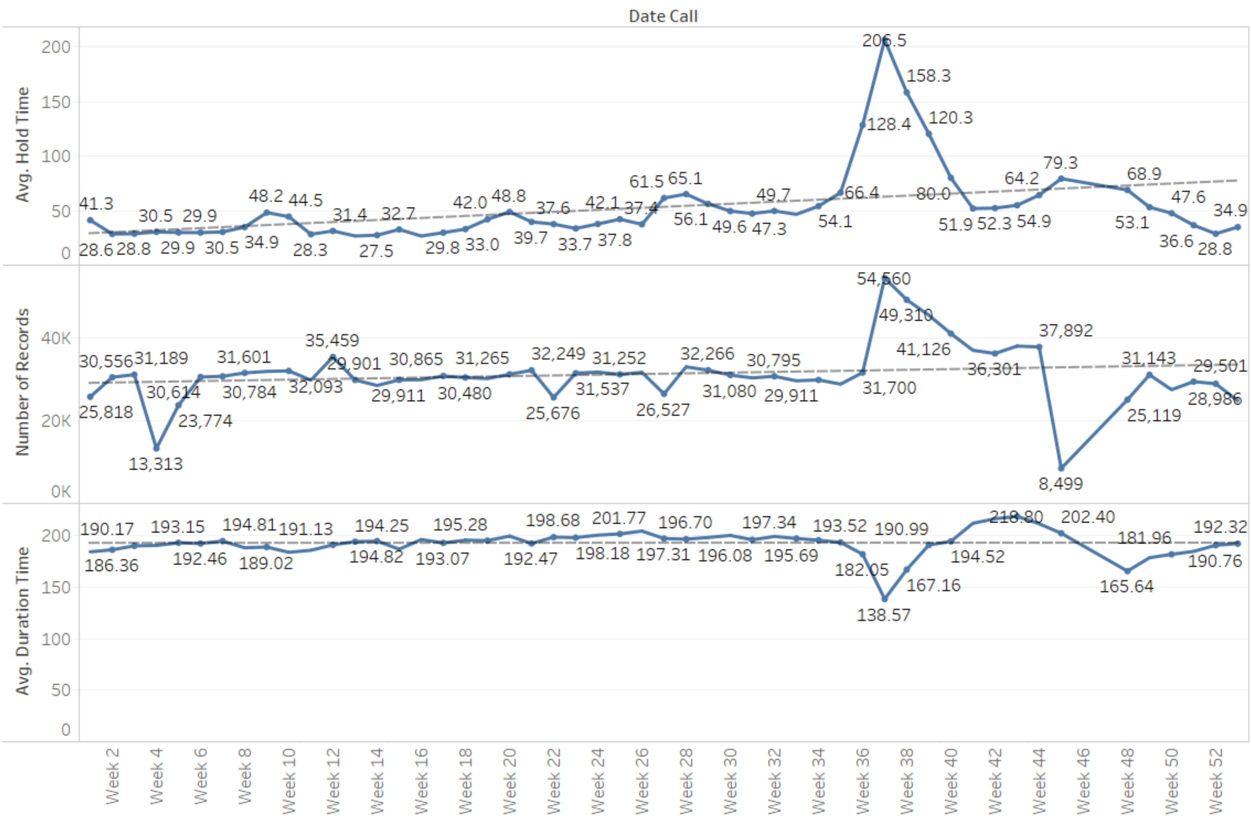

Figure 4-2. Call centers analysis with holding time greater than 24.5seconds (Group D)

Based on the previous analysis, it was found there was a significant fluctuation on the holding and duration time for a period of 8 weeks starting on the first week of July 2017. Figure 4-3 illustrates the relationship between daily average holding time and the call's arrival rate from July to October 2017. There was a positive relationship between the call arrivals and the holding time. Thus, when the call arrival rates were lower than 60 calls/hour, the majority of the holding time was below 50 seconds. In contrast, as the arrival rates were greater than 60 calls/hour, the holding time was greater than 60 seconds.

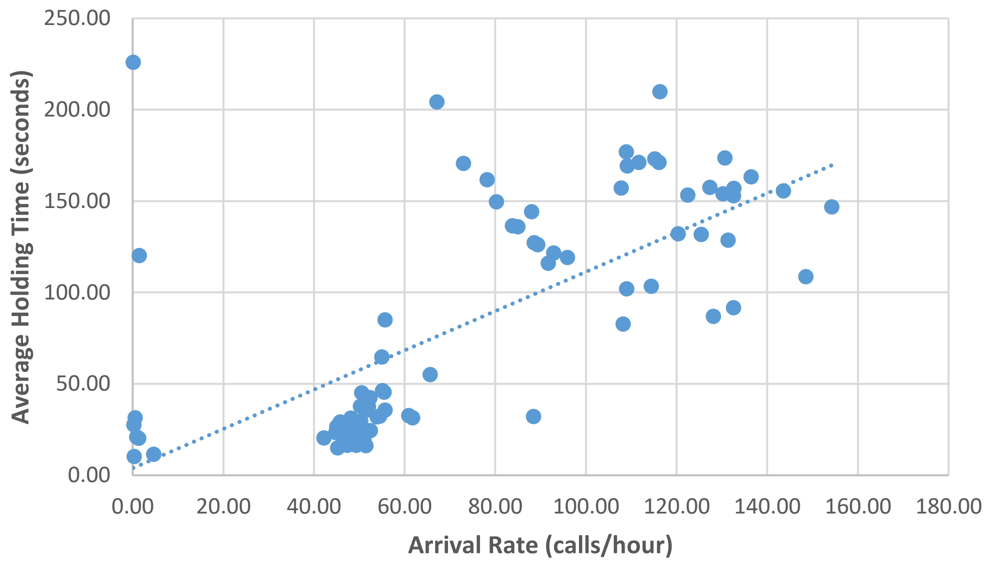

Figure 4-3. Relationship between call arrival rate and holding time

(July to October 2017)

Furthermore, from the relationship between holding time and calls' arrival rate (see Figure 4-3), it was captured that 60 calls/hour is a critical threshold for location 310. When the arrival rate is higher than 60 calls/hour, this call center had a higher likelihood of increasing its holding time to values above 60 seconds.

4.2 Validation of Service Level Agreement

The average holding time graph can be found in Figure 4-4 and it shows that during week 37 of 2017 there was an increase of around 60 seconds in holding time due to the occurrence of two simultaneous climate events in the US (Hurricane Harvey and Hurricane Irma). This was combined with an inefficient system for rerouting of inbound calls placed to the two affected call centers, since there was not an effective system in place to leverage resources across Company X. Additionally, the average duration time decreased in week 37 of 2017 as it can be seen in

Figure 4-5, which could translate in lower customer service from part of the call center personnel to be able to serve as many inbound calls as possible.

Although, when excluding the holding time outlier of week 37, the company had an average holding time of 23.3 seconds with a standard deviation of 10.3 seconds which validates the capabilities of the call centers to answer the inbound calls in less than 60 seconds under normal circumstances as it is stated in Company X's SLA.

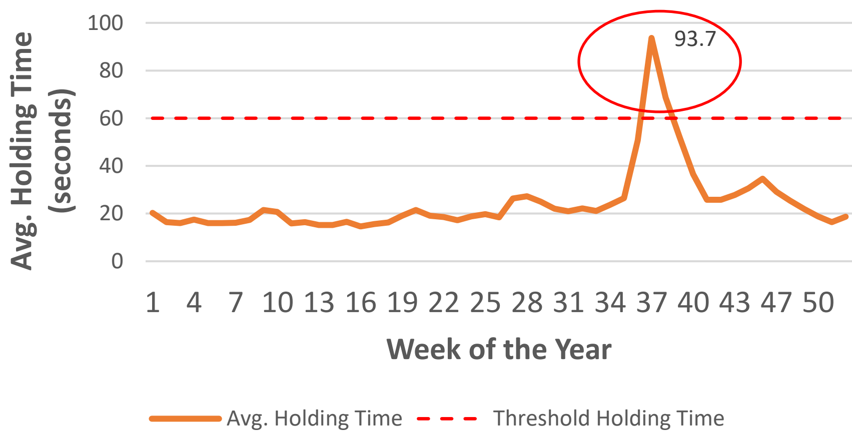

Figure 4-4. Average holding time across all call centers

Data source: Inbound call logs from November 2016 to November 2017

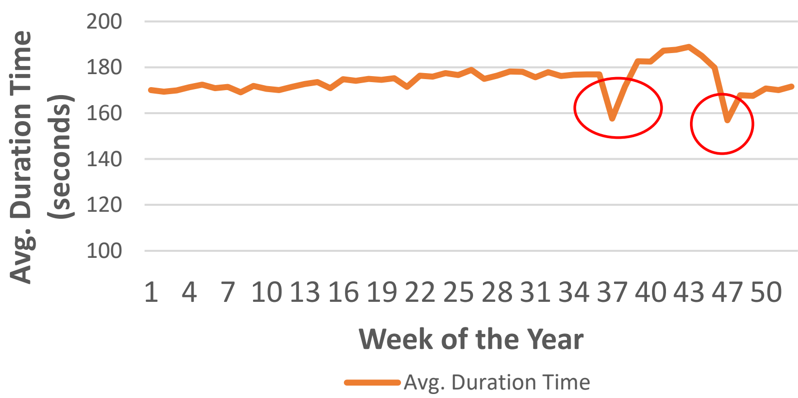

Figure 4-5. Average duration time across all call centers

Data source: Inbound call logs from November 2016 to November 2017

4.3 DEMAND RESULTS

The demand results were divided into three sections: (i) Overall demand analysis, (ii) Demand forecasting that served as input for the optimization model and, (iii) Demand analysis for the affected call center.

4.3.1 OVERALL DEMAND ANALYSIS

An exponential equation can be used to describe how the holding time behaves overtime (w = week of the year). The data related to climate events was excluded when establishing the trendline. In this analysis an alpha value of 0.05 was used with a null hypothesis expressing there is no trend for the holding time as a function of the week of the year. The statistical values obtained are below:

$$f_h(w) = 30.014 \times e^{(0.0162753*w)}$$

$$p - value \ < 0.0001$$

$$R - squared: 0.288986$$

The trendline could be regarded as the expected value according to the previous behavior of the system. But due to various events (e.g., hurricanes in this project), the balance of the call center network was disrupted and changed drastically as it can be seen in Figure 4-6.

The p-value of the forecast was less than 0.0001 and the R-squared value was 0.29. Therefore, the null hypothesis was rejected since there was enough evidence to support the holding time is affected by the week of the year. On the other hand, R-squared as a statistical measure that represents the goodness of fit is low in this case, hence the demand was analyzed by location throughout the model.

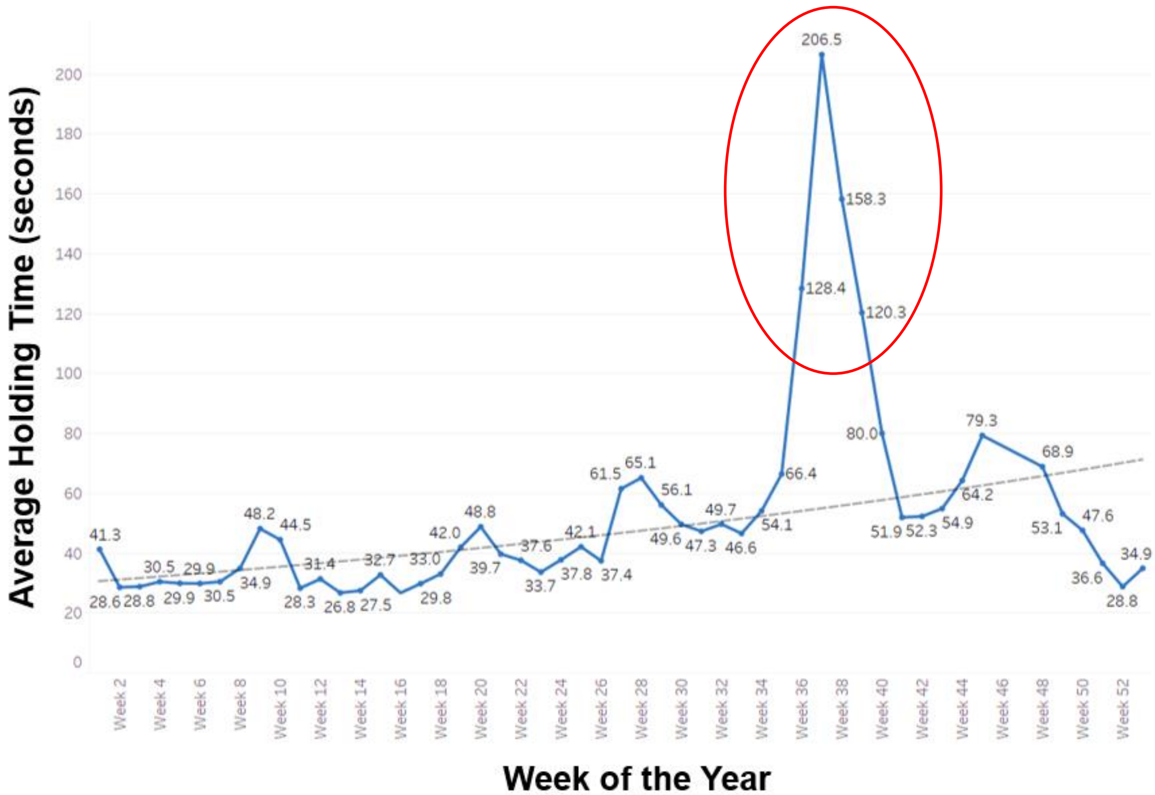

Figure 4-6. Average holding time of centers in Group D

From the P-value, we can find that the trendline defines the original data well. The peak between mid-July was regarded as "noise" and mid-September 2017 was in relation to Hurricane Harvey and Hurricane Irma in the US. Table 4-3 shows the difference between the value of the holding time trendline and the actual value in relationship with the week of the year. Growth rate is defined as:

$$Growth\ rate = \frac{average\ holding\ time - trendline}{trendline}$$

Table 4-3 shows the difference between the actual value and the expected value of the holding time across the group of call centers with an average holding time higher than 24.5 seconds. The deviation of the holding time in respect to the expected value can be seen in the following table.

Table 4-3. Difference between trendline and actual data (holding time)

Week of the Year	Avg. Hold Time (seconds)	Trendline (seconds)	Difference (seconds)	Growth Rate
Week 34	54.11	52.20	1.92	3.67%
Week 35	66.44	53.05	13.38	25.23%
Week 36	128.40	53.92	74.47	138.11%
Week 37	206.48	54.81	151.68	276.74%
Week 38	158.26	55.71	102.56	184.10%
Week 39	120.34	56.62	63.72	112.53%
Week 40	80.02	57.55	22.47	39.04%
Week 41	51.95	58.50	-6.55	-11.19%

Figure *4-7* illustrates the trendline across the historical data of number of records. The trendline was given by a polynomial equation with a p-value of 0.03 as it can be seen below:

$$f_r(w) = -0.550491 \times w^3 + 34.9979 \times w^2 - 372.275 \times w + 28865.2$$

$$p - value \ < 0.0298425$$

$$R - squared: 0.171929$$

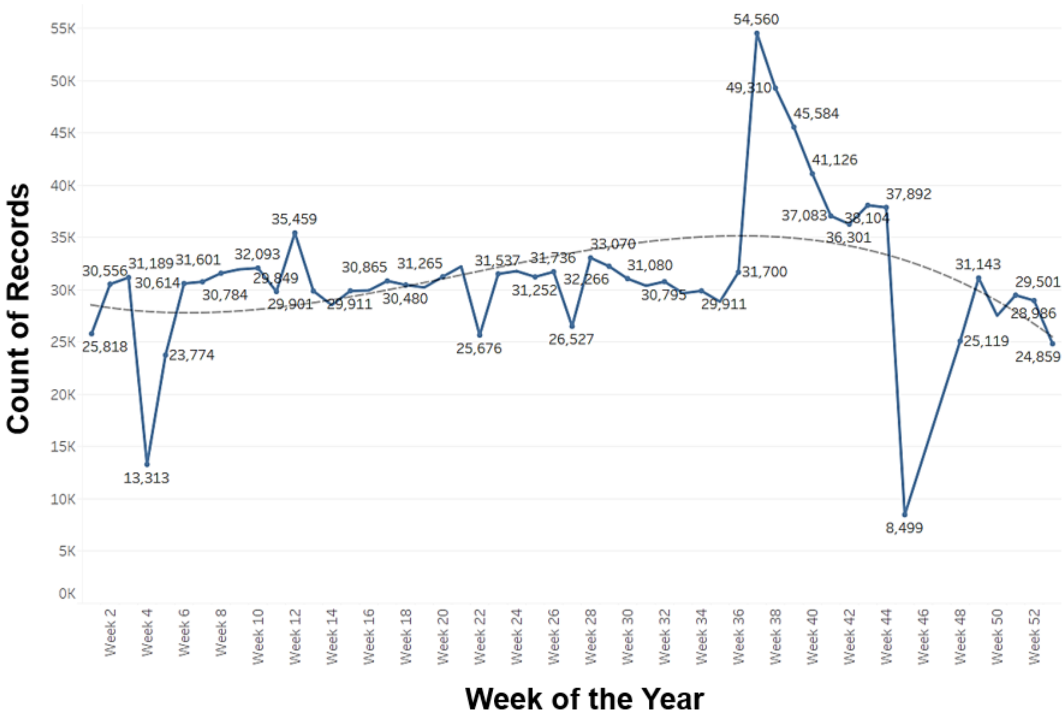

Figure 4-7. Average number of records for centers in Group D

Table 4-4 shows the difference between actual number of records and the value obtained through the trendline. A negative growth rate means the historical value was lower than the expected value based on the expected call arrival. Therefore, supporting the need of a model that deviates the call my leveraging the resources across the network of call centers at Company X.

Table 4-4. Difference between trendline and actual data (number of records)

Week of Date Call	Number of Records	Trendline	Difference	Growth Rate
Week 34	29911	35029	-5118	-14.61%
Week 35	28883	35106	-6223	-17.73%
Week 36	31700	35137	-3437	-9.78%
Week 37	54560	35119	19441	55.36%
Week 38	49310	35049	14261	40.69%
Week 39	45584	34924	10660	30.52%
Week 40	41126	34739	6387	18.38%
Week 41	37083	34493	2590	7.51%

4.3.2 DEMAND FORECASTING

Simple exponential smoothing was applied to all the call centers data grouped by date, location, queue and timeslot with their respective demand. The parameter alpha that was found to minimize the Root Mean Square Error (RMSE) across all locations was $\alpha = 0.21$ with an RMSE of 4.3356 calls as it can be seen on Table 4-5. This parameter was obtained after running a for loop that generated the RMSE statistic for all possible alpha values between zero and one with steps of 0.01 at a time using R.

Table 4-5. Alpha values with lowest RMSE

Alpha (α)	*RMSE*
0.21	4.335632
0.2	4.336268
0.18	4.337819
0.19	4.339523
0.17	4.343181
0.16	4.344477
0.15	4.347776
0.22	4.34808
0.23	4.350566

4.3.3 DEMAND ANALYSIS FOR LOCATION AFFECTED BY HURRICANE

Location 310 was affected by a hurricane in 2017. Therefore, we proceeded with the analysis of inbound call demand for location 310 before and after the catastrophe event. The distribution of inbound calls for location 310 is mapped in Figure 4-8 and it showcases that there was an increase in demand of around 250% for the first two weeks immediately after the hurricane occurred and it started decreasing steadily at the beginning of the fourth week after the hurricane.

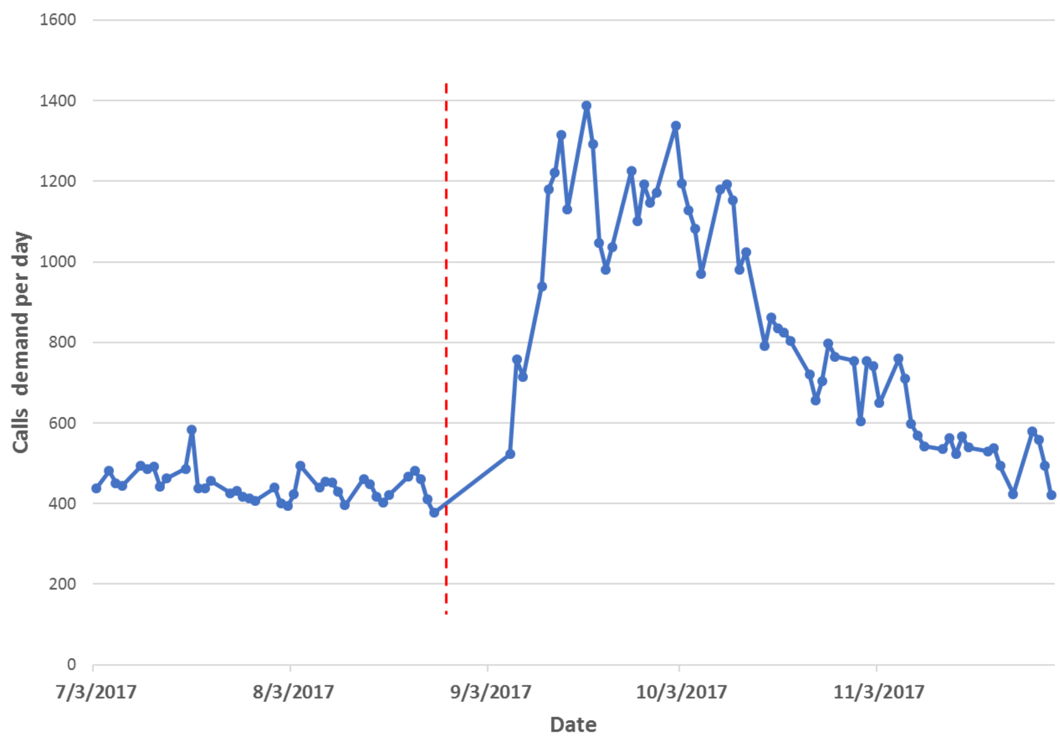

Figure 4-8. Calls Demand per day for location affected by Hurricane

Figure 4-9 is a graphical representation of the inbound calls for location 310 during the three weeks prior to the hurricane. During this time, the call center had the operation under control with an average of 4.5 calls in the system at any point in time, average holding time of 32 seconds and 4.3% abandonment call rate, therefore achieving the Service Level Agreement (SLA) Company X was promising its customers.

On the other hand, Figure 4-10 shows an increase of 141% in the average number of inbound calls during the three weeks after the hurricane. Waiting times of 143 seconds and abandoned calls went to rates of 28.9% due to the increased demand and lack of resources to serve the system at location 310.

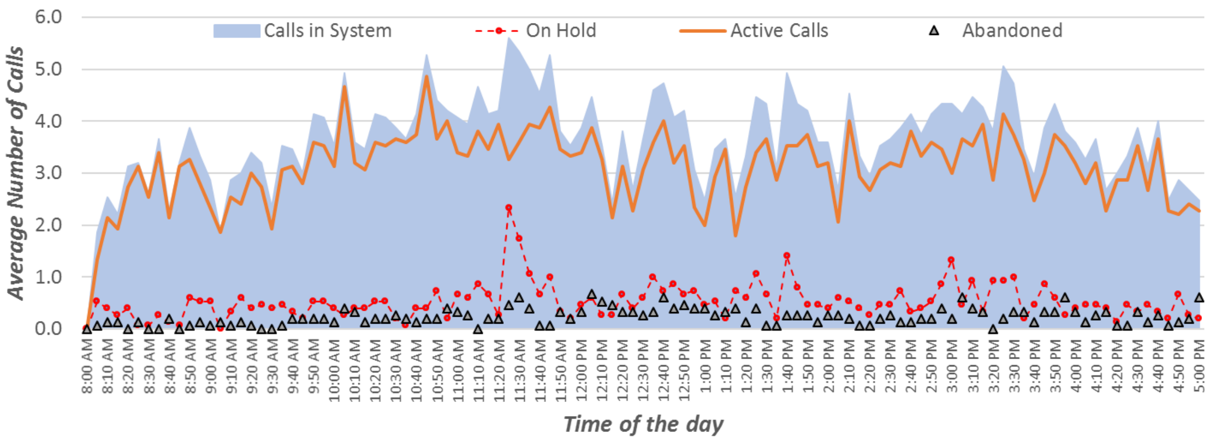

Figure 4-9. Inbound Calls Graph for calls placed three weeks before Hurricane

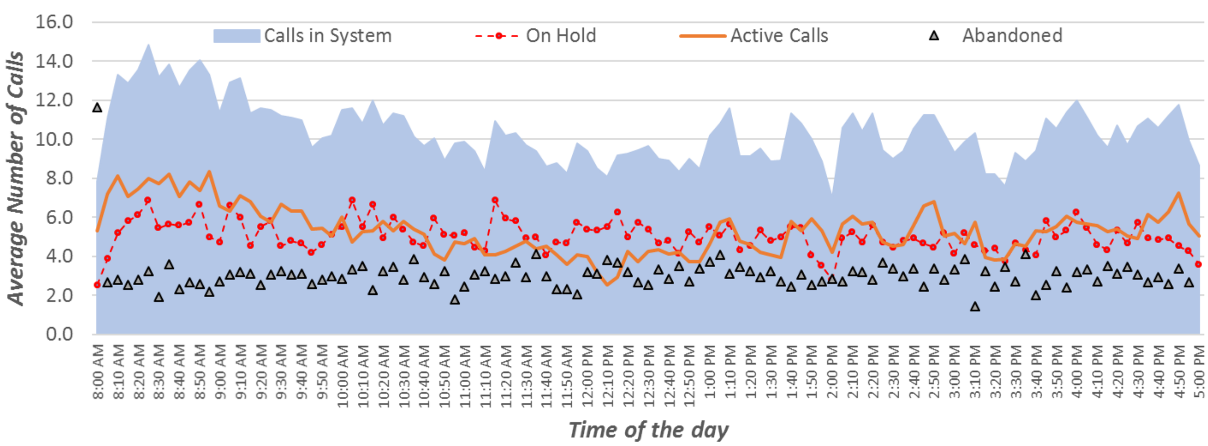

Figure 4-10. Inbound Calls Graph for calls placed three weeks after Hurricane

4.4 Capacity Results

The objective when building the capacity for each location was to have at least 95% of inbound calls answered within 60 seconds, due to the direct relationship between customer satisfaction and the holding time variable. Although, to account for the high variability of demand during catastrophe events, a target waiting time (t_q^{ijk}) of 45 seconds was used to find the associated maximum capacity for the combination of location (i), queue (j) and timeslot (k). Other variables that were included in the model to solve the queueing theory equation mentioned in the methodology were: rate of call arrival (r_a^{ijk}), coefficient of variation for interarrivals (CV_a^{ijk}), coefficient of variation of process time (CV_p^{ijk}), process time (t_p^{ijk}) and, number of parallel agents (m^{ijk}) available to serve the calls at location (i), queue (j) and timeslot (k).

The demand value was then subtracted from the maximum capacity value obtained through the queueing theory analysis to define the available capacity constraint (capacity bandwidth) that served as input for the optimization model. Table 4-6 contains a sample of the maximum capacity and capacity bandwidth results obtained for location 312.

Table 4-6. Sample of capacity results for location 312

Location	Queue	Timeslot	Demand	Max. Capacity	Capacity Bandwidth
312	Queue 1	1	6	54	48
312	Queue 1	2	6	58	52
312	Queue 2	3	4	58	54
312	Queue 2	4	3	46	43
312	Queue 3	5	3	30	27
312	Queue 3	6	3	23	20
312	Queue 5	7	3	24	21
312	Queue 5	8	3	4	1
312	Queue 4	9	3	42	39

4.5 OPTIMIZATION MODEL RESULTS

The next part of the program takes as inputs the forecasted demand for the location the calls need to be rerouted for and the capacity constraints established previously. Then, it populates a table with the results of the combination of location, queue and timeslot the calls should be deviated to by taking into consideration the demand, capacity and linking constraints mentioned in the methodology. The output from the optimization of one of the locations is shown in Figure 4-11. The last column makes reference to the timeslot the calls are coming from, for example if the target location 437 for calls in queue 1 was one hour ahead of the location the calls are being deviated for, the value of the timeslots for location 437 would increase by one when setting up the call receiving times at location 437 (e.g. Timeslots 437=c("3", "6", "8") given that the optimization accounts for this restriction.

Queue Location "A"	Location to	Timeslots "A"
1	349	c("1", "3", "4", "6", "8", "9")
1	437	c("2", "5", "7")
2	365	c("1", "2", "3", "4", "5", "6", "7", "8", "9")
3	429	c("1", "3", "7", "8")
3	391	c("2", "4", "5", "6", "9")
4	447	c("1", "2", "3", "4", "5", "6", "7", "8", "9")
5	349	c("1", "3", "4", "6", "7", "9")
5	447	c("2", "5", "8")

Figure 4-11. Example of optimization output

5. DISCUSSION

The discussion focuses on four parts: (i) Model iteration, (ii) Recommendations, (iii) Areas of Opportunity and, (iv) Further applications.

5.1 MODEL ITERATION

The Mixed Integer Linear Programming (MILP) model using the GLPK solver in R gives as an output the locations i to send the calls to for each combination of queue j and timeslot k. The first time the model was run it took six hours to give a result due to the size of the optimization matrix (7000 by 7000) and iterations needed to get a result considering all variables. Therefore, multiple trials with different number of locations were done to accommodate for a solution for Company X that will give them results in a shorter timeframe and without sacrificing the robustness of the model. Table 5-1 presents a summary of the time it took the model to run under different location quantities.

Table 5-1. Run Time for Optimization Model in R

Count of Locations	Run Time
6	4.3 sec
9	14.7 sec
10	1.1 min
12	27.3 min

The proposed solution to minimize run time was to run the optimization model 10 times with 9 random locations chosen as options for calls' supply in each iteration of the model. The total run time for the R code in the proposed solution was of 3.4 minutes, which is significantly lower than six hours run time. Additionally, if the optimization does not find a solution on the first sample there was a repeat loop at the end of the model that iterated the first part of the code

30 times. Therefore, increasing the probability of getting a result from the MILP model by using small samples for the combinations of location, queue and timeslot.

5.2 RECOMMENDATIONS

The research objective of the project was to find the effects of hurricanes to call centers and propose a solution for call deviation based on the combination of location, queue and timeslot. However, we recommend Company X to explore the feasibility of call deviation for the combination of queue and timeslot with holding time above 60 seconds for a period of two months by using the proposed optimization model. This will help Company X on improving service level by reducing variability in the inbound call process through the optimized use of resources.

5.3 AREAS OF OPPORTUNITY

In this case, due to the limitation of computer power and the characteristic of metaheuristics, the optimal results that were found were local optimal results. In this project, each time 6% of the call centers were picked to match the demand of the objective call center. The model ran 10 iterations to find the feasible/available results. Once there are no feasible results, the model has a reiterative loop built-in that stops on the thirtieth sample if no feasible solutions are found. Thus, for each objective call center, the model runs a maximum of 3000 times to search the results and each time picks a random sample of 6% call centers (Nine locations).

Consequently, for multiple samples the same locations can be chosen. Therefore, obtaining scenarios were the samples can overlap and the model might not cover all the call centers available to leverage the needs of the target call center. This means the results were associated with local optimal results, but are still feasible solutions for the inbound call scheduling process Company X was facing.

There are three main benefits captured by the proposed MILP model. First, the optimization model had the capability to give a response in under four minutes. Thus, bringing an easy to use model for the internal customer to take decisions under a high-pressure environment that these needs are generated in (e.g. climate events). Second, the model did not require the purchase of additional software for the organization, since it uses a free open source software infrastructure, R. Third, there was not need for Company X to acquire additional dedicated servers to run the model since they developed applications using R at the moment this project was done and were able to use the same servers for the model.

In contrast if there is a need for a more robust result, Company X should run the proposed MILP model taking all the locations into consideration with the use of a commercial software. Hence, reducing the running time of the model and providing a better communication with the internal user.

5.4 FURTHER APPLICATIONS

The optimization model defines a framework to balance resources in a network of call centers. However, the approach followed throughout this project can be applied to other industries that require to leverage the use of their resources with the objective of serving the customer, based on the Service Level Agreement promised to them.

Moreover, the model can be applied to Company X under other circumstances that require call deviation, some examples are outages and call center closures. This approach will help companies on reducing costs due to additional labor costs and minimize the risk of losing a customer due to bad service levels.

6. CONCLUSION

Catastrophe events like hurricanes critically impact the operation of Company X for a period of around two months after the hurricane has happened until the inbound call process starts to stabilize again for the affected location. A sudden increase of demand affects the service level agreement Company X has with its customers due to a shortage of labor resources to attend the inbound calls. Thus, the optimization model will help minimize the risk of losing a customer due to bad service during a catastrophe event at Company X.

This project used historical data from 152 call centers operated by Company X across the United States for the period between November 2016 and November 2017. The data showed patterns of sharp increases of inbound calls during and after climate events. The increases then lasted for periods of up to four months following the catastrophe event. Using, historical climate data from the NOAA, it was determined that the time frames of increasing calls coincided with major climate catastrophes and it was corroborated by Company X. Based on these findings, it was determined that an optimization model could help minimize the risk of losing a customer due to bad service during a catastrophe event at Company X by rerouting inbound calls to call centers that had available capacity.

A generalized framework for call rerouting was developed by breaking the problem up into four parts: (i) Data preprocessing, (ii) Demand analysis with the use of exponential smoothing, (iii) Capacity analysis utilizing queueing theory and, (iv) Optimization model using Mixed Integer Linear Programming (MILP) to define the locations to deviate the calls to during and after a catastrophe event based on Company X's requirements. In the preprocessing phase, holding and duration times above two hours were excluded from the model to increase reliability in the final result of the optimization. Demand analysis was done with the use of simple exponential smoothing with an optimal value of the alpha parameter of 0.21 that can be recalibrated in future stages by the company. The capacity analysis was done taking into consideration the Service Level Agreement (SLA) of answering at least 95% of the inbound calls under 60 seconds to find the maximum capacity for the combination of location, queue and timeslot. The optimization model was developed using the GLPK solver for MILP in R, it provides an understandable table that the telecom engineer can use after running the optimization model for three minutes to generate the rules in the phone system platform. The use of R throughout this project is important, since Company X was not required to acquire additional software for the optimization and the code developed has a seamless integration to the company dashboards that were using R programs at the moment of the project.

The results demonstrated that the best approach to handle such drastic increases in demand during and following a climate catastrophe is to reroute inbound calls to call centers with available capacity. The calls should be rerouted to leverage the resources in the organization and mitigate the risk of losing a customer due to bad service in the inbound call process at the location affected by a climate event. Furthermore, Company X can use the optimization model for other disruptions connected to the inbound calls like outages or leveraging general inbound capacity across the organization.

In conclusion, Company X will greatly benefit from this research because it is a real-life application to solve one of the major problems currently facing the company. Based on climate projections, catastrophe events are becoming more relevant and more common place. Therefore, it is critical that Company X implements this methodology to help relieve its overburdened call centers during future climate events. This will lead to quicker response times, better customer service and higher customer satisfaction. Thus, having the potential to yield additional business and further growth of Company X in the industry.

7. REFERENCE LIST

Ali III, L.F. (2010). A Call Center Simulation Study: Comparing the Reliability of Cross-Trained Agents to Specialized Agents

Batt, R., Holman D. & Holtgrewe U. (2007) The Global Call Center Report: International Perspectives on Management and Employment

Brown, T.C. et. al. (2017). Avoiding an uncertain catastrophe: climate change mitigation under risk and wealth heterogeneity. *Springer*

Ghirardato P, Maccheroni F, Marinacci M (2004) Differentiating ambiguity and ambiguity attitude. J. Econom. Theory 118(2): 133–173.

Gurvich, A., Luedtke, J. & Tezcan, T. (2010). Staffing Call-Centers with Uncertain Demand Forecasts: A Chance-Constrained Optimization Approach.

Koole, G. (2004) Performance analysis and optimization in customer contact centers

Loïc Berger, Johannes Emmerling, Massimo Tavoni (2017) Managing Catastrophic Climate Risks Under Model Uncertainty Aversion. Management Science 63(3):749-765. https://doi.org/10.1287/mnsc.2015.2365

Stolletz. R (2003) Performance Analysis and Optimization of Inbound Call Centers. *Springer*

Thirumaran M., Subham Soni , Brendha G. Gayathry. (2015) An Intelligent Interactive Voice Response System for Banking Domain. *in Proc. Int. Conf. on Advanced Research in comput. Sci & Eng.*

U.S. Billion-Dollar Weather and Climate Disasters (2017). Retrieved November 07, 2017, from https://www.ncdc.noaa.gov/billions/

Whit. W (2005) Engineering Solution of a Basic Call-Center Model. *INFORMS online.* Retrieved from https://pubsonline.informs.org/doi/abs/10.1287/mnsc.1040.0302

APPENDIX A – OUTLIERS OVERVIEW

The table below includes the records for the outliers found in the initial dataset provided by company X.

Table A1. Data overview of Outliers

Location ID	Average Duration Time	Average Holding Time	Average Total Time	Median Duration Time	Median Holding Time	Median Total Time	Number of Records	Percentage of Outliers included
310	171	72	243	104	23	163	132333	2.54%
311	178	121	298	111	27	186	113444	2.18%
352	166	41	206	109	12	146	95445	1.83%
366	214	42	256	137	10	172	88827	1.71%
395	156	186	342	85	42	210	87974	1.69%
354	201	26	228	132	11	157	84384	1.62%
329	213	31	244	150	13	179	79381	1.53%
342	236	43	279	154	22	195	75353	1.45%
345	297	33	330	180	10	212	66179	1.27%
432	174	28	202	107	11	136	61485	1.18%
374	221	42	263	147	9	178	57404	1.10%
393	182	29	212	122	12	150	55459	1.07%
314	175	34	209	110	12	143	54854	1.05%
383	168	32	201	117	11	148	50517	0.97%
321	154	52	206	96	11	134	49688	0.95%
430	172	28	200	118	10	145	48399	0.93%
317	167	29	196	112	10	138	46483	0.89%
450	248	97	345	164	14	239	43817	0.84%
394	191	32	223	131	12	162	42373	0.81%
348	198	32	230	129	10	153	40219	0.77%
307	200	62	262	134	25	195	38653	0.74%
451	208	50	259	137	13	184	37428	0.72%
332	208	55	263	120	10	166	35195	0.68%
362	198	33	231	142	10	175	30614	0.59%
371	163	41	204	107	9	142	29261	0.56%
497	161	109	271	109	20	176	23830	0.46%
449	135	26	160	91	12	117	23379	0.45%
							Grand Total	30.60%

APPENDIX B – DEMAND ANALYSIS CODE

The code below shows the preprocessing of the input data to generate the demand data frame, in addition to the demand forecast code using simple exponential smoothing from the package forecast in R.

```
#### BUILDING DEMAND DATA FRAME ####


## DEMAND DATAFRAME

# Building Timeslots of one hour between arrival times
call.logs$timeslot_arrivals <-
  ifelse(call.logs$at>= 8 & call.logs$at <9,1,
       ifelse(call.logs$at>= 9 & call.logs$at <10,2,
            ifelse(call.logs$at>= 10 & call.logs$at <11,3,
                 ifelse(call.logs$at>= 11 & call.logs$at <12,4,
                      ifelse(call.logs$at>= 12 & call.logs$at <13,5,
                           ifelse(call.logs$at>= 13 & call.logs$at <14,6,
                                ifelse(call.logs$at>= 14 & call.logs$at <15,7,
                                     ifelse(call.logs$at>= 15 & call.logs$at <16,8,
                                          ifelse(call.logs$at>= 16 & call.logs$at <17,9
                                               ,0)))))))))


# Summary of statistics of call arrivals per timeslot based on arrivals
calls.summary.demand <-as.data.frame(as.data.table(call.logs)[, list(countCalls = length(id)),
                                                         by = list(dateCall,location_id, queue,
timeslot_arrivals)])

setorder(calls.summary.demand,'dateCall')

# Exponential smoothing demand with and alpha of 0.2 for next two weeks
location.demand <-
as.data.frame(as.data.table(calls.summary.demand[ccalls.summary.demandsd$timeslot_arrivals!=0,])[,
                              list(calls.demand = max(ses(countCalls,h=10,
                              alpha=alpha.ses, initial="simple")$upper[,1])),
                              by = list(location_id, queue, timeslot_arrivals)])


location.demand <- location.demand[,c('location_id','queue','timeslot_arrivals','calls.demand')]


###task Add demand data to capacity dataframe to find left capacity

names(location.demand) <- c('location','queue','timeslot','rhs.demand')

df.demand <- location.demand

# df.demand <- df.demand[df.demand$location==location.number,]
```

```
df.demand$queue.timeslot <- paste0(substr(df.demand$queue,7,7),"_",df.demand$timeslot)

df.demand$rhs.demand <- ceiling(df.demand$rhs.demand)

df.demand$queue <- substr(df.demand$queue,7,7)
```

APPENDIX C – CAPACITY ANALYSIS CODE

The code below shows the preprocessing of the input data to generate the capacity data frame. Furthermore, it uses the package raster in R to define the rate of arrival accepted as maximum capacity for the combination of location, queue and timeslot.

```
#### BUILDING CAPACITY DATA FRAME ####
# Building Timeslots of one hour between arrival time and service time
call.logs$timeslot_serv2endT <-
  ifelse(call.logs$st>= 8 & call.logs$et <9,1,
         ifelse(call.logs$st>= 9 & call.logs$et <10,2,
                ifelse(call.logs$st>= 10 & call.logs$et <11,3,
                       ifelse(call.logs$st>= 11 & call.logs$et <12,4,
                              ifelse(call.logs$st>= 12 & call.logs$et <13,5,
                                     ifelse(call.logs$st>= 13 & call.logs$et <14,6,
                                            ifelse(call.logs$st>= 14 & call.logs$et <15,7,
                                                   ifelse(call.logs$st>= 15 & call.logs$et <16,8,
                                                          ifelse(call.logs$st>= 16 & call.logs$et <17,9
                                                                 ,0)))))))))

#### Get number of agents available ####
summary.servers <- as.data.frame(as.data.table(call.logs[call.logs$durationTime!=0
                                          & call.logs$timeslot_serv2endT!=0,])[, list(callsAgent = length(id)),
                                          by = list(dateCall,location_id, queue,
timeslot_serv2endT,agentCode)])


ssf <- as.data.frame(as.data.table(summary.servers)[, list(avgCalls = mean(callsAgent)),
                                             by = list(location_id, queue,
timeslot_serv2endT,agentCode)])

names(ssf) <- c('location','queue','timeslot','agentCode','avgCalls')


ssf <- merge(ssf,location.demand,by=c('location','queue','timeslot'), all.x=TRUE)

ssf$perc <- ssf$avgCalls/ssf$rhs.demand
# hist(ss2$perc)

# INclude only agents that have serve calls above the 25% percentile
list.agents <- ssf[ssf$perc>=as.data.frame(quantile(ssf$perc))[2,1],]

# List of agents available by location, queue and timeslot
agents.available <- as.data.frame(as.data.table(list.agents)[, list(servers =
length(unique(agentCode))),
by = list(location, queue, timeslot)])

#### agents available end ####

#### Get arrival data ####
```

```
# Mean ariival and sdArrival in hours
sum.arrival <- as.data.frame(as.data.table(calls.summary.demand)[, list(meanArrival =
mean(countCalls, na.rm = TRUE ),
                                        sdArrival = sd(countCalls, na.rm = TRUE )),
                                        by = list(location_id, queue, timeslot_arrivals)])

# Coefficient of variation of interarrivals
sum.arrival$cvArrival <- 1/(sum.arrival$sdArrival/sum.arrival$meanArrival)
names(sum.arrival) <- c('location','queue','timeslot','ra','sdArrival','cva')

sum.arrival <- sum.arrival[sum.arrival$timeslot!=0,c('location','queue','timeslot','ra','cva')]

#### Arrival Data end ####


#### Get process time data ####
# Summary of statistics of calls serviced per time slot where duration time <> 0 (Calls not
answered)
csc <- as.data.frame(as.data.table(call.logs[call.logs$durationTime!=0,])[, list( meanProcess =
mean(durationTime, na.rm = TRUE ),
                                            sdProcess = sd(durationTime, na.rm =
TRUE ),
                                            cvProcess = cv(durationTime, na.rm =
TRUE )/100
                                           ),
                                           by = list(location_id, queue,
timeslot_serv2endT)])


# Generate tp hours per call
csc$tp <- csc$meanProcess/(60*60)


names(csc) <- c('location','queue','timeslot','meanProcess','sdProcess','cvp','tp')

csc <- csc[csc$timeslot!=0,c('location','queue','timeslot','cvp','tp')]



#### process time data end ####

#### Capacity Calculation ####

cc <- merge(sum.arrival,csc,by=c('location','queue','timeslot'),all.x=TRUE)
cc <- merge(cc,agents.available,by=c('location','queue','timeslot'),all.x=TRUE)
cc <- merge(cc,location.demand,by=c('location','queue','timeslot'),all.x=TRUE)

# Define waiting time desired in seconds
tq <- tq

# Transform to hours
tq <- tq/(60*60)

cc$a <- tq*2/(((cc$cva^2)+(cc$cvp^2))*cc$tp)

cc$b <- ((2*(cc$servers+1))^(1/2))-1
```

```
# Define utilization
# cc$u <- 0.5
# cc$totalCap <- (cc$servers*cc$u)/cc$tp

cc$totalCap <- 0

# Defining total Capacity
for (i in 1:nrow(cc)){
  if(sum(is.na(cc[i,])) || cc$servers[i]==0){
    cc$totalCap[i] <- 0
  }else {
fun <- function (x) -cc$a[i] + x*(cc$a[i]*cc$tp[i]/cc$servers[i]) + (x*cc$tp[i]/cc$servers[i])^cc$b[i]
cc$totalCap[i] <- uniroot(fun, c(0, 1000))$root

  }

print(i)
}

cc$capConstraint <- floor(cc$totalCap) - ceiling(cc$rhs.demand)

cc$capConstraint <- ifelse(cc$capConstraint<0,0,cc$capConstraint)

## CAPACITY FILTERING FOR LOCATION THAT NEEDS CALLS REROUTING
df.capacity <- cc[,c('location','queue','timeslot','capConstraint')]

# Include specific location time off
df.capacity <-
merge(df.capacity,location_data[,c('location_id','loc_time_off')],by.x='location',by.y='location_id',all.x=TRUE)
```

APPENDIX D – OPTIMIZATION MODEL CODE

The code below uses the GLPK solver in R to develop a Mixed Integer Linear Programming (MILP) model that solves for the locations that the calls should be rerouted in each queue and timeslot combination.

```
library(Rglpk) # Optimization solver for MILP
library(data.table)

### It simulates 30 random optimization models and looks at
# the common solutions between the models to give the final
# recommendation to the company

wd_path <- 'C:/Users/vivi_/Dropbox (MIT)/Capstone MIT/model_ps/data'
setwd(wd_path)

optimization.run <- function(){

load("capacity.Rdta")
load("demand.Rdta")


## Load location data
load("locationData_11232017.Rdta")

list.locations <- unique(df.capacity$location)

st <- Sys.time()

# Location to deviate the calls from
# location.number <- 310

read.location <- function(){
  n <- readline(prompt="Please, enter the location number you want to deviate calls from: ")
}

location.number <- as.integer(read.location())

while (!location.number %in% list.locations) {
  print("Sorry, the location is not active in the phone system. Please use a location existent in
the location list.");
  location.number <- as.integer(read.location());
}

location.number <<- location.number

# Filtering demand
df.demand <- df.demand[df.demand$location==location.number,]
setorder(df.demand,'queue','timeslot')
```

```
	df.demand <<- df.demand

	print(paste0("The current demand for location ",location.number," is:"))
	print(df.demand[,c('location','queue','timeslot','rhs.demand')])

	print(paste0("If the location is suffering from a catastrophe event is recommended to increase
the call demand in the optimization"))

	  read.demand.multiplier <- function(){

	    print("The analysis of demand under Hurricane Harvey showed an increased in demand of up
to ")
	    print("240% in the first two weeks after the Hurricane and 140% between week 3 and 5 :")

	    n <- readline(prompt=paste0("Based on hurricane level, please enter 1 if you want to keep
the same demand. Otherwise enter a number between zero and three: "))
	  }

	  # , please note that 4 would mean that",
	  #                  "location ",location.number," will have four times the calls it usually has

	  demand.multiplier <- as.numeric(read.demand.multiplier())

	  while (demand.multiplier<=0 | demand.multiplier>4) {
	    print("Sorry, this multiplier is not accepted.")
	    demand.multiplier <- as.numeric(read.demand.multiplier())

	    if(!(demand.multiplier<=0 | demand.multiplier>4))
	      print("Thank you!")

	  }


	  df.demand$rhs.demand <- floor(df.demand$rhs.demand*demand.multiplier)


	#### Preprocess DF capacity based on location for rerouting ####
	## Time difference for location that requires calls rerouting
	loc.time.off <- location_data$loc_time_off[location_data$location_id==location.number]

	# Defining timeslot to match based on location that demands rerouting of calls
	df.capacity$match.timeslot <- df.capacity$timeslot-df.capacity$loc_time_off+loc.time.off

	# Ignore timeslots that are not between 1 and 9
	df.capacity <- df.capacity[df.capacity$match.timeslot>=1 & df.capacity$match.timeslot <=9,]

	#Ignore capacity for which the calls need to be reroute from
	df.capacity <- df.capacity[df.capacity$location!=location.number,]

	# Define column location.queue
	df.capacity$loc.queue <- paste0(df.capacity$location,"_",substr(df.capacity$queue,7,7))

	# Define column location.queue.timeslot that matches
	df.capacity$loc.queue.timeslot <-
paste0(df.capacity$location,"_",substr(df.capacity$queue,7,7),"_",df.capacity$match.timeslot)
```

```
#
df.capacity$queue.timeslot <-
paste0(substr(df.capacity$queue,7,7),"_",df.capacity$match.timeslot)

# Dataframe capacity
df.capacity <-
merge(df.capacity,df.demand[df.demand$location==location.number,c('rhs.demand','queue.timeslot')],
by='queue.timeslot',all.x=TRUE)

df.capacity$rhs.capacity <- df.capacity$capConstraint/df.capacity$rhs.demand

# df.capacity <-
df.capacity[,c('loc.queue.timeslot','queue.timeslot','loc.queue','rhs.capacity','rhs.demand')]
df.capacity <-
df.capacity[,c('location','queue','match.timeslot','rhs.capacity','rhs.demand','capConstraint')]

names(df.capacity) <-
c('location','queue','timeslot','rhs.capacity','rhs.demand.mult','capConstraint')

df.capacity$queue <- substr(df.capacity$queue,7,7)

#### ####



df.capacity$lqt <- paste0(df.capacity$location,"_",df.capacity$queue,"_",df.capacity$timeslot)
df.capacity$lq <- paste0(df.capacity$location,"_",df.capacity$queue)
df.demand$qt <-  paste0(df.demand$queue,"_",df.demand$timeslot)

#### Choosing 6% random locations as option to send the calls to ####

loc.list <- as.data.frame(unique(df.capacity$location))
perc <- 0.06

# Empty dataframe to save recommendations of the model
rec <- data.frame(
  var=character(),
  sol=as.integer()

)

df.cap <- df.capacity

recommendations <- function(){

for (a in 1:10){

loc.index <- sample(1:dim(loc.list)[1], dim(loc.list)[1]*perc)
loc.df <- loc.list[loc.index, ]
loc.df <<- loc.df

df.capacity <- df.cap[df.cap$location %in% loc.df,]

# df.capacity <- df.capacity[df.capacity$rhs.capacity>=1,]

#### end ####
```

```
setorder(df.capacity,'timeslot','location','queue')
setorder(df.demand,'timeslot','location','queue')

# Defining matrix row and column names
lqt_cap_cons <- unique(df.capacity[,c('lqt','rhs.capacity')])
lqt_cap <- lqt_cap_cons$lqt

qt_dem_cons <- unique(df.demand[,c('qt','rhs.demand')])
qt_demand <- qt_dem_cons$qt


lq_cap <- unique(df.capacity$lq)

loc <- unique(df.capacity$location)
timeslot <- unique(df.capacity$timeslot)


r.names <- c(lqt_cap,qt_demand,lq_cap)
c.names <- c(lq_cap,lqt_cap)


# Inbound call matrix for one location
inbound.matrix <- matrix(0, nrow = length(r.names), ncol = length(c.names))

rownames(inbound.matrix) <- r.names
colnames(inbound.matrix) <- c.names


# Objective function
obj <- vector(mode='numeric',length=length(c.names))
obj[1:length(lq_cap)]  <- 120
obj[(length(lq_cap)+1):(length(lq_cap)+length(lqt_cap))]  <- 30



# Capacity Constraints
for (i in lqt_cap){

  inbound.matrix[i,i]<-1

}
rm(i)


# Demand constraints
for (i in qt_demand){

  for (j in loc){

    if (paste0(j,'_',i) %in% c.names){
    inbound.matrix[i,paste0(j,'_',i)] <- df.demand$rhs.demand[df.demand$qt==i]
    }else{

    }
```

```
  }
}
rm(i,j)

# Linking Constraints part 1
m <- -1000

for (i in lq_cap){

  inbound.matrix[i,i]<-m

}
rm(i)

# Linking Constraints part 2
for (i in lq_cap){

  for (j in timeslot){

    if (paste0(i,'_',j) %in% c.names){
      inbound.matrix[i,paste0(i,'_',j)] <- 1
    }else{}
  }
}
rm(i,j)

# Direction of the constraints

  dir <- vector(mode='character',length=length(r.names))

  dir[1:length(lqt_cap)]  <- "<="
  dir[(length(lqt_cap)+1):(length(lqt_cap)+length(qt_demand))]  <- ">="

dir[(length(lqt_cap)+length(qt_demand)+1):(length(lqt_cap)+length(qt_demand)+length(lq_cap))]  <-
"<="

#
  rhs <- vector(mode='numeric',length=length(r.names))

  rhs[1:length(lqt_cap)]  <- lqt_cap_cons$rhs.capacity
  rhs[(length(lqt_cap)+1):(length(lqt_cap)+length(qt_demand))]  <- qt_dem_cons$rhs.demand

rhs[(length(lqt_cap)+length(qt_demand)+1):(length(lqt_cap)+length(qt_demand)+length(lq_cap))]  <- 0


# Restriction for all variables to be Integers
types <- 'I'

# Type of optimization (Minimize costs)
min <- TRUE


# Run Optimization
result.opt <- Rglpk_solve_LP(obj, inbound.matrix, dir, rhs, types = types, min = min)
```

```
# Dataframe with recommendations per iteration
result.iteration <- data.frame('var'=colnames(inbound.matrix),'sol'=result.opt$solution)

result.iteration$var <- as.character(result.iteration$var)
result.iteration <- result.iteration[result.iteration$sol!=0,]

if (nrow(result.iteration)==0){

}else{

  # Dataframe that appends results for each iteration
  rec <- rbind(rec,result.iteration)

}

rm(result.iteration)

# print(a)

}
  ## Return recommendations
  return(rec)
}

rec <- recommendations()
sample.number <- 2

while (nrow(rec)==0 & sample.number<=30) {
  if(sample.number<30){print(paste0("Sorry, there were not recommendations found with
random sample ",sample.number,
              ". The optimization will run again with a new random sample of ",length(loc.df),"
locations"))}

  # print(sample.number)
  rec <<- recommendations()

  if(nrow(rec)!=0){
    print(paste0("The optimization model run succesfully after ",sample.number, " sample
iterations"))}

  if(nrow(rec)==0 & sample.number==30){
    print(paste0("It was not possible to find a recommendation after ",sample.number, " sample
iterations"))
    print("Please consider running an optimization with all locations with the use of a commercial
solver")}

  sample.number <- sample.number+1
  }

if(nrow(rec)!=0){

rec.summary <- as.data.frame(as.data.table(rec)[, list(frequency = sum(sol, na.rm = TRUE )),
                                    by = list(var)])

rec.summary <- rec[nchar(rec$var)!=5,]
```

```
rec.summary$qt <- substr(rec.summary$var,5,7)

setorder(rec.summary,-'sol')

rec.final <- data.frame('qt'=unique(rec.summary$qt))
rec.final$qt <- as.character(rec.final$qt)
rec.final$var <- ''

for(i in 1:nrow(rec.final)){

   data <- subset(rec.summary, qt == rec.final$qt[i])

   data2 <- data[order(-data$sol), ]

   rec.final$var[i] <- data2$var[1]

}

rec.final$loc.to <- substr(rec.final$var,1,3)

rec.final$queue.from <- substr(rec.final$qt,1,1)
rec.final$timeslot.from <- substr(rec.final$qt,3,3)

#Locations recommended to reroute calls to per queue

results.lq <- unique(rec.final[,c('queue.from','loc.to')])

colnames(results.lq) <- c('queue','location')

setorder(results.lq,'queue')

#Locations recommended to reroute calls to per queue and timeslot
results.lqt <- unique(rec.final[,c('queue.from','loc.to','timeslot.from')])
colnames(results.lqt) <- c('queue','location','timeslot')
setorder(results.lqt,'queue','timeslot')


final.results.lqt <- results.lq



for (i in 1:nrow(final.results.lqt)){
final.results.lqt$timeslots.origin[i] <-
list(unique(results.lqt[results.lqt$location==results.lq$location[i]
                                    & results.lqt$queue==results.lq$queue[i],'timeslot']))
}


colnames(results.lq)  <- c('Queue Location "A"',"Location to")
colnames(final.results.lqt)  <- c('Queue Location "A"',"Location to",'Timeslots "A"')



print(paste0('Below you can find the list of locations that is recommended to reroute calls to per
queue in location',location.number))
```

```
print(results.lq)

print(paste0('Below you can find the combination of location, queue and timeslot that is recommended to reroute the inbound calls to for location ',location.number))
print(final.results.lqt)

results.lq <<- results.lq
final.results.lqt <<- final.results.lqt

}

}

optimization.run()
```