Improving Call & Email Blending

A Call Center Simulation Study

Enzo van Kessel

Student number: 5790174

Supervisor 1: J.M. van den Akker

Supervisor 2: J.A. Hoogeveen

May 2019

Content

Call Center Simulation – Thesis Proposal

This chapter details the proposal phase of the thesis and serves as an introduction to this work. The chapter starts with an introduction to the problem environment and discusses how the idea behind the project came to be. The tool used in this study, discrete event simulation is explained, followed by the reviewed documents from the literature study. The research question alongside several sub questions are discussed, setting a goal for the project. Finally the scientific relevance and research methodology are discussed, followed by the project plan.

Thesis Proposal

Introduction

For many companies, the call centre is the central contact interface for customers. It is used to sell products or offer to assistance. A call centre including its perceived quality is vital to any company. Across the day calls are made to the call center, during certain hours or even half hours the volume can fluctuate, which makes scheduling of agents difficult. If there are more calls than agents can handle the calls are transferred to a queue. When the waiting time in a queue becomes too large callers get annoyed possibly resulting in a bad reputation and call abandonments. Given that many companies sell products from their call centre abandoned calls and long queues result in a loss of revenue.

In order to optimize productivity and revenue call centres typically track certain performance measures. These are handling time, number of waiting calls in a queue or agent occupation. However, merely having this information is not enough to efficiently staff the call centre. In order to gain (near) optimal results various aspects of the call center should be thoroughly studied. For instance, factors like the frequency and duration of calls make staffing difficult these vary from day to day and even hour to hour. Simply looking at the performance parameters will give an idea of the problem and not the solution. Exploring possible solutions to this problem in a production environment is infeasible due to the number of participators, cost and possible downtime of business.

This project will be performed for CC4Skype, who requests a generic tool for call centre simulation. CC4Skype is a Dutch software company specialized in customer care solutions based on Skype 4 Business. CC4Skype created its own call centre platform branded CC4skype (Contact Centre for Skype). The mission of CC4Skype is to provide a complete customer care solution (call centre/receptionist software), that relies on Microsoft's Skype 4 Business.

To test and improve new features and increase attractiveness of CC4skype as a product, the purpose of this project is to create a proof of concept of a CC4Skype generic simulation tool based on the configuration options of their call centre solution. This provides the opportunity to pursue research related to email handling. Nowadays it is common for call centers to shift their focus towards multi-channel communication such as email and webchat. This raises the question: Should agent planning adapt to the amount of incoming emails or webchat messages? When contact via email becomes preferable over calling, it can be assumed that the response time becomes more relevant to the contact centre service level. The purpose of this research is to better understand the impact of email. Asking questions like: At what rate of incoming emails will service levels suffer? Can structures like routing strategies for email handling improve service levels? Until now hardly any research has yet been performed on the impact of email handling in call centres which makes this a relevant topic for call centre simulation.

Problem Statement

For many years call centres have been subject of various mathematics and computer science related projects aimed at solving planning, forecasting and routing problems. The results of these projects quickly became the standard in the industry. The success of a call centre is measured by the quality of service. That is the time it takes to speak with an agent and how well this agent is able to help a customer. In order to guarantee a certain degree of quality or service level, enough agents should be scheduled throughout the day. However, this should be done whilst balancing call centre performance parameters: cost, agent occupation and service level. In the early years of call centres the mathematical framework Erlang c was used to forecast the call volume in call centers, due to Erlang c's generic properties it's unsuitable for every call center [12].

Later, simulation greatly improved forecasting and allowed exploration of new performance increasing methods. Many papers have been written about simulation in call centers. Topics as: exploring new routing mechanisms; algorithms for the staffing of queues; the forecasting of future calls based external factors such as the weather.

Due to a shift in trends and technology, telephony is losing ground as main communication channel. Companies that operate in the communications sector frequently offer multiple channels like webchat, WhatsApp and email. Since multichannel communications slowly becoming the norm in call centers, multichannel solutions are being pushed by the companies that create call center solutions. This means that the daily operations for call centre agents will change as well. In call centres where agents handle emails as well as calls, it is to be expected that this will possibly influences service levels and definitely affects agent occupation. This raises the questions: What is the impact of multichannel communications on call centre agents? and how to deal efficiently different kind of jobs such as email?

Discrete Event Simulation

For many real-world applications, it is a very expensive process to test new ways of operating or different configurations on a system. In order to overcome this, statistical methods or simulation can be employed in order to save time, cost and harm. In the call centre industry workforce is typically evaluated and improved with use of Erlang c workforce management. This is a generic statistical workforce planner. Based on the number of agents and the call centre load, Erlang c produces a generic solution to staffing and workforce management. This means that call centres that do not fit the standard are likely to obtain a less reliable result. To overcome this problem in this industry Discrete event simulation (DES) is typically used to validate the outcome of Erlang c.

DES is simulating real world problems through sequencing discrete events that occur in the real world, where each event changes the state of the system. DES only moves forward in time by handling new events, if no event is handled the state remains the same. Skipping from event to event drastically decreases the runtime of the simulation if compared to continuous simulation where the simulation tracks the time. Events are stochastically generated by the system, for example the arrival of a customer at a queue at a specific time. Because the arrival rates are modelled on production data, DES is able to accurately recreate real world scenario's. Simulating a system with DES allows testing and evaluation of systems in increased time with relatively low cost.

The low cost and flexibility of DES allows for "what if" situations to be evaluated. This can be easily implemented by changing protocols that model the real world or by simply changing the number of agents in the system. The performance of the simulation can be evaluated with one or multiple performance parameters. Based on the outcome of the simulation choices and strategies can be validated or rejected.

Literature study

Due to call centres being textbook simulation examples, there is a lot of work performed in this field and abundant material is available for a literature study. However, call center simulation consists of various aspects: Model creation; Simulation methods(software/programming); model validation and verification; routing techniques; output analysis. Research performed should be broad enough to cover all aspects of performing a simulation study.

The goal of this literature study is to get an understanding of how a simulation experiment at a call center should be performed. What are the best practices to take into consideration. Most of the literature was chosen because it subjects had the possibility to further and or improve this research.

The literature can be categorized on the following topics:

- Case studies
- Model Creation
- Call routing (I.E. structures like overflow queues)
- Call centre simulation software
- Design of Experiment
- Latest Research

Case studies, Model Creation & Call routing

[1]. Knowledge Acquisition and Model Abstraction.

According to Kotiadis and Robinson, model creation is best done by interactive data acquisition. Their paper encourages a method of data acquisition where the persons acting in the problem environment are all to be heard in order to identify what the problem is and what factors of influence exist. The reason that this process has to be an interactive one, is that persons belonging to the problem environment usually have something else to do and you will most likely suffer from their lack of interest. Therefore, these actors need to be triggered, the authors encourage dialogue between opposing parties in the problem environment, by opening (heated) dialogue between the two the author expects useful insights in the problem environment.

[3]. Modelling and Simulation of a Telephone Call Center.

In the paper modelling and simulation of a telephone call centre the author describes the process of constructing a call centre simulation and specific problems faced in the process. According to the author, call centers typically deal with high variance in call arrival, which make it unrealistic to model arrival times by Poisson process with a deterministic time varying arrival rate. Instead it would be better to model as a nonhomogeneous process where the mean arrival rate is allowed to vary.

Another important takeaway from this paper is that an agent's availability will most likely be less than scheduled due to all kinds of breaks (pee and smoke breaks for instance). The reason for absence of agents during their scheduled availability could be anything, which makes it difficult to account for in a model.

[4] Call Centre Simulation Modelling: Methods, Challenges and Opportunities.

This paper offers valuable insight in model creation for call centre simulation, as it is dubbed a tutorial for call centre simulation. The authors give an overview of key in- and output parameters (figure 1) and heeds warning of shrinkage in call center. Shrinkage is the a-priori unknown absence of call centre agents, which can be as high as 30% according to the authors. Other takeaways from this paper are the support for call arrivals modelling with use of Poisson distribution. Modelling the average handling time by means of an exponential distribution is however strongly discouraged although this has been the norm for some time. Exponential distribution has been used more frequently than it should be due to aggregation of forecasting data. Abandonment in queues also expressed as patience factor, is encouraged to be modelled as an exponential random variable as data about this statistic is commonly sparse.

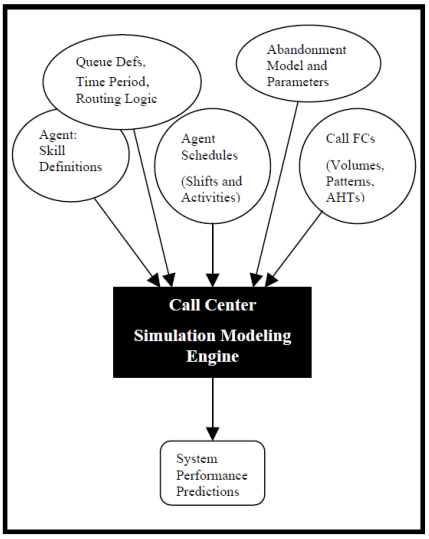

Figure 1, Turning Key Input Parameters into Output parameters, according to Methora et al, 2003

When performing a simulation experiment the number of replications of the simulation is an important factor. The reason for this is the trustworthiness of the simulation.

[5] On the modelling and forecasting of call centre arrivals

This paper is strongly focussed on forecasting call arrivals which is not necessary for this project. Since forecasting the future isn't of interest. However, it gives some insight in how to model call arrivals. Calls exhibit intraday, daily weekly and monthly seasonality's. The author proposes and extension of the arrival process, nonhomogeneous Poisson process with a time varying arrival rate function, with timeslots per 15-30 minutes.

Another point is that due to the Poisson modelling assumption the number of arrivals is equal to the expectation. However, it is observed that that variance in the arrival rate is usually much higher than represented by a Poisson process. The simulationist should therefore use a doubly stochastic arrival process. This is a Poisson process where the arrival intensity is a stochastic process as well.

ARMA & ARIMA

In order to make better forecasts of call arrivals the author proposes to use Autoregressive Integrated Moving Average (ARIMA) or ARMA Models. These models are used to model the future call arrival rate. Based on the data from past to present AR(I)MA is able to predict future data. The author discusses various variations of the AR(I)MA models because these allow dynamic updates of the variables this allows to model seasonality's that occur in call centers. The deviations of AR(I)MA allow for accurate forecasting of future calls with presumably small root mean square error. The reason for the presumption is that the total number of calls is not given.

Staffing and Routing

[2]. Exploiting Simulation for Call Centre Optimization.

This work presents a detailed description of how call centres operate as fundament for DES modelling, and showcases a simulation experiment to support claims. The author goes in depth about call centre dynamics and how multi skilled agents can be employed to deal with overflowing queues. In multi skilled call centers agents with a certain skill handle a certain type of calls, these agents can be grouped per skill. When a set of agents handles a set of skills they can be strategically pooled such that when all agents in a skilled group are busy less skilled agents or agents from another group can handle the overflow of calls making call centers more efficient. One important notion of the author is that agents work faster when dealing with a smaller set of skill types. During the simulation experiment the author focusses not on optimizing agent scheduling but on routing techniques and utilization of multi skilled agents.

The author of the paper proposes 5 different scenarios for call routing, however only two scenarios are tested in this work. One of the scenarios can be seen in figure 2,. which shows a popular form of skill based routing. In the routing setup there are two queues where calls arrive, each queue is handled by a group of specialists who have a lower call handling time. The two queues are also served by an overflow group, calls arrive here when both group 1 and 3 are busy. The overflow group consists of generalists who assumably have a longer handling time because of a more diverse the skillset the slower the agent operates.

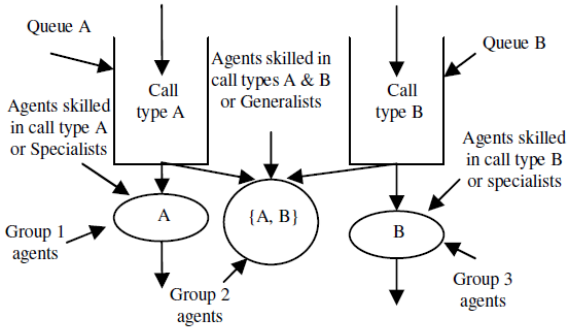

Figure 2, Skill-based routing with overflow group

Simulation software

[6] *A Java Library for Simulating Contact Centers*

The Java simulation library is written as a master thesis by a student of prominent simulationist P. L'Ecuyer. It contains all the building blocks to code your own event driven call centre simulation. The package offers queues, agent groups, simulation event queues and arrival processes. The code can easily be edited by overriding the java classes. The advantage of the simulation lib is that you have total control in contrast to complete simulation solutions. This gives the developer the freedom to implement any call centre design. According to the author the simulation speed is highly competitive with other simulation solutions.

Experiments

[9]. *Designing Simulation Experiments*

This paper describes how Design of Experiments (DOE) can be applied to simulation studies. DOE is a method where a regression model is used to identify the relationship between factors and the contribution of individual factors towards the performance measure.

In order to setup the experiment first all factors (variables in the regression model) need to be identified. This can be done with the use of a cause effect diagram as can be seen in Figure 3.,

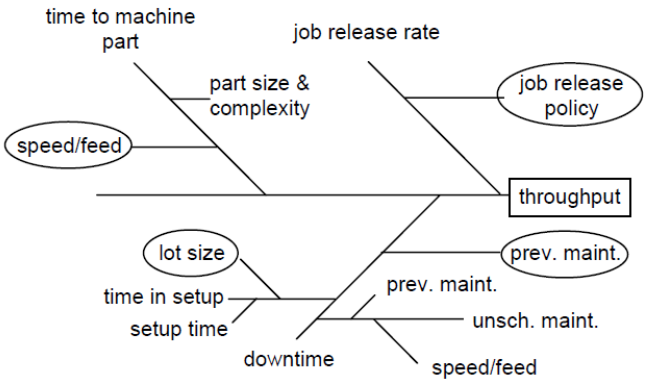

Figure 3, Cause effect diagram of a job scheduling simulation

The parameters can be classified as four different parameters: Independent variables these result in system performance; Dependent variables are typically the output parameters; Nuisance variables are uncontrollable parameters that affect the system, these should be known as they can explain random variation; Intermediate variables cannot be controlled as an independent variable as they are a product of the independent variables, but cannot be classified as dependent variables. They should be identified so they are not mistakenly classified as independent variables.

The leaves with circles are independent input variables and without circles are nuisance variables. If figure 3 were a tree the root would be the dependent variable and the branches would be the intermediate variables. The idea is that once all variables are identified a regression model can be made. One drawback, creating a rich model would mean that all combinations of input variables need to be simulated (to generate the data needed for a regression model). Due to the possible infeasibility of simulating all configurations fractional factorial design can be employed to effectively lower the amount of input combinations. Which is done based on the number of factors and the available time for experiment runs.

Once the fractional factorial design is made and all combinations of input parameters are known then simulation runs can be made. This gives us all the input needed to establish the model, the impact of each factor can be estimated by a statistical library, given that the factors chosen are correct. The regression model can be analysed to see what the impact is of certain factors or relationship between factors.

[10] Analysing skill-based routing in call centers using discrete event simulation and Design Experiments

This paper is selected in order to gain a deeper understanding of the DOE. In this paper the author attempts to learn more about skill based routing and complete resource pooling. This occurs when there are no agents idle there are no waiting calls. The way this research is performed is of more interest than the actual topic. The experiments performed can be translated to this research. The paper details the classifying of factors, and how a synthetic environment can be verified by performing some common simulation experiments. An important takeaway from this paper is that not entire cycles of operation have to be simulated for instance, when performing regression method, it is not necessary to simulate all the days of the month but simulate single days with varying arrival rates.

Latest research

[13] *Blended call center with idling times during calls*

This paper explores the possibility of scheduling jobs in between other jobs. In the case of this paper that means that in certain call centers calls have a period where the caller has to perform some activity and the agent has to wait for this activity to finish. This could be the case that the caller has to restart his/her modem. This provides the opportunity for the agent to perform some task like an outbound call or answering an email. The problem here is: When can an agent perform such a task and when does this become infeasible with regard to the service levels?

To study this problem the authors modelled a single-server and multi-server call center where each call that arrives consists of 3 stages: Stage 1 the agent and customer perform an initial conversation; Stage 2 the customer is to perform some task with a stochastic time, within this period the agent is able to perform some task; Stage 3; Starts when the customer or the agent finishes its intermediate job. In this stage the conversation is finished. Agents can also serve outbound jobs outside of calls when there are no waiting calls.

The optimal policy to solve this problem depends on multiple factors which makes it difficult to make one policy that solves this issue to optimality. Instead the authors choose to go with a probabilistic model. The probabilistic model holds two variables p and q.

Between calls there is a chance of probability p that agent performs an outbound job, or 1 – p that the agent does not perform an outbound job.

Inside call: there is a chance of probability q that the agent performs an outbound job, or 1 – q that the agent does not perform an outbound job.

The authors seek optimal values of p and q to optimize the amount of outbound jobs under service level constraints. For the single queue study the authors perform a Markov chain analysis on the performance measures. With the Markov chain analysis, the authors are able to describe the system states and then calculate the steady-state probabilities. The reason for the usage of the Markov chain analysis is that it is computationally tractable. From here on the authors test four cases were p and q values are varied and derive cases in which high (low) p and q values are desired. Since an instance where multiple queues are present can't be modelled with Markov-chain analysis (multi-queue systems are typically computationally intractable). the authors simulate the multi-queue environment to see if their claims from the single server environment still hold.

Conclusion

As conclusion to the literature study follows a summation of all subjects that will be used in this study.

- Doubly stochastic and time varying arrival rates [3].
- Design of experiments with fractional factorial design [9]
- Various routing algorithms [2,3]
- Models including shrinkage [3]
- Multi queue call center models should remain small for computational tractability [8]
- The java simulation library [5] for programming the simulation.

These are the most notable subjects to include in this research however, this does not necessary exclude other takeaways that can be described as "do's and don'ts".

Research question

The goal of this research is to find out if the handling of emails by call centre agents has a notable impact on the workload in a call center. If this is the case this changes the way how forecasting should be handled in call centres and how many agents should be handled. By performing a simulation study, we expect to gain insight in the problem environment and learn how the impact of email corresponds to different strategies, and constraints. As well as how different existing call routing strategies can be applied to the routing of email.

Research Question

"How can discrete event simulation be used to understand the impact of email handling in contact centres on output parameters?"

Sub Questions

- "How can a multi-channel call centre be modelled as a discrete event simulation?"
- "How can different routing policies be used and compared in a discrete event simulation?"
- "What routing policies are relevant with respect to email handling?
- "How can discrete event simulation be used to identify the relationship between input and output parameters?"

A synthetic simulation model established from real data will be employed to test different scenario's. The simulation model will be synthetic in the sense input distributions originate from real data but are to be scaled to fit simulation experiments.

Routing techniques will be tested to explore different methods of email handling, as well as online scheduling where email handling can be interrupted by phone calls. Various configurations will be evaluated to find out how output parameters react under different circumstances. Afterwards simulation results will be analysed to answer the sub questions. Once all sub questions are answered, sufficient information should have been gathered to answer the main question.

Relevance

In this section the relevance of this study will be elaborated on.

Call centre simulation has been a popular topic in simulation studies for many years. Many papers have been released during that period. Recent studies however, seem to have shifted focus towards regression methods aimed at forecasting current and future call volume. As technology changes so do call centres which now more commonly take the form of contact centres. In a contact centre a service is provided over multiple communication channels (Multichannel), for instance over chat and email. While email is not a new concept in this service industry, these channels are becoming the norm in call centre solutions.

During that time little to no research has been performed aimed at identifying the implications caused by email handling in call centres, making this research project more unique. Gaining insight in how email works will make it easier and more accurate to model multichannel contact centres and other instances where email is modelled.

If it is the case that the volume of incoming email is big enough that other processes might be disturbed then appropriate action is required. Email brings its own form of customer expectation as the time frame of the expected response is longer than the time someone is willing to wait in a queue. Given the difference in expectation we assume that email cannot simply be modelled as another call queue. Besides quantifying the workload that comes with email, the way email is presented to agents might influence the process as well. Are emails pushed to agents? Or are the pulled by agents? Can agent email handling be interrupted by an incoming call? These are all relevant questions when manufacturing a call centre solution.

With this application we hope to gain more insight in how the handling of email works and how it could be improved. This is valuable to CC4Skype as they can improve their email integration and or support superior routing strategies. Besides the results of this study this application provides CC4Skype with the possibility to simulate load tests akin to real situations. As well as a platform for testing changes made to the internal mechanisms of the call center software.

Research methodology

In this section the different aspects of this research project are described as well as the characteristics of the dataset. The study itself is a simulation study. This is mainly a quantitative study as many iterations of simulations will yield data over the numerous settings and configurations. The foundation is mostly qualitative as literature will be studied and interviews taken in order to gain understanding of the subject and the problem domain.

Qualitative Research methods for designing a call centre simulation:

- Literature study, best practice solutions for simulation studies, getting to know call centre environments and various configurations for agent utilization.
- Interviews with call centre specialist, developers and other domain experts to establish the fundament of the simulation.
- Analysing call centre infrastructure, to model the call flow.
- Design of a simulation tool based on modular call centre infrastructure.
- Validation of the simulation tool.
- Implementation of the simulation model

Quantitative research for interpreting the simulation results:

- Analysing call centre database, containing the information of calls and agents.
- Analysis of Email data obtained from service desk.
- Analysing simulation results of multiple simulation runs with different agent configurations.
- Running the simulation with various routing policies.
- Comparing experiment results
- Validation of the simulation tool.

Research methodology - The Dataset

Due to difficulties in obtaining a dataset from a call centre that uses email, the dataset and the model will be a synthetic resemblance of a real call center. Real call centre data will be used and scaled to a size that is interesting for research. Since CC4Skype at the time of writing doesn't have any customers using their email integration, the data needed has to be created. CC4Skype has their own service desk that handles calls and emails. For this project the data of incoming and outgoing mails will be extracted from the environment, yielding email arrival times email size. However, this leaves us with the question how long it actually takes to write a response email to technical questions. A program has been developed that measures how long it takes from opening a new email to sending it and how many characters are contained in the answer. This data will

Project plan

The book simulation modelling and analysis [7] discusses the steps of a ‘sound simulation study’, a ten-step flow chart style plan. The plan is an iterative process which revolves around drawing conclusions and the validation of conclusions. This plan is chosen as a guideline because of the many pit falls that can occur in a simulation study if the model is not correct. The plan has been altered such that it is in line with this project, since the data used in this project is synthetic version of the real environments. The steps of a sound simulation study are as followed:

1. Formulate the problem and plan the study
2. Collect data and define the model (make assumptions document).
    a. Scale dataset call center
    b. Scale Email dataset
        i. Collect Email data.
        ii. Categorize email dataset in multiple categories.
    c. Tie datasets together
3. Validate the assumptions document with problem owner (if not valid return to step 2).
    a. (In the case of this project suitable approach of call center modelling will be discussed with domain experts with reference to literature)
4. Construct a computer program(simulation) and verify (debug the program).
5. Make pilot runs on the program to make a case for validation.
    a. A run without email.
    b. A run with email and telephony.
6. Validate the computer program.
    a. Run a simulation on the current situation of the problem environment.
    b. Review the system with problem owners & domain experts.
    c. Perform a sensitivity analysis on performance parameters.
7. Design experiments.
8. Make production runs.
9. Analyse output data.
10. Document, present and use results (actual thesis document).
    a. Documents the entire system, how is the simulation made, performed, what design decisions were made and what results were achieved.
    b. Present study results, Discuss model building.

Call Center Simulation
- Conceptual Model

In this chapter the conceptual model is discussed, during this phase of the thesis the properties of the problem environment are mapped and identified. These properties are then used to create a simulation that is a correct representation of the problem environment. All stochastic process are fitted to the corresponding distributions, followed by a detailed overview of the simulation event handlers.

Conceptual Model

Introduction

In this document the call center model will be explained starting with an informal description of the call center which serves as an overview the simulation. Followed by a detailed description of the system, describing key factors of the system such as the mechanisms behind agent selection and call routing.

System Description

This simulation model represents a call center that works five days a week. In the simulation both calls and emails are handled by the agents. This call center is a synthetic representation of a small call center however this only serves as a study example. Arrival rates and the number of agents will be modified during the experiments.

The system should be able to support switching between the multi skill and single skill configurations. This means that some agents are able to handle multiple jobs while others do not. The number of active queues should be variable.

The simulation works in the following manner, the call center opens at 8'o clock. At this time calls are accepted in the system. Calls arrive and depending on their call type, each call is routed to a specific agent group(s) or queue. Each queue handles a different call type and each queue is operated by at least one agent group. If all agents serving the queue are busy the call is placed in a waiting queue. The queue is a FIFO (First in first out) system and when an agent becomes idle, the longest waiting call in the queue is removed from the queue and routed to the most competent agents. The call is then handled by the agent, the time this takes depends on the call type's handling time distribution and a fixed time (10 seconds) for the agent to log the proceedings of a call. Once the call is handled the agent status changes to idle and if calls remain in the queue the agent continues serving calls or emails.

Emails enter the system the same way as calls enter the system but can be handled in a more flexible manner. Depending on the experiment this flexibility will be exploited. Emails can be handled with or without pre-emption. This way the work on email can be interrupted to answer more urgent phone calls.

At five o'clock the call center stops accepting calls and remaining active and enqueued calls will be finished. Calls may still abandoned. Unfinished emails will be handled the next day(s).

Detailed Operation description

- Running time
  - The call center is open from 08:00 till 17:00.
- 2 queues for calls and 1 queue for email depending on the experiment 1 or 2 queues for calls will be used.
  - Both call queues are obtained from the modelled environment, each with different handling times.
- Schedule ~ 8-15 – 4 - 5 agents , 15-17 - 2 agents.
- Call arrivals
  - When a call enters the system the router first checks if there are agents available to handle that call. If there are no agents available the call is placed in a queue based on the corresponding call type. The arrived contact has a patience time, when the time spent in the queue exceeds the patience threshold the contact abandons. If there are agents available to handle a call of this call type the longest idle agent is selected and starts the call.
  - (**When the interruption of email is allowed**) If there are no idle agents available an agent working on email is to put its email on hold to handle this call.
- Email arrivals
  - When an email arrives in the system and there are no waiting calls and agents available the system will route the email to the longest idle agent capable of handling email. If there are no available agents the email is placed in the email queue. Without patience time.
- Call routing
  - In CC4Skype calls are normally routed to the agent with the longest available status in skype, status change in skype can be manipulated if for instance an agent has a meeting he/she can change her status. However the simulation does not take other work then answering calls into account.
  - Agent groups can be set up with overflow routing. This means that one agent group primarily serves queue 1 but can serve queue 2 with a lower priority. If queue 2 overflows agents of queue 1 can serve calls from queue 2. Each call type can be configured with a rank for agent groups. When all agents in the highest ranked agent group are busy the system looks for a free agent in the group ranked second and so on. This allows for complex routing policies where important queues have higher priority over other queues.
  - CC4Skype is queue based instead of agent group based meaning a queue holds a number of agent connected to the call type and agents can be connected to several queues. The simulation is agent group based but can be translated to queue based routing by assigning unique agent queue relations to agent groups.
- Call types
  - The system allows for numerous call types and even email; each call type arrives at its own queue and is picked up by an agent group which is enabled for this type.
- Schedule change
  - Agents can be scheduled per hour (this can be less). every hour the period change event checks if the number of agents is equal to last period, and if agents need to be removed or added.
  - If the number of agents decreases in the new period the agent with the earliest log-in time that is also idle is removed. If there are no idle agents, the agent that has the earliest login time regardless of being idle is set as a ghost agent. Once the ghost agent

finishes the ongoing service and any email the agent has been working on, the ghost agent leaves the system

- Ghost agents
  - Ghost agents are agents that have been scheduled to leave the system once they finished their ongoing services.
- Queues
  - CC4Skype queues hold no real capacity limit because it depends on the resources of the machine. However CC4Skype queues are performance tested with 100 waiting calls. For this reason the queues in the simulation hold no limit on the number of waiting customers.
- Multi-Tasking costs
  - When an agent handles more than one task at a time (email interruption by a call) or handles tasks in rapid succession so called switching costs occur. This is extra time needed to perform two tasks in succession instead the summed time of performing the tasks separately. The switching cost increase as the complexity of the tasks increase (2001 Rubinstein et al). This makes it difficult to account for correct switching costs in the simulation model. However taking switching costs out of the model completely seems to do more harm than simply increasing the remaining handling time. Thus when an email is resumed after a call, the handling time is increased by 20% of the initial handling time (due to a lack of a better alternative). 2001, Joshua Rubinstein, PhD, Jeffrey Evans, PhD, and David Meyer, PhD,

Input data/distributions

The section bellow lists all the aspects of the simulation that can/should be varied in order to examine and compare the outcome of different experiments.

Input Data

- Number of agents
- Number of agents assigned to email
  - Either entire agent groups handle calls and emails where email has a lower priority. Another scenario could be where the agent group is divided and a subset of the agent group has the lowest priority for calls but also handles emails.
- Call and mail arrival rate
- Number of queues / Number of call types
- Pre-emptive scheduling Yes/No.
  - The term pre-emptive scheduling originates from typical job-scheduling where jobs can be interrupted and continued in order to serve higher priority jobs.

Distributions

This section details for each of the stochastic processes in the simulation what distribution best fits the data. For the first distribution, call handling time and the entire process of decision making has been documented. This has been omitted for the rest of the distributions considering relevance and the time consuming nature of the process.

Call Handling time

There are 77000~ records available for the handling time. The data was analysed using R studio. In order to get an idea of the underlying distribution the data density was plotted using R's plotdist function. Looking at *Figure 1*, a Log-normal/Weibull/Gamma distribution might be appropriate. To get more insight into which distribution might come close to our data we plot a Cullen and Frey graph as in *Figure 2*. The Cullen and Frey Graph is a map of the $(Skewnes^2, Kurtosis)$ space.

Where skewness and kurtosis are both descriptors of a probability distribution. Skewness describes the asymmetry and kurtosis the tail. The skewness and kurtosis are estimated by taking bootstrap samples from the dataset.(Delignette et al, 2014) Based on the skewness and kurtosis of the data the Cullen and Frey Graph summarizes how these properties compare to those of several distributions.

As can be seen in the cullen and frey graph, the data is observed to be close to the Log-normal distribution. We then try to fit the data to Log-normal, Gamma and Weibull distributions using R's fitdist function.

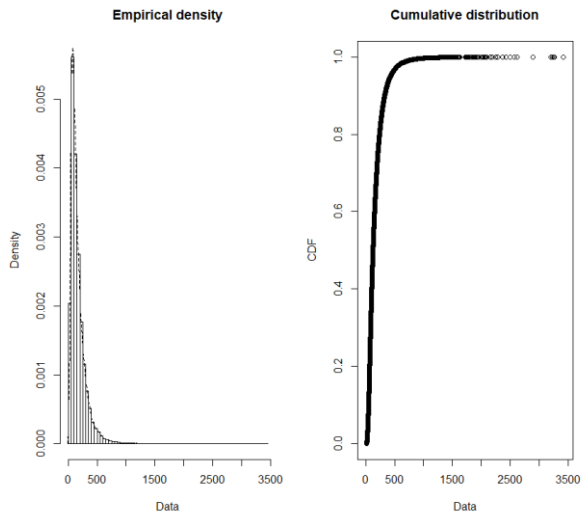

Figure 1. Distribution density and cumulative density of the handling times

For each data set to be fitted to a theoretical distribution a set of graphs as in *Figure 3,4 & 5* will be plotted. These *Figure*s show four different graphs that indicate how well a theoretical distribution fits the data. The empirical and theoretical density shows the relative likelihood of a value occurring in the dataset. The Cumulative distribution function(CDF) gives the probability that a random variable of this distribution is x or less. The Q-Q plot is made by dividing the data in to quantiles and then match the distance of the theoretical quantiles and empirical quantiles. The P-P plot is the comparison of the empirical and theoretical distribution functions.

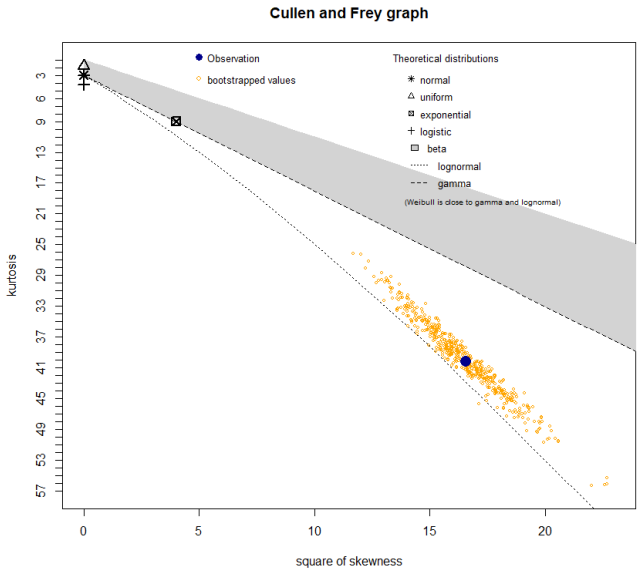

Figure 2. Cullen and Frey graph of all the handling times over 2017

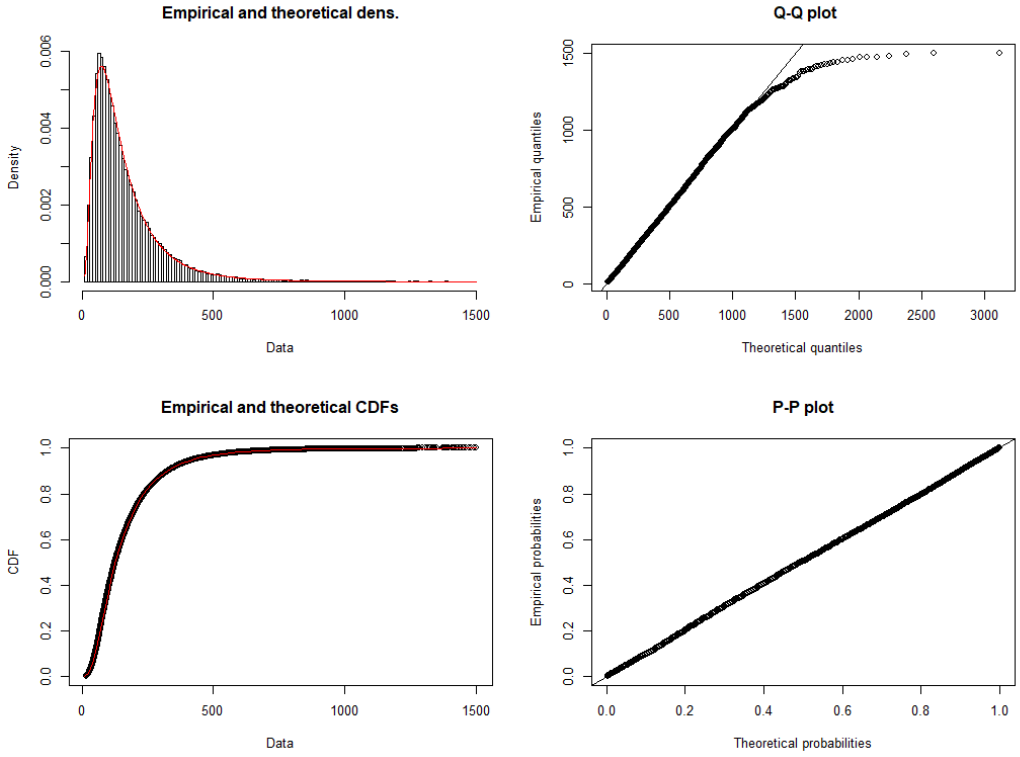

Figure 3. Plot of the fitting of the lognormal distribution to the data

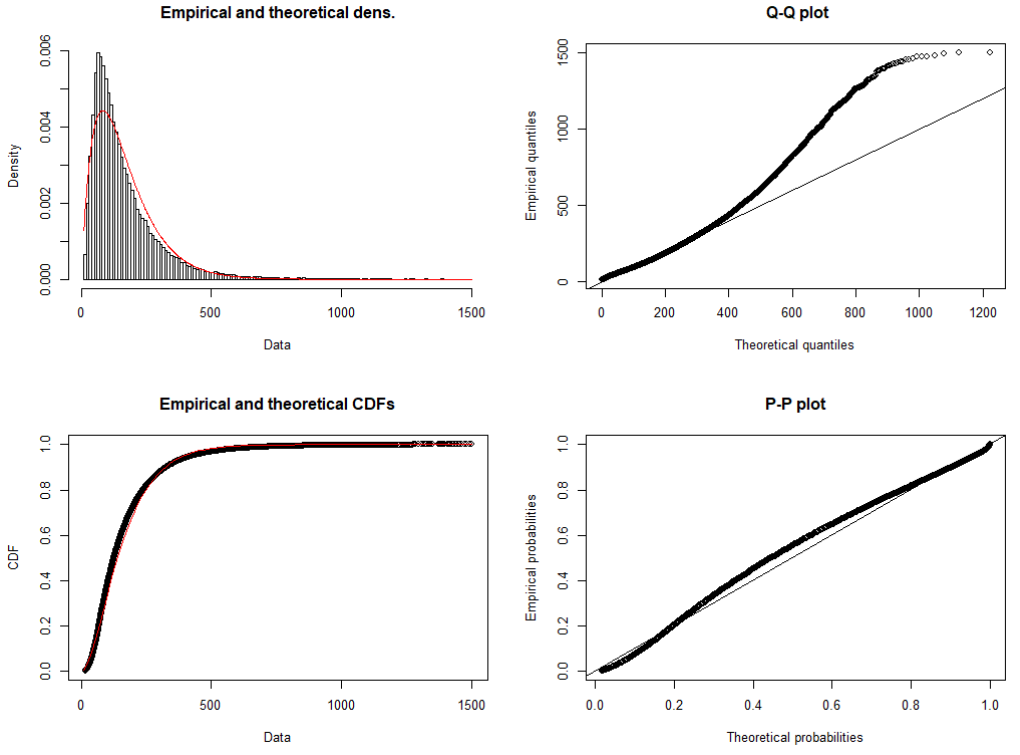

Figure 4. Plot of the fitting of the gamma distribution to the data

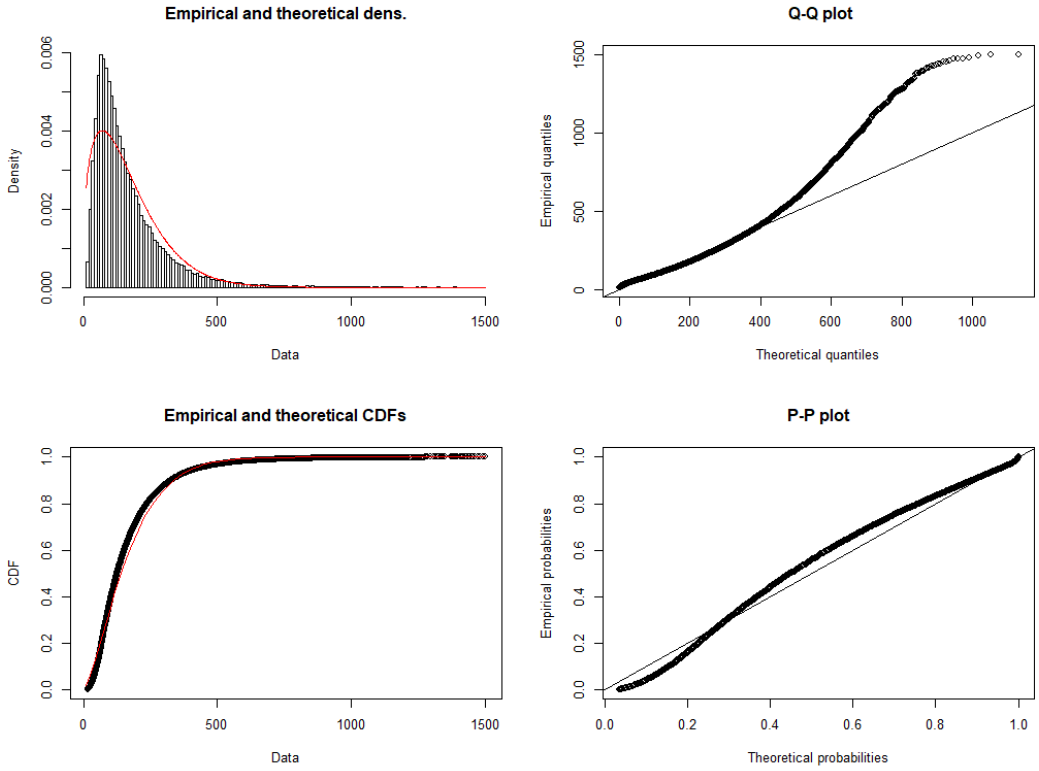

Figure 5. Plot of fitting the Weibull distribution to the data

Goodness-of-fit statistics	Log-normal	Gamma	Weibull
Kolmogorov-Smirnov statistic;	0,00999	0,05563	0,06361
Cramer-von Mises statistic	1,20816	87,29133	133,69484
Anderson-Darling statistic	6,70629	507,94758	920,21517
Goodness-of-fit criteria	Log-Normal	Gamma	Weibull
Akaike's Information Criterion	916802,50	923688,80	929422,80
Bayesian Information Criterion	916821,00	923707,30	929441.3

Table 1. Goodness of fit statistics of call handling times over 2017

As a result of the Cullen and Frey graph, R was used to fit the data to a Weibull, Gamma and Log-normal distribution as can be seen in *Figures 3,4,5*. The plots already give a strong indication about the goodness of fit for the Weibull and the Gamma distribution. The density functions both have a much lower peak than the observed data.

For those same fitted distributions goodness of fit statistics and criterion are calculated. The statistics correspond to three goodness of fit tests, where a lower statistic is generally better as for each test this is a measure of distance towards the distribution. These three fitness tests were chosen because each has pros and cons. BIC(Neath et al, 2012) and AIC *(Akaike, 1974)* measuring the likelihood of fitting the distributions to the data, these measure the entropy (chaos) in the system where less entropy is always preferred .

From *Figures 3 ,4, 5* and *Table 1* we can draw the conclusion that the log normal distribution best fits our data. However when performing ks.test (Kolmogorov-Smirnov test inR) we get a p-value of 9.739e-07. If we take the logs of the handling time and perform an Anderson Darling test (ad.test in R, a test for normality)(Delignette-Muller et al, 2015) we get a p-value of 7.791e-09. The same goes for a Cramer von Mises test (cvm.test) we obtain a p-value of 0.0008067.

The reason for this could be monthly seasonality of calls where the handling time of calls varies.

In order to successfully fit the data to the log-normal distribution we take the observations from February instead of the entire year. This reduction of the dataset leaves us with 6018 observations, see *Figure 6*. In order to get a significant p-value outliers above 1100 seconds have been removed these are 21/6081 calls. Calls shorter than 15 seconds are also removed as we assume no conversation that short seems realistic. We obtain the results as seen in *Table 2*.

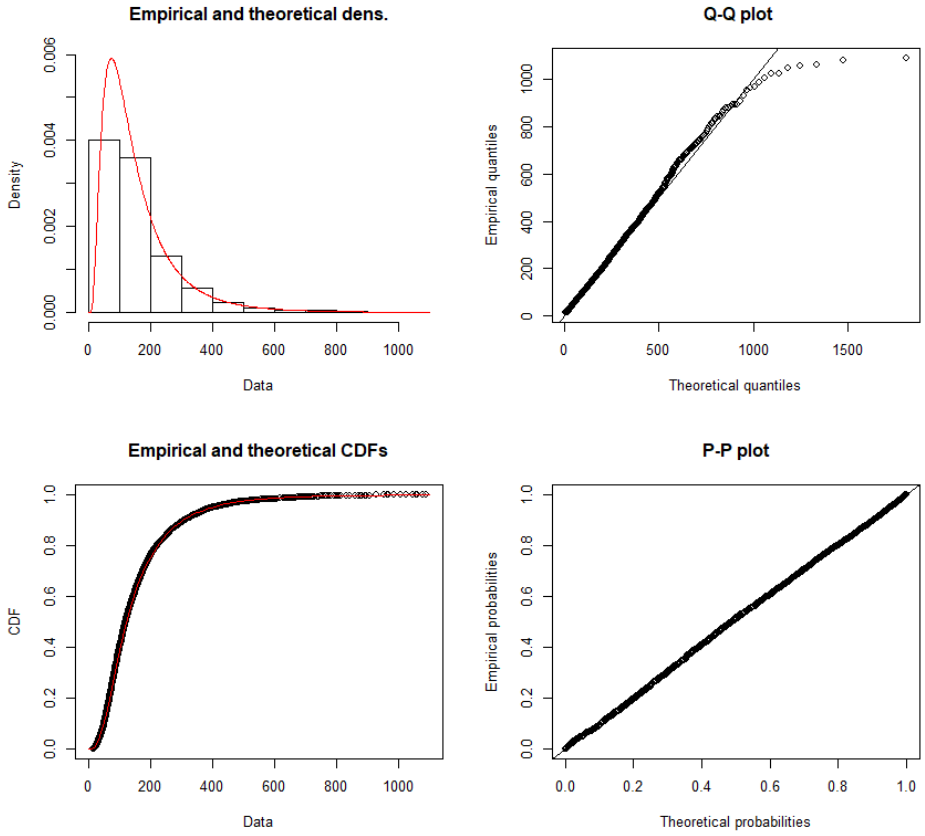

Figure 6. Plot of the fitting of the log-normal distribution to Handling times of February.

Mean	4. 8018632	
Sd	0.7168358	
Test	Statistical value	P-value
Anderson Darling	1.2102	0.2637
Cramer von Mises	0.20087	0.2658
Kolmogorov (approx.)	0.04544	0.1373

Table 2. Outcome tests of log-normality for call handling times in February

Figure 6's Q-Q plot indicates that the handling times of the theoretical distribution is more skewed than the empirical data. The result of this is that the maximum values generated are possibly higher than the maximum values occurring in the dataset.

From the results of *Table 2* we can conclude the distribution function passes the Anderson Darling, Cramer von Mises and approximation of the Kolmogorov-Smirnov test with a P-value > 0.05.

Abandonment

For the coming distributions the steps and methods of finding out which distribution closely fits the data have been purposely omitted.

To find the distribution related to call abandonment we initially took the data over 2017. In order to fit the data to the log-normal distribution (which it resembles) waiting times shorter than 15 seconds and longer than 3000 seconds have been removed. *Figure 7* displays the outcome of the fitting with various plots. Although it seems like the log-normal distribution is the right choice of distribution it didn't fit well enough according to the Anderson Darling test.

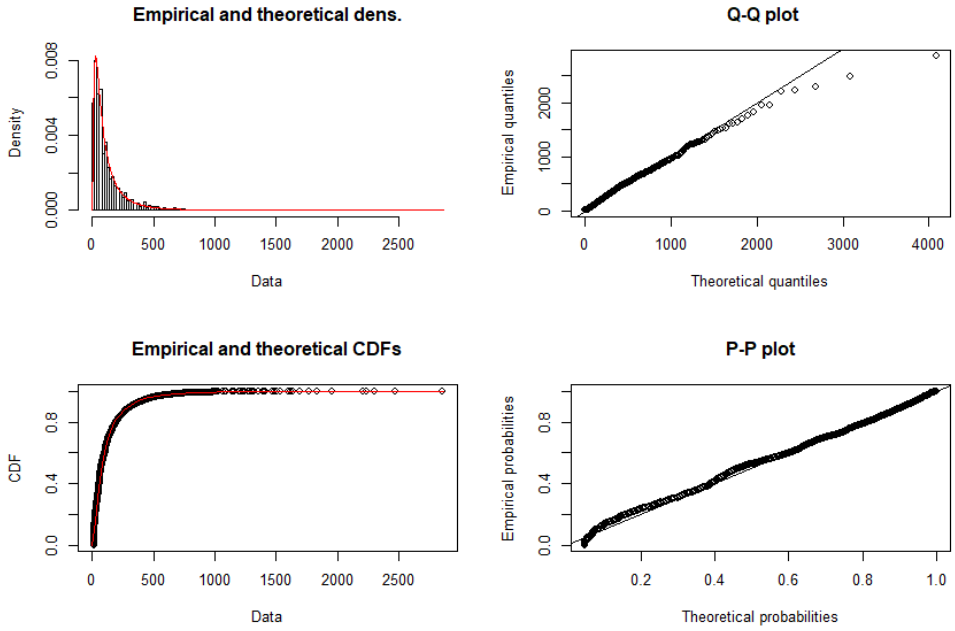

Figure 7. Fitting the patience time for call abandonment over 2017 to a log normal distribution.

To fit the data of the abbandond calls to a distribution only entries from the month Febuary were used and outliers above 2500 and below 10 seconds have been deleted. The results of the fitting can be seen in *Figure 8* and *Table 3*.

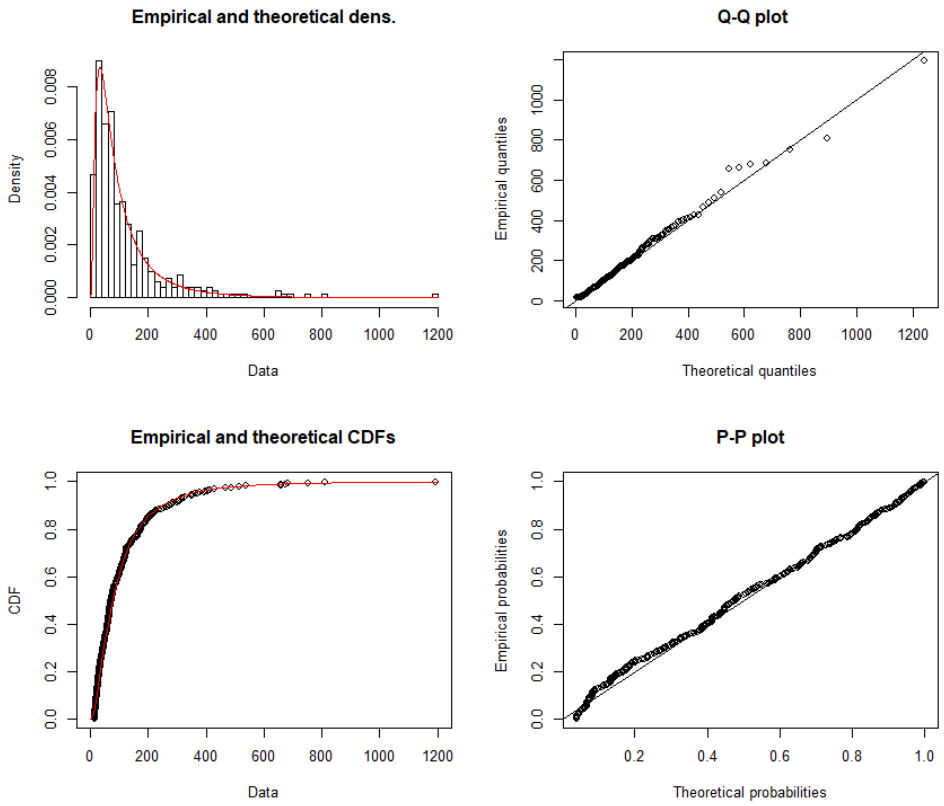

Figure 8 Goodness of fit log-normal plot abandonment February 2017

Mean	4.3277150	
Sd	0.9256815	
Test	Statistical value	P-value
Anderson Darling	1.2243	0.2585
Cramer von Mises	0.13127	0.4522
Kolmogorov (approx.)	0.04544	0.3869

Table 3. Outcome tests of log-normality for call abandonment in February.

Table 3 shows the fitting of the Log-normal distribution which passes the Anderson Darling, Cramer von Mises and approximation of the Kolmogorov-Smirnov test with a P-value > 0.05.

However not all customers leave the queue as most of them are actually patient enough to wait until they are served. To arrive at a reasonable estimation we look at the calls that wait longer than the median of the abbandoned calls, 71 seconds. We divide the number of abbandoned contacts over the total. This is a probability of 237/1520 = 0.155.

Multi-Queue scenario (2 queues)

The modelled contact center turned out to have multiple call queues. The first section of distributions describes the handle time of the total amount of calls arriving at the contact center. This section describes the fitting of the two individual queues. The second queue will only be used if experimentation demands it.

A multi-queue scenario is where contact center simulation gets interesting because this quickly becomes mathematically untractable via Markov Chain analysis (Koole et al, 20), although small contact centers are tractable through queueing models as can be seen in[17] (T.A. Kota et al, 2017). In a multi-queue scenario there are multiple queues and one or more agent groups serve these queues.

For both datasets we removed calls with a handling time of less than 15 seconds.

Queue 1

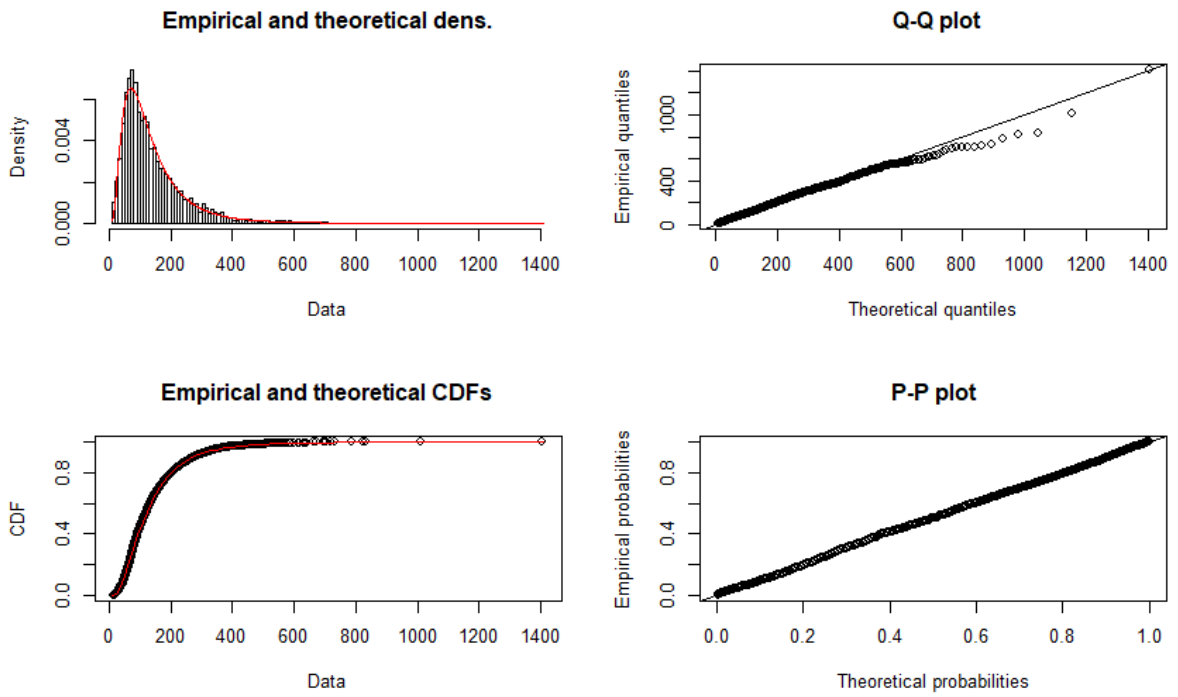

Figure 9. Goodness of fit log-normal plot Call handling time of queue 1, February 2017

Mean	4.7259060	
Sd	0.6813851	
Test	Statistical value	P-value
Anderson Darling	1.2984	0.2329
Cramer von Mises	0.18364	0.3015
Kolmogorov (approx.)	0.018275	0.09296

Table 4. Outcome tests of log-normality for call handling times of queue 1 in February.

Even though *Figure 9* shows that the theoretical distribution won't generate the same peak the data shows in the density plot the log-normal distribution fits significantly with P value > 0.05.

Queue 2

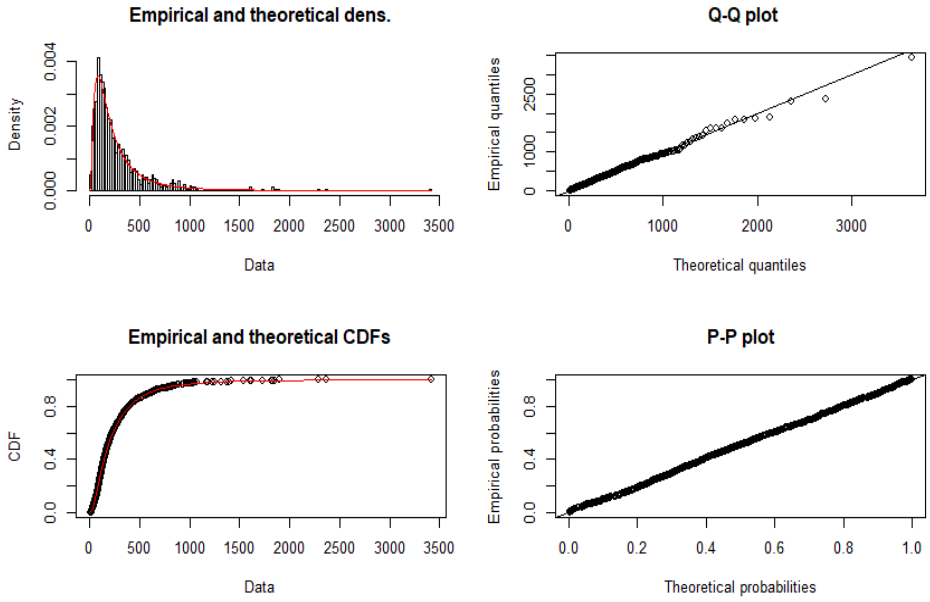

Figure 10. Goodness of fit call handling of queue 2, February 2017.

Mean	5.2248035	
Sd	0.8993385	
Test	Statistical value	P-value
Anderson Darling	0.44652	0.8016
Cramer von Mises	0.053877	0.8525
Kolmogorov (approx.)	0.022184	0.6768

Table 5. Outcome tests of log-normality for call handling times of queue 2 in February.

With a P value > 0.05 we fail to reject the null hypothesis that the data is log-normally distributed.

Email handling Time

The data is obtained of two helpdesk emloyees from CC4Skype by logging the time it takes to open and respond to an email. The program that logged the emails recorded the time that the email is opened by the employee until the email was sent. It contains 900~ records. The raw data (uncleaned) has a strong resemblance to the log-normal distribution.

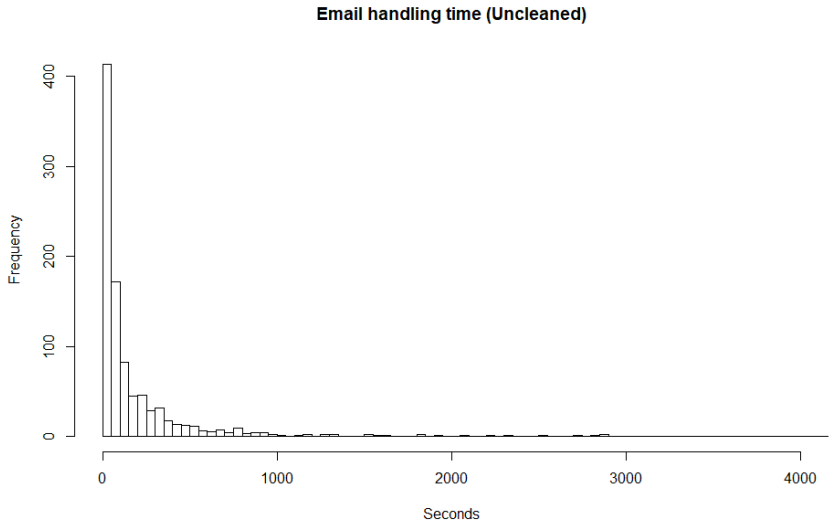

Figure 11. Density plot for Email handling Time.

As can be seen from *Figure 11*. the majority of the emails is handled within a few seconds. Which can be an indicator that an email is forwarded or a very short response to a very short email, I.E. a confirmation email. We Assume that it is not realistic to spent less than 20~ seconds to answer an email. Therefore we cleaned records > 10, 15, 20 seconds and some of the larger records. However we were unable to fit the cleaned data to any distribution (Log-normal, Exponential, Gamma, Weibull, Pareto, Burr, Log-Logistic). The uncleaned data however was easily fitted to the Log-normal distribution with P-values bigger then 0.05 as can be seen in *Figure 12*.

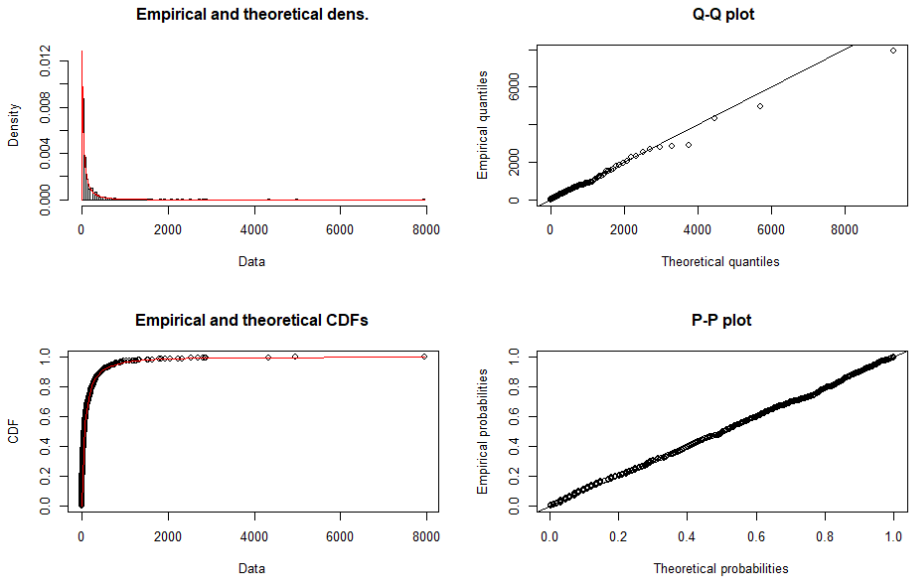

Figure 12. Fitting email arrivals to the log-normal distribution

Mean	4.159607	
Sd	1.521251	
Test	Statistical value	P-value
Anderson Darling	0.4154	0.8333
Cramer von Mises	0.052884	0.8587
Kolmogorov (approx.)	0.025026	0.5948

Table 6. Outcome tests of log-normality for email handling times.

In order to keep working with this distribution the outcome of the stochastic value generator will be filtered. For every number generated lower than 15, there will be a replacement generated. This however will result in a shifted mean in the overall data. In order to overcome this, multiple datasets containing stochastic values have been generated and compared in order to find a parameters able to compensate on the shifted mean values. Based on the mean the log-normal parameters will be tweaked.

Name	Min	1st Qu	Median	Mean	3rd Qu
cleaned	15	40,5	87.0	188,5	233.0
Generated with replacement	15	39,23	86.21	188,6	209,28

Table 7. Comparison mean values when generated with replacement

Table 7 depicts the comparison of the cleaned data cut off between 15 and 2000 seconds and the outc ome of ten million entries created with a log normal distribution with slightly shifted parameters mean log 4.153607, sd log 1.515051.

Updated Email arrivals

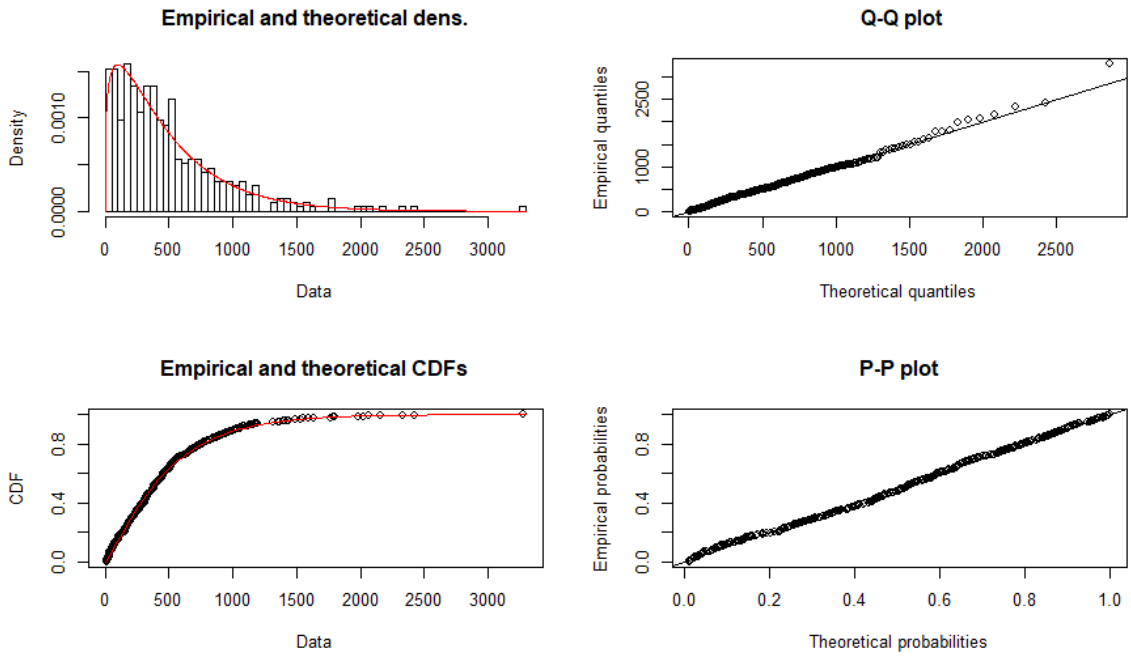

Figure 13. Fitting email handling times to a gamma distribution.

The data for the email handling time comes from presumably a large call center which handle 14000 emails in a month. The dataset was obtained through CCMath a company in the Netherlands that specializes in call center simulation. The reason for this is that the data collected at CC4Skype wasn't production data logged by the system. Instead it was collected by an application listening in on outlook, logging the time it takes for a support agent from opening till sending an email. This happened not without error, on some occasions agents could read the email before logging started.

Cleaning of the data wasn't deemed necessary since the lowest value in the email dataset is 12 seconds which is close to the original cutoff point. The highest occurring value in the dataset is 3278 seconds. Close to an hour but this doesn't seem unheard off .

Shape	1.249448140	
Rate	0.002578327	
Test	Statistical value	P-value
Anderson Darling	0.529120	0.7172
Cramer von Mises	0.076627	0.7118
Kolmogorov (approx.)	0.034814	0.6717

Table 8. Outcome tests goodness of fit test for fitting the email handling time to the Gamma distribution.

Table 8, shows that we fail to reject the null hypothesis with a p-value > 0.05. *Figure 13* shows that the data shows some extreme values compared to the gamma distribution. This doesn't seem too bad for the simulation because as the chance seems low for this extreme value to occur in the dataset.

Period	8	9	10	11	12	13	14	15	16	Outside business
Arrivals / hour	0.6378	1.7696	1.9684	2.002	1.6566	1.7126	1.8862	1.9222	1.5664	1.4718

Table 9. displays the original hourly email arrival times.

Assumptions

- Callers do not hang up before entering the queue.
- Call transfers are omitted, if a call is transferred to another agent it means that the call continues but just another agent becomes busy and another agent idle.
- Emails can be handled in idle time unless policy forbids it.
    - If email is interrupted 20% of the initial handling time is added to the remaining handling time, by lack of a better alternative. This method takes the "complexity" of the email into account (complexity is measured in handling time).
    - Prolonged email handling time cannot exceed the initial handling time.
- Emails sent overnight arrive at the start of the workday.
    - Emails sent overnight shouldn't penalize the service level, and therefore arrive at the start of the workday.
- The acceptable waiting time for an email is 4 hours.
- Answering an email takes at least 15 seconds.
- Answering a phone call takes at least 15 seconds.
- Arrivals are assumed to be a Poisson process.
- After analysis call handling time is assumed to be Log-normal distributed .
- After analysis email handling time is assumed to be gamma distributed .
- Call and email arrival rate is assumed to be steady over periods of one hour.

Run length

Experiment length: Monday – Friday : 8:00 - 17:00

Output Parameters/performance measures

- Number of handled calls
- Number of handled emails
- Average waiting time per queue for calls
- Average waiting time per queue for emails
- Service level per queue
    - For calls the service level is the percentage of calls handled within 20 seconds.
    - For Email the service level is the percentage of emails handled within 4 hours.
- Overall agent utilization
- Number of abandoned calls

Level of Detail

The scope of this simulation is the entire machinery of the call center. This means that agents are assumed to work optimal and call ringing time can be omitted because agents are at their seats. This is because information about absence of agents during work time conflicts with research goals. As absence of agents will most likely interfere with analyzing results of improving a process.

State

- List of Idle Agents
- List of busy Agents
- List of agents busy with Email
- List of ghost agents
- Queues
    - The number of waiting calls/emails per queue and the assigned agent groups.

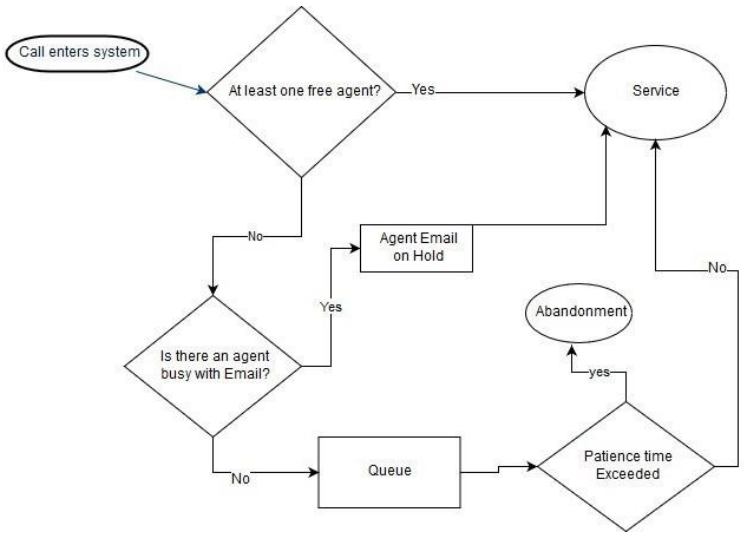

Figure 14. Detailed Flowchart of the life cycle of calls.

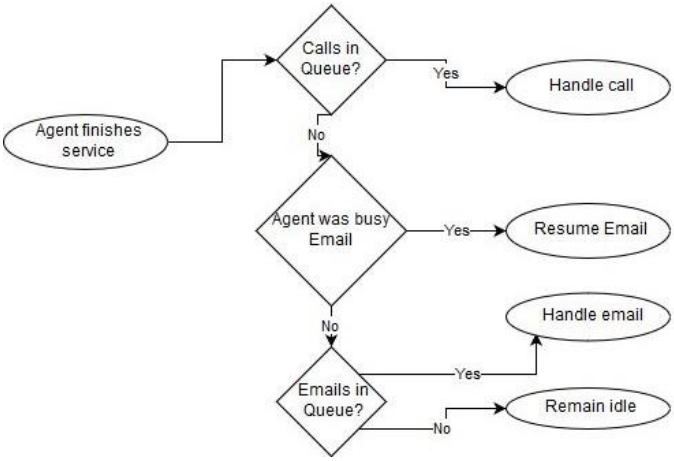

Figure 15. Flow chart depicting what happens once an agent finishes a service.

Events

Event Graph

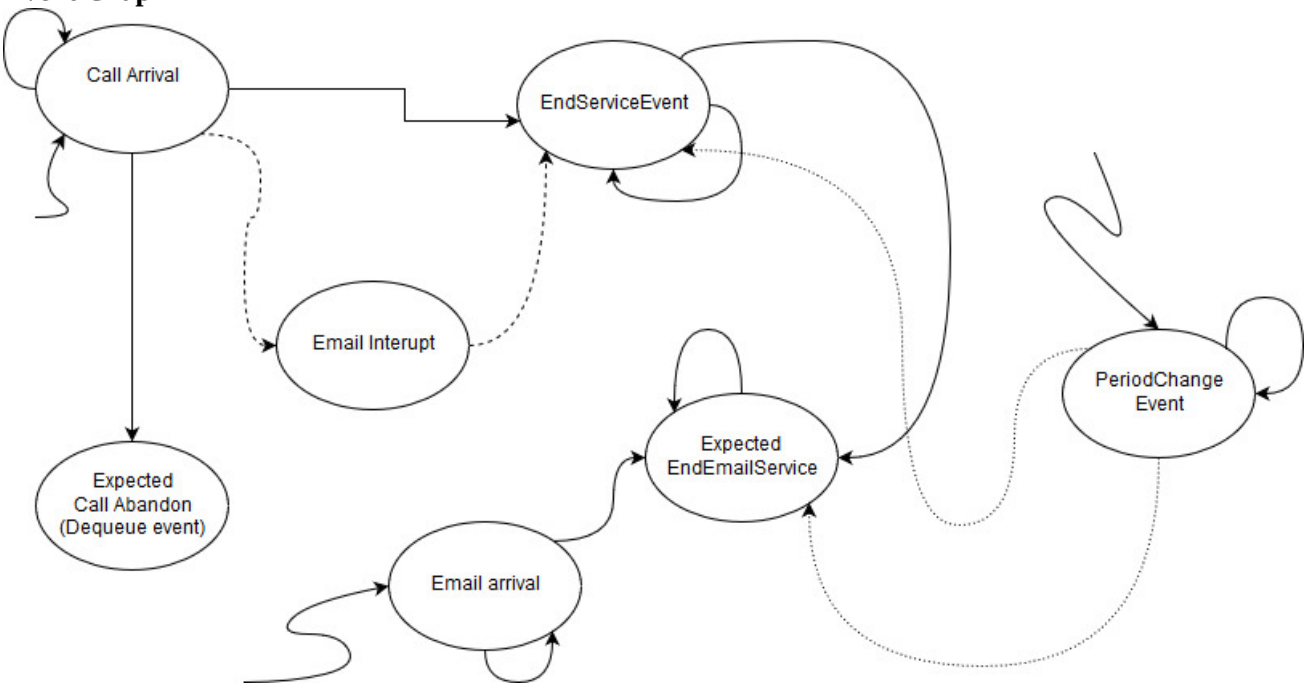

Figure 16. Event graph for calls & emails entering the system.

Call Arrival

A call arrives in the system (*Figure 16*), if one or more agents are available the call will be handled immediately(schedules end Service event). If there is no agent available a dequeue event (call abandon) will be scheduled with an expire time equal to the patience of the caller. If the call is handled before the patience time runs out, the call abandon event will not trigger changes to the system.

Email Arrival

When an email arrives in the system (*Figure 16)* it is handled immediately if there are no waiting calls and no waiting emails. Else the email is queued (without a patience time). Once an agent becomes available and there are no waiting calls (calls always have priority over emails) then a new Expected EndEmailService is scheduled.

If pre-emption is allowed mails can be interrupted by incoming calls if there are no agents available and there is an agent working on email. The email is then put on hold and resumed once the call service is completed and there are no more waiting calls.

When pre-emption is not allowed, the event graph remains the same, except the Email interrupt event and all its arrows are to be removed. Agents start on emails if there is are no waiting calls left. Once finished working on an email the agent goes back to handling calls and emails with precedence for calls.

For a more detailed representation consult the flow diagrams in *Figure 14 and 15*. *Figure 14* shows what happens when a call enters the system. *Figure 15* shows what happens when an agent finishes its service.

Period Change Event
Depending on how periods are defined (number of minutes) the period change event (*Figure 16*) will update the number of agents at every interval. If there are more agents scheduled in the previous period agents will be removed. Vice versa if there are less agents in the previous period agents will be added to their respective groups. The queues will be checked on waiting calls and scheduled to be handled by the idle agents. Once the last period is complete and all ongoing calls are handled the call queues will be reset.

EventHandlers

- Call arrival
  - update arrived calls.
  - Initiates call with an agent if one is available and sets the agent status to busy.
    - This schedules an EndServiceEvent at T_{now} + serviceTime
  - If there are no available agents and there are agents that are handling emails, an interruption event is scheduled immediately for an agent working on email to handle a call.
    - This schedules an EndServiceEvent at T_{now} + serviceTime
    - This marks the email as interrupted. When the interrupted ExpectedEndEmailServiceEvent is executed the event will be skipped and the state of the simulation will not be altered.
  - Initiates a Dequeue event if no agent Available, and adds call to the appropriate queue.
    - This starts a clock on the callers patience time.
- EndServiceEvent
  - Increase the number of handled calls and update service level.
  - If the agent is a ghost agent(see section: Detailed Operation Description) the agent leaves the system. Steps below won't be executed.
  - Set Agent idle if no more calls in the queue.
  - If there are no waiting calls and the agent has an on hold email, then the email will be resumed.
    - This schedules an Expected EndEmailServiceEvent at T_{now} + RemainingServiceTime + JobSwitchTime(initial handlingTime * 0.2)
  - If there are waiting calls, the longest waiting call is removed from the queue and a new EndService event is scheduled.
  - If there are waiting emails, no waiting calls and no on hold emails and a new Expected EndEmailServiceEvent is scheduled.
- PeriodChangeEvent
  - Update number of Agents, longest active preferably idle agents are removed if the number of agents in the next P_{now} is less than P_{-1}.
  - If the agent to be removed is in a call, flag agent to leave the system after service ends.
  - Initiate EndServiceEvent or Expected EndEmailServiceEvent for all idle agents and after PeriodChangeEvent.
  - Close system if workday has ended.
  - If it's the end of the day, clear all the queues. (Except email, depending on experiment setup).

- Email arrival
    - Start emails service if an agent is available and there are no waiting calls/emails and set agent status to busy.
        - This schedules an EndEmailServiceEvent at T_{now} + ServiceTime
    - Initiates a Dequeue event if no agent Available, and adds email to the appropriate queue.
- Expected EndEmailServiceEvent
    - Check if the email is interrupted. If this is the case none of the below actions are performed.
    - Sets agent to idle if no calls or emails are in a queue.
    - Update performance statistics.
    - If the agent completing the service is a ghost agent the agent is deleted from the system.
    - Initiate Dequeue event if there are waiting calls/emails.
- Dequeue Event
    - If the patience time has exceeded remove call from queue and update abandonment statistic.
- Email interrupt
    - Scheduled without delay.
    - Binds the ongoing email to the agent together with the amount of handled time.

Call Center Simulation
-VERIFICATION & VALIDATION

In this part of the document the verification and validation of the call center will be discussed. First a description will be given about how the model was verified. Then the validation of the simulation model will be explained, starting off with the modeled call center. Later on tests are performed detailing the correct execution of email handling. The reason for this is that there is no production data of email available.

Verification & Validation

Verification

The model itself was designed by the author of this experiment who is also a long time employee of CC4Skype. The model was then verified by colleague T. van der Maaten who is a functional and technical support specialist. T. van der Maaten frequently installs the call center software and is responsible for the setup of customer environments such as queues, agent groups and call flows.

Since the simulation program was made with the help of the Java simLib, verification of the call center software was not a major concern, except for the mechanisms where email is involved. To ensure that the simulation behaves as expected to trace files are generated. These detail every event and display the change in idle/busy/busy email per agent group. Furthermore the simulation program is extensively debugged and Java assertion checks have been placed to ensure that changes made to the state by each method equals the desired output.

For each event occurring the trace shows the type of event, how many agents are available at time of the event, the arrival, service, departure and queue time of the call/email related to the event. Lastly it shows the last agent group interacting with the event and the time period at which the event happens. The trace file was checked that every entity that entered the system actually leaves the system at some point.

Call center validation

Introduction

The validation of this particular contact center is a difficult process as 1 to 1 comparison with the real system, will not yield 1 to 1 results. There are two reasons for this, the first we do not know the exact agent schedules. The schedules used in the simulation are inferred from the call handling logs, this makes it difficult to see if agents work the full hour or half. The second reason is that the simplifying assumptions made in the conceptual model are conflicting with the actual system. The absence of shrinkage (unexpected absence of agents) in the model makes it very hard to compare it to the real system. Especially since the modelled call center shows extraordinary amounts of shrinkage.

In order to deliver some sort of validation, arrivals from the original call center are exactly implemented in the simulation. The call handling times of that day are used as an empirical distribution to facilitate the handling times. This "empirical" simulation is then compared to the simulation where all processes are stochastic. Since this simulation is of exploratory nature and not meant to improve/optimize the call center used to create the simulation. This seems like the only reasonable approach to validation given the circumstances.

To avoid confusion, the original simulation model will be referred to as the stochastic model, and the model that uses fixed arrivals and the empirical distribution for call handling times will be referred to as the empirical model.

Evaluation

Evaluation without email

To validate the call center the 6th of February(first Monday of the month) was picked for comparison. *Table 1* shows the per hour statistics for the inferred number of agents, service level and the ratio of calls that are handled within acceptable waiting time. Once its nine o'clock the service level drops disproportionally with the amount of agents and arrivals. This strengthens the assumption that there

is a considerable amount of shrinkage in the call center. However, this could also be because of a series of unfortunate arrival and handling times.

hour	Nr of agents	Service level	Service level/arrivals
8	4	0.875	35 / 40
9	4	0.6842105	26 / 38
10	5	0.7391304	32 / 46
11	5	0.8478261	39 / 46
12	5	0.595238	25 / 42
13	5	0.7346939	36 / 49
14	3	0.2033898	12 / 59
15	3	0.56	28 / 50
16	2	0.25	7/ 28

Table 1. Real call center results of the to be validated day 6th of February

Table 1 will serve as a three way comparison between the real world, the stochastic model and the empirical. To match the 6th of February as close as possible the log normal distribution used for the call handling time was fitted again.

A new (also log-normal) distribution was fitted with mean 4.6584387 and std dev. 0.6905334 which according to a Cramer von Mises test closely fits the empirical distribution with a p-value of 0.9776. *Table 2* shows a comparison between the empirical and log normal distribution.

Well aware of the fact that this does not compare the actual arrival rates and handling time of the entire month of February as initially described in the conceptual model. This method however seems like the only one suitable with the absence of shrinkage and the agent schedules a factor of uncertainty.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max
Empirical	17.0	66.0	103.5	135.2	172.8	1406.0
Log-normal	4.571	66.753	106.407	135.444	169.754	1578.104

Table 2. Comparison of Empirical and newly updated log-normal distribution.

Tables 3 & 4 display the statistics for the number of contacts handled within acceptable waiting time for both the stochastic and empirical model. The first hour looks close to the actual number of calls handled within service level(*Table 1*) which is 35. Unfortunately the rest of the hours are too far off compared to the simulated hours. Using the actual arrival times in the empirical model does enforces the assumption of shrinkage as there is no disproportional drop in service level as can be seen in *Table 1*.

	num obs.	min	max	average	std. dev.	conf. int.	Average arrivals
period 0	1000	21.000	52.000	37.597	5.242	95.0% (37.272, 37.922)	40
period 1	1000	18.000	54.000	36.049	5.349	95.0% (35.717, 36.381)	38
period 2	1000	22.000	65.000	45.073	6.049	95.0% (44.698, 45.448)	46
period 3	1000	27.000	66.000	44.799	6.586	95.0% (44.390, 45.208)	46
period 4	1000	21.000	63.000	41.472	6.045	95.0% (41.097, 41.847)	42
period 5	1000	27.000	65.000	47.276	6.296	95.0% (46.885, 47.667)	49
period 6	1000	10.000	52.000	33.531	7.222	95.0% (33.083, 33.979)	59
period 7	1000	12.000	53.000	33.581	5.963	95.0% (33.211, 33.951)	50
period 8	1000	9.000	32.000	19.216	3.732	95.0% (18.984, 19.448)	28

Table 3. Results of stochastic model. Hourly overview of the service Levels

	num obs.	min	max	average	std. dev.	conf. int.	Arrivals
period 0	1000	30.000	40.000	39.027	1.285	95.0% (38.947, 39.107)	40
period 1	1000	29.000	38.000	35.611	1.607	95.0% (35.511, 35.711)	38
period 2	1000	32.000	46.000	43.597	2.205	95.0% (43.460, 43.734)	46
period 3	1000	35.000	46.000	45.509	1.011	95.0% (45.446, 45.572)	46
period 4	1000	32.000	42.000	41.393	1.164	95.0% (41.321, 41.465)	42
period 5	1000	38.000	49.000	48.275	1.440	95.0% (48.186, 48.364)	49
period 6	1000	9.000	54.000	37.755	7.218	95.0% (37.307, 38.203)	59
period 7	1000	12.000	43.000	29.299	4.753	95.0% (29.004, 29.594)	50
period 8	1000	7.000	23.000	15.390	2.439	95.0% (15.239, 15.541)	28

Table 4 Results of empirical model. Hourly overview of the service Levels.

Table 3 and 4 show some serious distance between the stochastic and the empirical model. The reason for this is expected to be the fixed arrivals in the empirical model. Unfortunate (and vice versa) series of calls arrive at the same time every simulation run, thus having a large impact on the service level per hour. The standard deviation in the empirical model is also a factor ~5 smaller than the stochastic model, except for period 7 which is the busiest period. This makes the hourly comparison less meaningful as the overall comparison.

Name	num. obs	min	max	average	standard dev	95.0% confidence
served contacts	1000	324.000	445.000	388.660	18.066	(387.539, 389.781)
Abandoned contacts	1000	0.000	31.000	8.736	4.284	(8.470, 9.002)
Service level	1000	-	-	0.855	0.039	(0.852, 0.857)
Occupancy ratio	1000	-	-	0.428	0.025	(0.427, 0.430)

Table 8. Results of stochastic model. Detailing the entire day.

Table 4 and 5 show the statistics for main output parameters related to handling calls for both the stochastic and the empirical model over an entire day. The tables show that the arrival process work as intended the average number (and confidence interval) of served contacts in the stochastic simulation matches that of the empirical simulation and the same goes for the abandoned calls. The

service level and occupancy ratio are somewhat higher in the stochastic model. The difference is less than 1% and is expected to be causal to the lack of variance in arrivals and the empirical distribution's lower bound of 15 seconds, used to fit the distribution to the data.

Name	num. obs	min	max	average	standard dev	95.0% confidence
served contacts	1000	371.000	396.000	388.194	3.243	(387.993, 388.395)
Abandoned contacts	1000	1.000	26.000	8.806	3.243	(8.605, 9.007)
Service level	1000	-	-	0.849	0.024	(0.847, 0.850)
Occupancy ratio	1000	-	-	0.433	0.025	(0.432, 0.434)

Table 9. Results of Empirical model. Detailing the entire day.

Email Verification

Since there is no dataset available to validate our email data to, something else has to be used. Email will be verified to its expected functionality. By manual input of arrivals and handling times scenarios can be created of which the expected output can be verified with the output of the simulation. An important factor determining the nature of these validation experiments is the aspect where email differ from calls in terms of functionality, such as interruptions of emails and the switching cost occurring. The validity of email will be tested with two experiments.

- Experiment 1: One agent is active and ten emails will arrive in one hour. The ten emails will be completed exactly after the sum of the email handling time + roundup (10 seconds) time for each email.
- Experiment 2: One agent is active and ten emails will arrive plus one phone call. The phone call will interrupt one email. It is expected that once the phone call arrives exactly $\frac{\text{call arrival time}}{\text{email handling time} + \text{roundup}}$ emails will be handled. Once all emails are handled the simulation clock will be at the sum of all the email handling time + EmailHandlingTime *0.2+ phone handling time.

First experiment

In the first experiment the interarrival time of emails is set to 6 minutes and a handling time of 6 minutes and 10 seconds roundup time. No calls arrive during the entire simulation. The expected result of the first experiment is that 9 emails arrive within the first hour. The time that the 9th email is handled will be exactly = 61.500 = 6 minutes for the first email to arrive + 9* 6.167(6 minutes and 10 seconds). This can be seen in *Figure 1* at line 20 under extTime.

	Event	Step	Type	Period	ArvTime	QueueTime	Outcome	Group	SrvTime	extTime	Group	Total	idle	busy	mail
1	Event	Step	Type	Period	ArvTime	QueueTime	Outcome	Group	SrvTime	extTime	Group	Total	idle	busy	mail
2	ArrivalA	0	1	1	6.000	0.000	-	0	0.000	0.000	0G	1T	1I	0B	1M
3	ArrivalB	0	1	1	12.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
4	DepartureA	0	1	1	6.000	0.000	Served	0	6.167	12.167	0G	1T	1I	0B	1M
5	ArrivalC	0	1	1	18.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
6	DepartureB	0	1	1	12.000	0.167	Served	0	6.167	18.333	0G	1T	1I	0B	1M
7	ArrivalD	0	1	1	24.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
8	DepartureC	0	1	1	18.000	0.333	Served	0	6.167	24.500	0G	1T	1I	0B	1M
9	ArrivalE	0	1	1	30.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
10	DepartureD	0	1	1	24.000	0.500	Served	0	6.167	30.667	0G	1T	1I	0B	1M
11	ArrivalF	0	1	1	36.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
12	DepartureE	0	1	1	30.000	0.667	Served	0	6.167	36.833	0G	1T	1I	0B	1M
13	ArrivalG	0	1	1	42.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
14	DepartureF	0	1	1	36.000	0.833	Served	0	6.167	43.000	0G	1T	1I	0B	1M
15	ArrivalH	0	1	1	48.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
16	DepartureG	0	1	1	42.000	1.000	Served	0	6.167	49.167	0G	1T	1I	0B	1M
17	ArrivalI	0	1	1	54.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
18	DepartureH	0	1	1	48.000	1.167	Served	0	6.167	55.333	0G	1T	1I	0B	1M
19	ArrivalJ	0	1	2	60.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M
20	DepartureI	0	1	1	54.000	1.333	Served	0	6.167	61.500	0G	1T	1I	0B	1M

Figure 17. Trace of email arrivals, regarding validation experiment 1.

Second experiment

The second experiment is almost identical to the first experiment except that one call arrives at the 40 minute mark. The call takes 5 minutes and 10 seconds to handle. The expected output is that at the 40 minute mark 5 emails have been handled (40 – 6) / (5* 6.167) = 5.5~. The expected time that the 9th call is handled is 61.5 + (0.2 * 6) + 5.167 = 68.867.

Figure 2 displays the trace result of the second experiment. Orange markings indicate the interrupted email and blue markings the call in the system (note that the column ArvTime marks the arrival time of the interrupted event. extTime marks the time of the onHold event). At line 12 a departure event is logged of contact E (of type 1 which is email). This is the 5th email handled which validates the first expectation. At line 14 the scheduled phone call arrives and interrupts the 6th email. Once the call is completed at exactly 40 + 5.167 the sixth call F is resumed. At line 25 under extTime the exit time of the 9th call is displayed as 67.867.

	Event	Step	Type	Period	ArvTime	QueueTime	Outcome	Group	SrvTime	extTime	Group	Total	idle	busy	mail
1	ArrivalA	0	1	1	6.000	0.000	-	0	0.000	0.000	0G	1T	1I	0B	1M |
2	ArrivalB	0	1	1	12.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
3	DepartureA	0	1	1	6.000	0.000	Served	0	6.167	12.167	0G	1T	1I	0B	1M |
4	ArrivalC	0	1	1	18.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
5	DepartureB	0	1	1	12.000	0.167	Served	0	6.167	18.333	0G	1T	1I	0B	1M |
6	ArrivalD	0	1	1	24.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
7	DepartureC	0	1	1	18.000	0.333	Served	0	6.167	24.500	0G	1T	1I	0B	1M |
8	ArrivalE	0	1	1	30.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
9	DepartureD	0	1	1	24.000	0.500	Served	0	6.167	30.667	0G	1T	1I	0B	1M |
10	ArrivalF	0	1	1	36.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
11	DepartureE	0	1	1	30.000	0.667	Served	0	6.167	36.833	0G	1T	1I	0B	1M |
12	OnHold-F	0	1	1	36.000	0.833	-	0	3.167	40.000	0G	1T	1I	0B	0M |
13	ArrivalG	0	0	1	40.000	0.000	-	0	0.000	0.000	0G	1T	0I	1B	0M |
14	ArrivalH	0	1	1	42.000	0.000	-	999	0.000	0.000	0G	1T	0I	1B	0M |
15	Departure	0	0	1	40.000	0.000	Served	0	5.167	45.167	0G	1T	0I	1B	0M |
16	Resume-F	0	1	1	36.000	0.833	-	0	0.000	45.167	0G	1T	1I	0B	0M |
17	ArrivalI	0	1	1	48.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
18	DepartureF	0	1	1	36.000	0.833	Served	0	4.200	49.367	0G	1T	1I	0B	1M |
19	ArrivalJ	0	1	1	54.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
20	DepartureH	0	1	1	42.000	7.367	Served	0	6.167	55.533	0G	1T	1I	0B	1M |
21	ArrivalK	0	1	2	60.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
22	DepartureI	0	1	1	48.000	7.533	Served	0	6.167	61.700	0G	1T	1I	0B	1M |
23	ArrivalL	0	1	2	66.000	0.000	-	999	0.000	0.000	0G	1T	1I	0B	1M |
24	DepartureJ	0	1	1	54.000	7.700	Served	0	6.167	67.867	0G	1T	1I	0B	1M |

Figure 18. Trace of email arrivals, regarding validation experiment 2.

Call Center Simulation
-Experimentation

This part of the document details the experiments, conclusion, discussion and future studies. Starting with the experiment setup detailing why and how the experiments will be performed, this is followed by the execution of the experiments. The results of the experiments will be reviewed in the conclusion. The discussion touches up on the subjects that didn't work out as planned. Finally, the future studies section details the potential of continuing this study into various directions.

Experiment protocol

For every experiment a 2^k factorial experiment design will be used, where the factors (*k)* allow for two scenarios, for more information see [9]. This allows us to perform an A-B test for two different methods over a range of varying input variables. With the use of statistical software package Design Expert 11 (www.statease.com) the setup of the factors will be calculated. The input factors are based on the two point input used in factorial design of experiments. Design Expert will be able to identify significant factor interactions and generate response surfaces (A 3D representation of the relationship between input variables and the corresponding response variables). It is expected that this will deliver sufficient data to give a qualitative answer about the performance of the setups compared in each experiment. For every experiment common random numbers (CRN) will be used for each factor set (multiple runs).

For the experiment we vary the following factors.

First we vary between a big and a small size call center. The reason for this is that having more agents allows for more flexibility in how to assign agents. This is a luxury that small service helpdesks do not have. For example we expect that scheduling agents to solely handle email will result in a greater throughput of email since there are no interruptions. Big call centers could miss a few agents during quiet periods for small call centers this might not be the case.

Six different policies will be evaluated [1]:

- 1 agent group handling both calls and email with Pre-emption (1ServP).
- 1 agent group handling both calls and email without Pre-emption (1Serv!P).
- 2 agent groups, 1 for calls and 1 for calls and emails. Where email is handled with Pre-emption (2ServP).
- 2 agent groups, 1 for calls and 1 for calls and emails. Where email is handled without Pre-emption (2Serv!P).
- 2 agent groups, 1 for strictly calls and 1 for strictly emails, here the proportion of agents handling email is lower than in the next policy (2ServSep). When handling calls and email separately there is no notion of pre-emption since calls and emails are not blended.
- 2 agent groups, strictly calls and 1 for strictly emails, here the proportion of agents handling email is higher than in the previous policy. (2ServSep (Large))

Next we want to vary the arrival rate of calls and email separately.
Here we differ between three stages small, normal, high.

Small/large(2)* policy(6)*arrival rate call(3)*arrival rate email (3) =108.

[1] *Policy notation:*

- *The digit stands for the number of agent groups followed by the word serv (short for servers).*
  - *When there is one agent group, the agent groups handles both calls and emails.*
  - *When there are two agent groups, 1 of the two strictly handles calls, the other depends on the policy postfix (next bullet point).*
- *The postfix {P,!P, Sep} stands for the method of handling, pre-emption/ non pre-emption and separate handling. Pre-emption and non-pre-emption imply that there is a group that handles both calls and email (since emails can be pre-emptively stopped to serve a call). Postfix Sep stands for separate handling, meaning the second agent group only handles email.*[1]

Since the experiment involves both small and large call centers a fitting workload for these models need to be determined. First the proportions of the call center need to be set. This will allow the model to be scaled from small to large by simply multiplying the number of agents and the work load. The originally modelled call center is small in size. The call queue is manned by 5 people at the peak hours of the day. This is without man power to facilitate email.

Initially the number of agents were doubled such that there is an equal amount of agents for an equal work load. The original email arrival rates (obtained from the CC4Skype's helpdesk) are too low for this simulation. The email arrival rate is based on a helpdesk operated by two agents, and can still be considered low. To overcome this the expected workload (total handling time) of the calls will be matched by emulating an equal mean workload for emails. There is no other reason for matching the workload except that an equal workload seems only fair in light of this research. The amount of emails needed to match the workload of calls will be distributed proportionally over the original email arrival rates.

The mean handling time of the Log-normal generated calls is 157.3 seconds. For emails generated with the Gamma distributions this is 484.6 seconds. 399 calls arrive over the course of one day resulting in a workload of 62762.7 seconds. To match this 129.5 emails need to arrive over the course of one day. The results can be seen in *Table 1*.

Arrivals Per hour	Overnight	8	9	10	11	12	13	14	15	16
original	1.472	0.638	1.770	1.968	2.002	1.657	1.713	1.886	1.922	1.566
scaled	11.486	4.9775	13.810	15.362	15.624	12.928	13.366	14.720	15.001	12.225

Table 10. Updated email arrivals after scaling. The top row shows the original email arrivals, the bottom row the scaled email arrivals used in the simulation..

Scheduling 5 agents for email and 5 for calls, resulted in too many agents for the amount of work (overstaffed). Especially where agents handle both calls and emails service level parameters easily reached 100%~. Therefore the amount of extra agents was multiplied by 0.75 and rounded down. Still this resulted in occasions were quality of output parameters could not be measured.

To overcome this issue without too much tinkering factors used to vary the call and email intensity will be 120%, 130% and 140%, instead of 90%,100%,110%. Increments of 10% were chosen as the email arrivals have a strong effect on the service level.

Disclaimer: The proportion of agent groups and schedules are based on the initially modelled call center. Change in agent schedules to serve a policy better is kept at a minimum to avoid unfair disadvantage of accidentally exploiting the problem environment.

Experiment1:-pre-emptive vs non pre-emptive

Goal

Experiment 1 serves as a look into the difference between offline and online scheduling. Both setups are simple approaches to handling both calls and emails. The only logic here is that Calls > emails. This experiment will set a base line for the online and offline scheduling.

Setup

In this experiment we compare a call center with two queues one for email and one for calls. Two setups will be compared to each other:

There is one agent group serving both calls and emails. Pre-emption is enabled. The other setup will be the same only pre-emption is disabled Emails have an increased handling time when resumed. In both setups calls have priority over email at all times. First all waiting calls are handled and then waiting emails.

Output Parameters

The experiment will run for one week. Important output parameters here are service: level for both email and calls; the occupancy ratio as a strong indicator of the amount of stress; the amount of extra time due to resuming email; the average waiting time for calls and emails.

Hypothesis

Setup 1, (1ServP). Is expected have a better service level for calls than in setup 2, since calls aren't hindered by emails. Occupancy will be high due to resuming emails.

Setup 2 (1Serv!P).It is expected that the service level of calls will suffer. The number of emails handled and email service level will be higher than in setup 1.

Experiment 2: use periods where emails are handled vs online scheduling.

Goal

The goal of this experiment is to see if circumstances can be improved by adding logic to experiment 1 and which type performs better. Call centers typically employ routing policies to improve performance measures. In this experiment agents will be assigned to agent groups that handle different tasks. Here 3 setups are compared. Two setups (2ServP & 2Serv!P) use a routing policy where a portion of the agents handle calls and email according to a policy. The other setup (2ServSep) doesn't blend calls meaning that calls and emails are handled by separate groups.

Setup

This experiment is similar to the first experiment but email will be handled in a more intelligent manner. The majority of the agents will handle calls only. A portion of the agents will be scheduled to handle email and calls. These agent have the a call > email priority but have the lowest priority of serving a call compared to other agents. This setup will be ran twice one time with and one time without pre-emption.

The call centers where calls are blended will be compared to another call center will be simulated where emails can't be interrupted. The majority of the agents focus on call handling. This setup will be performed twice once where the portion of agents scheduled to handle email is equal to online-scheduling experiment and one where the portion of agents is half of that. This is due to the number of different jobs the agents perform.

The experiment will run for one week.

Hypothesis
Adding a second group to the blended call centers will have a different result depending on whether pre-emption is used or not. With pre-emption it is expected that interruptions are reduced in the second group. As the first group that handles only calls, functions as a buffer.

The setup without pre-emption, having a group that strictly handles calls, will give more priority to calls. It is expected that this will give a more balanced result for both calls and emails.

The call center where calls and emails are handled separately is expected to lack flexibility and either scores high on one of the two service level, calls or emails.

Output Parameters
Important output parameters here are:

- service level for both email and calls, here the desired level is at least 0.8.
    - For calls this is: number of calls handled within 20 seconds / number of arrived calls.
    - For email this is: Number of emails handled within 4 hours / number of handled emails + number of emails that are waiting longer than 4 hours.
- Occupancy ratio.
    - This is the time agents are occupied divided over their total time present in the call center.
- Number of call interruptions.
- The average waiting time for calls and emails.
- Number of abandoned calls.

Experimentation

Experiment 1: Pre-emption vs Non-preemptive

When we compare Pre-emption vs non pre-emption besides the main performance measures, secondary output parameters such as occupancy are important. An occupancy ratio of 100% is likely not feasible for real call center agents. First the service levels for emails and calls will be compared. Afterwards the trade-offs of choosing a policy based on secondary output parameters as occupancy, abandonments, number of email interruptions and speed of answer. Finally a comparison will be written about the pro's, cons and possible improvements about both methods.

1ServP

1ServP is the policy where both calls and emails are handled by 1 agent group. Calls have priortity over emails and in case of 1ServP can even interupt emails such that calls can be handled as quickly as possible.

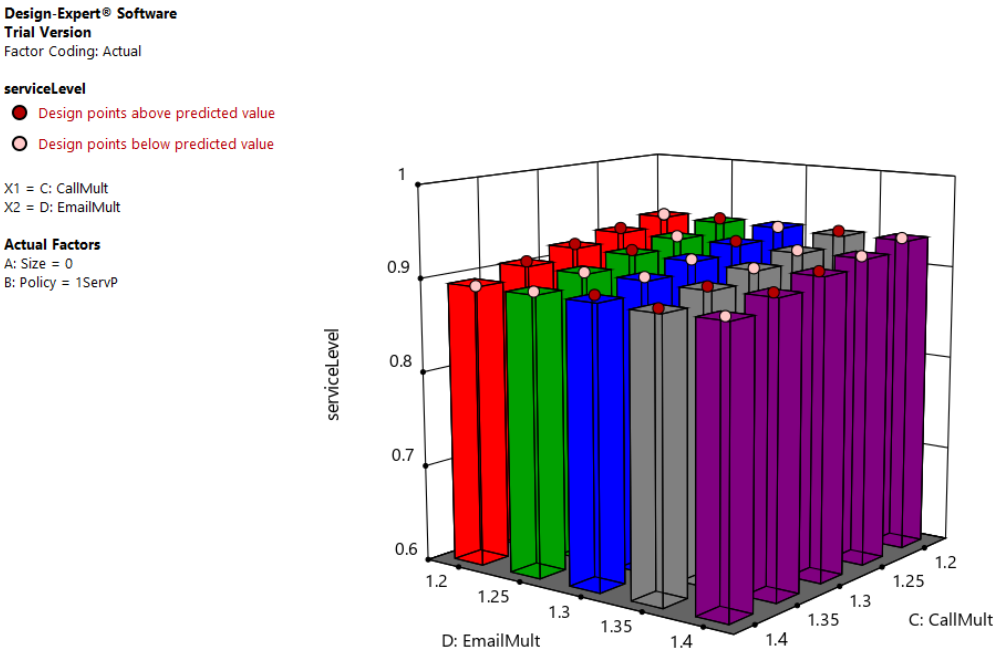

Figure 19. Service level, 1 group of servers with pre-emption (small call center) .

Figure 1 and 2 display the performance of the service level for both emails and calls with pre-emption(1ServP). As expected service level of calls doesn't suffer in this model. It is however included to compare it to a non-pre-emptive environment. It seems that due to merging two groups there are plenty of resources available to accommodate most incoming calls (90% and up). *Figure 2* shows that the service level of emails is decent. Either the service level of email is low due to there being too many calls or too many emails. This results in the requirement of a right amount of calls and emails.

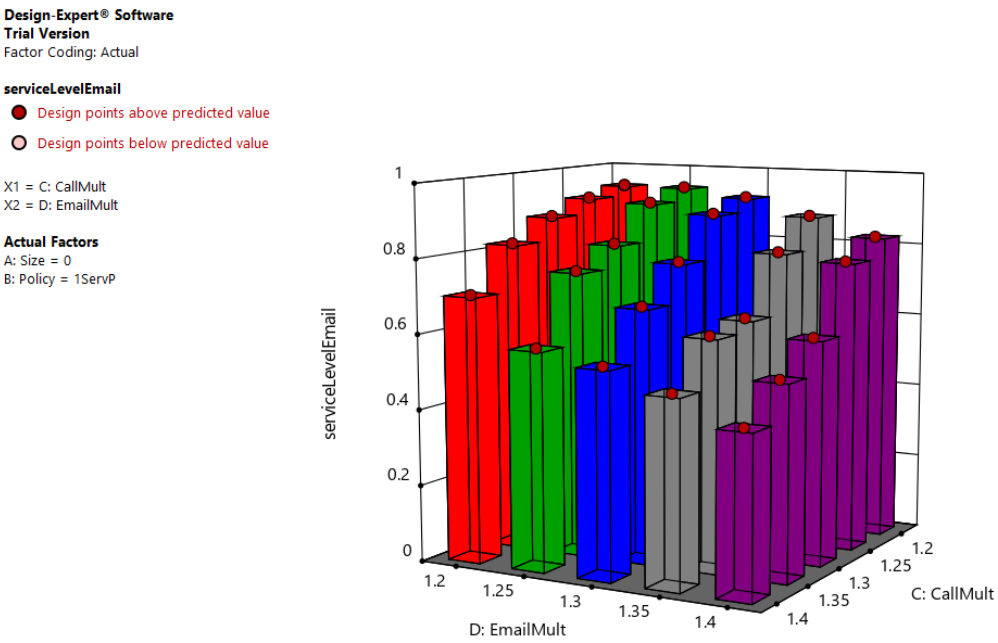

Figure 20. Service level email for 1 group of servers with pre-emption (small call center).

policy	callMult	emailMult	SL C	SL E	SoA	SoA E	OCC1	abandoned	Interrupted
1ServP	1.2	1.2	0.936	0.943	0.081	59.788	0.841	11.45	1420.15
1ServP	1.2	1.3	0.934	0.922	0.084	78.071	0.883	11.73	1621.51
1ServP	1.2	1.4	0.935	0.828	0.082	113.168	0.920	11.35	1808.68
1ServP	1.3	1.2	0.919	0.882	0.107	93.146	0.891	16.83	1749.89
1ServP	1.3	1.3	0.916	0.775	0.111	133.048	0.926	17.14	1940.75
1ServP	1.3	1.4	0.919	0.597	0.106	188.429	0.947	16.92	2058.77
1ServP	1.4	1.2	0.893	0.705	0.149	150.231	0.931	24.74	2053.6
1ServP	1.4	1.3	0.900	0.547	0.137	200.077	0.946	23.49	2147.36
1ServP	1.4	1.4	0.896	0.427	0.146	253.078	0.954	24.9	2191.71

Table 11. Displays a truncated report of the output parameters for 1 group of servers with pre-emption., The headers from left to right are: the policy used; the call arrival intensity multiplier; the email arrival intensity multiplier; the service level for calls; the service level for emails; the speed of answer for calls & for email (in minutes); the occupancy ratio for the designated group; the amount of abandoned calls; the total amount of email interruptions. This notation is used throughout the rest of the experiments.

Table 2 displays the input and output parameters for 1ServP. It shows that the occupancy is very high as well as the amount of time email is interrupted over one week. Occupancy ratios of near 100% might be undesirable in real call centers as this puts a lot of stress on employees. This however could be resolved by shorter agent shifts. The row where the call multiplier is 1.4 and email multiplier is 1.3 shows an email service level of 54%~ and an average speed of answer for email of 200 minutes. The combination of these output parameters hints strongly at an increasing email backlog as more emails arrive than are completed.

1Serv!P

Alternatively *Figure 3 and 4* show reversed for handling both emails and calls without pre-emption (1Serv!P). Near perfect service level for emails and still reasonable, (desirable even) service levels for calls. The near perfect score for emails might be desirable as this eliminates the buildup of an email backlog. Otherwise this would result in an ever increasing backlog that would require periodic attention in order to avoid perpetual drop in email service level.

1Serv!P seems more robust than when pre-emption is allowed. The service level of calls drops only a little below 80%, which is seen as the target service level. Making this policy preferable when variance in arrivals is expected. This policy might benefit from having a small portion of agents that handling only calls during peak hours or during opening hours. This would ensure that not all agents are busy with the remaining emails from the day before as the day starts. Limiting the amount of concurrent agents working on emails might also help as a safety net.

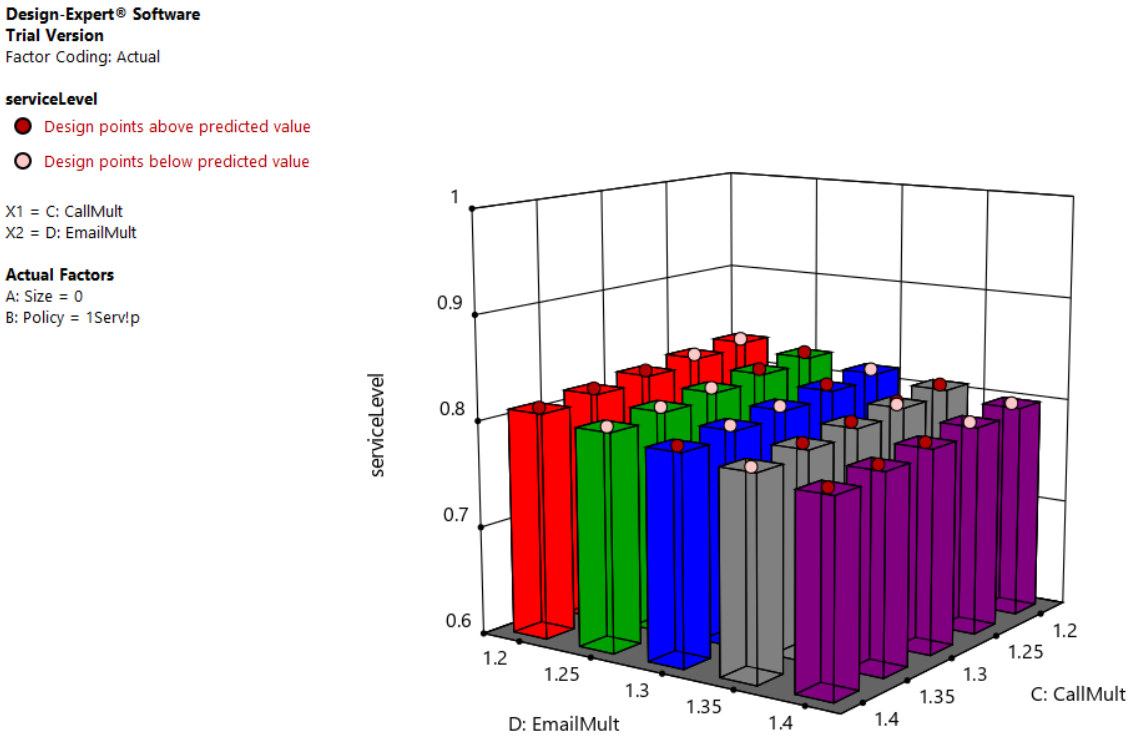

Figure 21. Service level for 1 group of servers without pre-emption (small call center).

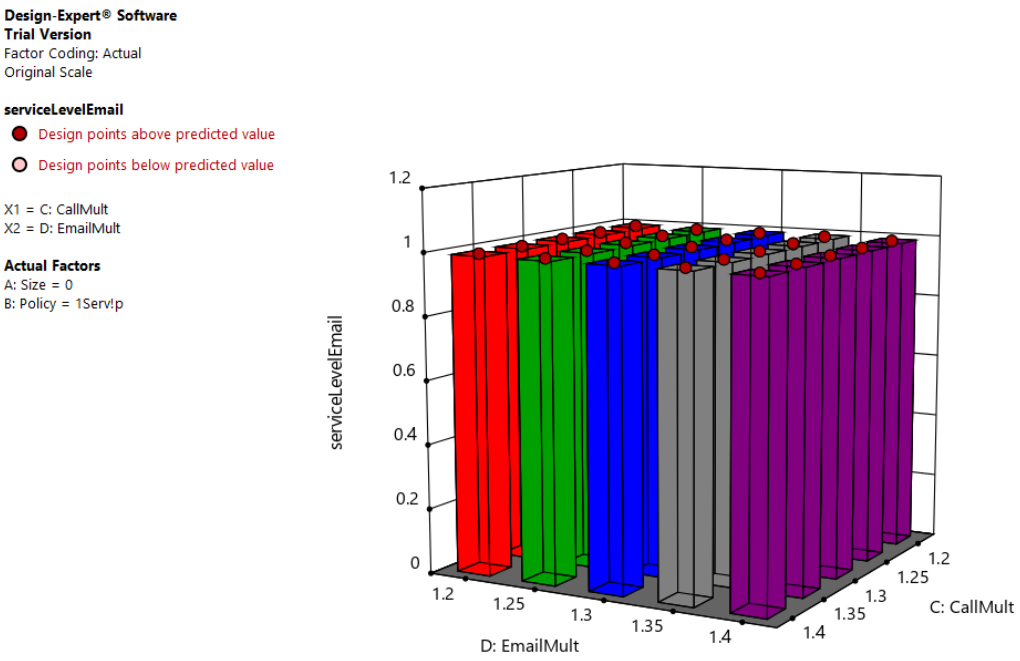

Figure 22. Service level email for 1 group of servers without pre-emption (small call center).

Table 3 shows the results of 1 agent group without pre-emption (1Serv!P). The absence of interruptions is clearly visible in the occupancy which averages around 60-70 %~. The 99% email service level and Speed of answer for email indicate that the rate at which email is handled can be relaxed in some cases to boost the call service level. This would also help the high number of abandonments compared to 1ServP.

policy	callMult	emailMUlt	SL C	SL E	SoA	SoA E	Occ1	abandon.	interrupt.
1Serv!p	1.2	1.2	0.833	0.996	0.186	10.204	0.661	40.81	0
1Serv!p	1.2	1.3	0.818	0.996	0.204	10.629	0.682	46.13	0
1Serv!p	1.2	1.4	0.802	0.995	0.224	11.167	0.702	47.67	0
1Serv!p	1.3	1.2	0.825	0.997	0.198	10.403	0.678	45.81	0
1Serv!p	1.3	1.3	0.808	0.996	0.213	11.050	0.701	50.11	0
1Serv!p	1.3	1.4	0.793	0.994	0.235	11.473	0.718	52.94	0
1Serv!p	1.4	1.2	0.814	0.996	0.206	10.558	0.698	51.11	0
1Serv!p	1.4	1.3	0.798	0.996	0.226	11.344	0.717	55.31	0
1Serv!p	1.4	1.4	0.783	0.993	0.243	11.744	0.737	60.87	0

Table 12. Shows the resulting output parameters over all arrival multipliers for 1Serv!P.

Comparison

To give a qualitative answer about which policy is better depends on the objective and priorities of each call center. All of the outcomes have been subjected to an one sample t-test (in R) with 95% confidence Each of the output parameters are significantly different from each other (*Appendix Tables 1 & 2).* Email service level seems like a treacherous parameter as emails pile up before this is notably visible in the parameter. This is especially notable in a system with pre-emption as the service level decreases faster compared to the rate at which emails increases, as can be seen from the speed of answer.

Both models perform "well", except that one favors calls over emails, trading abandonments over high occupancy of workers. Interesting is that 1ServP would be able to raise the occupancy ratio to near 100%. One argument for 1Serv!P is that it is the more robust policy of the two. The service level for both emails remain stable over the increasing amounts of arrivals.

Both models could possibly be improved by switching between policies when emails pile up. More specifically use 1ServP during peak hours and 1Serv!P during quiet hours. The maximum number of times an email can be interrupted could possibly add an interesting twist to 1ServP. The extreme opposite results of both methods leads to believe that a combination of the two would deliver ideal results.

Experiment 2: Multi server models.

2ServP & 2Serv!P the idea behind these models is that there are two groups of servers. The first group that handles strictly calls acts as a buffer for the second group of servers that handles calls and emails. This buffer is expected to work both ways. When the amount of calls becomes too much for group 1 group 2 can assist. Alternatively the agents of group 2 only handle calls when all agents of group 1 are busy, as a result of this agents can focus on email. However these policies require that email can be handled by group 2 within the time frame of four hours and the amount incoming calls is not too disruptive on group 2.

2ServP

2ServP is a variation of 1ServP. In this policy there are two groups of servers where one groups strictly handles calls and the other group handles calls and emails, where emails are interrupted by calls when all agents in the first group are occupied.

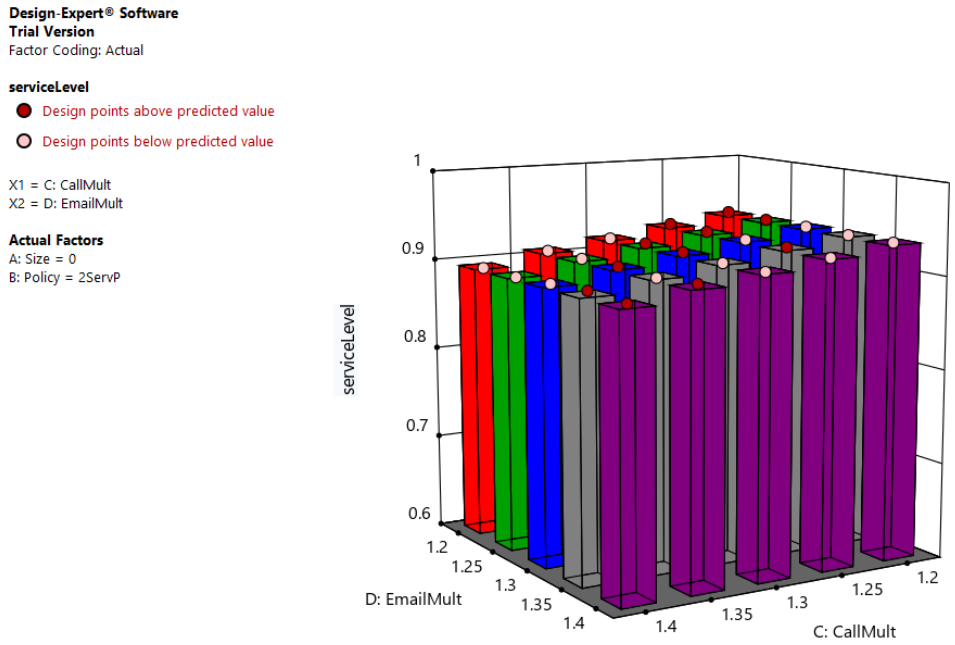

Figure 23. Service level for 2 groups of servers. Where one group handles calls and emails with pre-emption (small call center).

Two servers with pre-emption (2ServP) obtains similar results for the call service level as 1ServP. From Figure 5 alone it is immediately visible that there are enough resources for handling calls available (as expected). Emails on the other hand (Figure 6) shows less promising results as it seems

that there aren't enough agents available to deal with the amount of emails. The service level is worse than its 1 group variant. Most likely the agents can't keep up with the amount of incoming emails and get swamped due to pre-emption for calls. It is clear that 2ServP offers the second group no protection from the constant flow of incoming calls. Simply adding a surplus of agents to handle email doesn't seem to work for 2ServP as it result in a loss of flexibility. A more sophisticated agent schedule should be created in order to specifically tailor the amount of agents to the call forecast.

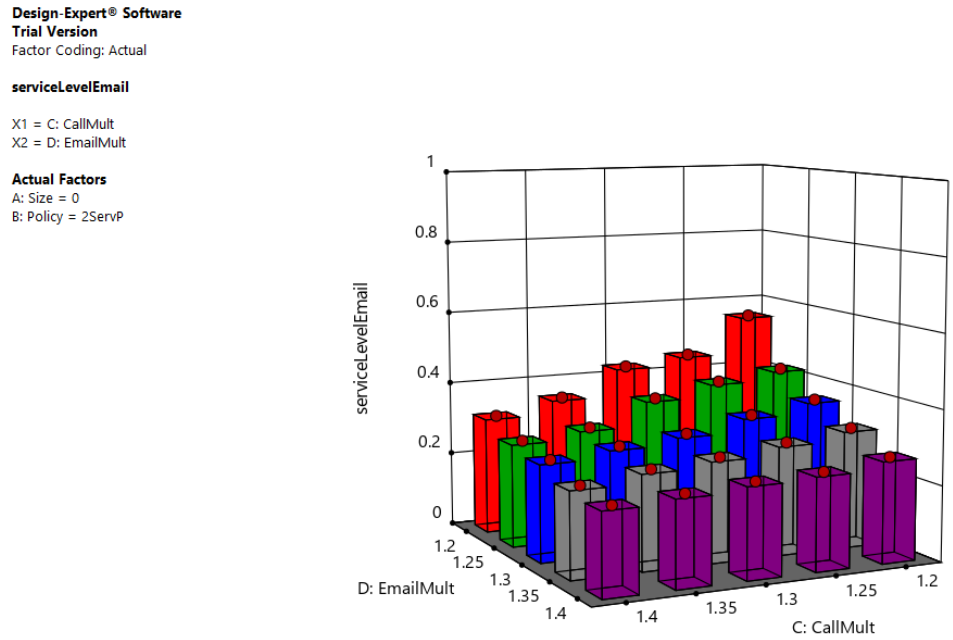

Figure 24. Service level email for 2 groups of servers. Where one group handles calls and emails with pre-emption (small call center).

Table 4 displays the results of 2ServP where the initial amount of agents serve only calls, and the adittional agents only emails. The huge gap between service level for calls and emails hints at possible performance improvement. This should be achieved by assigning less agents to the strictly call group (and more to the email group). This improved schedule should exploit the arrival rates in order to have enough agents to cater the incoming calls. This schedule should keep the number of interupted email at a minimum.

Policy	CallMult	EmailMult	SL call	SL mail	SoA C	SoA E	Occ 1	Occ 2	abandon.	interrupt.
2ServP	1.2	1.2	0.938	0.560	0.962	0.078	196.746	0.509	0.958	11.29
2ServP	1.2	1.3	0.937	0.366	0.834	0.081	275.669	0.510	0.968	11.42
2ServP	1.2	1.4	0.936	0.273	0.706	0.081	333.034	0.511	0.975	11.27
2ServP	1.3	1.2	0.917	0.433	0.922	0.111	235.885	0.536	0.964	17.88
2ServP	1.3	1.3	0.919	0.301	0.751	0.107	311.032	0.539	0.971	16.57
2ServP	1.3	1.4	0.917	0.244	0.627	0.109	370.787	0.539	0.976	17.03
2ServP	1.4	1.2	0.894	0.319	0.774	0.145	303.215	0.569	0.968	25.1
2ServP	1.4	1.3	0.897	0.263	0.664	0.141	352.717	0.567	0.974	23.58
2ServP	1.4	1.4	0.898	0.221	0.569	0.142	406.661	0.567	0.979	25.27

Table 13. Shows the resulting output parameters over all arrival multipliers for 2ServP.

2Serv!P

2Serv!P a variation on 1Serv!P where there are two agent groups. One group strictly handles calls and the other group handles calls as well as emails. *Figure 7 and 8* show the service levels for 2Serv!P. The Service level for calls seem to decrease as it gets busier, but overall remain quite stable.

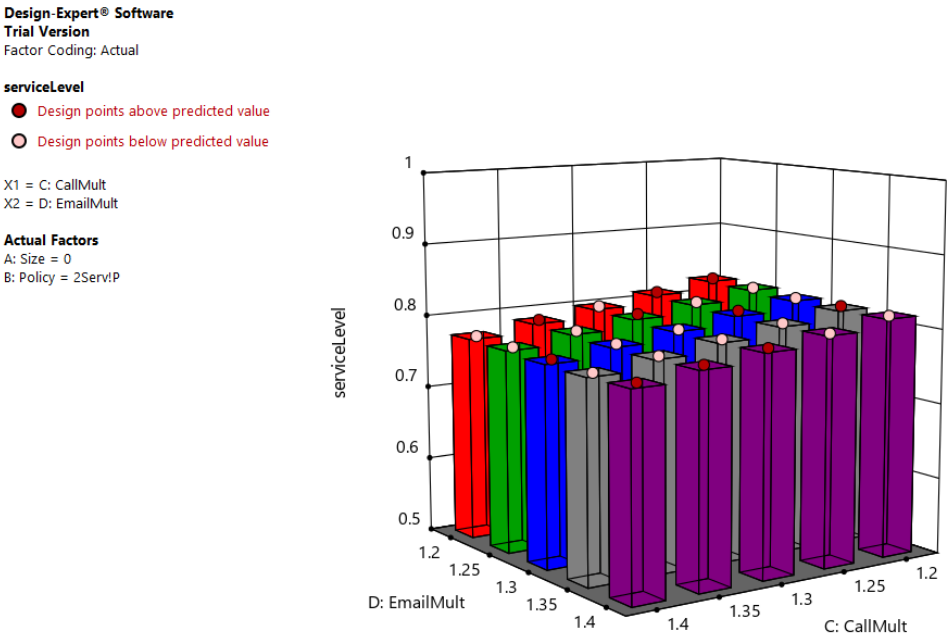

Figure 25. Service level for 2 groups of servers. Where one group handles calls and emails without pre-emption (small call center).

There seems to be no improvement over its simple variant. The speed of answer for email is notably higher for all configurations *(Table 5)*. Emails are suffering under this policy this is especially notable where the call and email multipliers are high.

The email service level drops to 0.658 compared to 0.995 in 1Serv!P. While the service level for calls roughly stays the same. Just like 2Serv!P a better solution might be found in an optimized agent schedule.

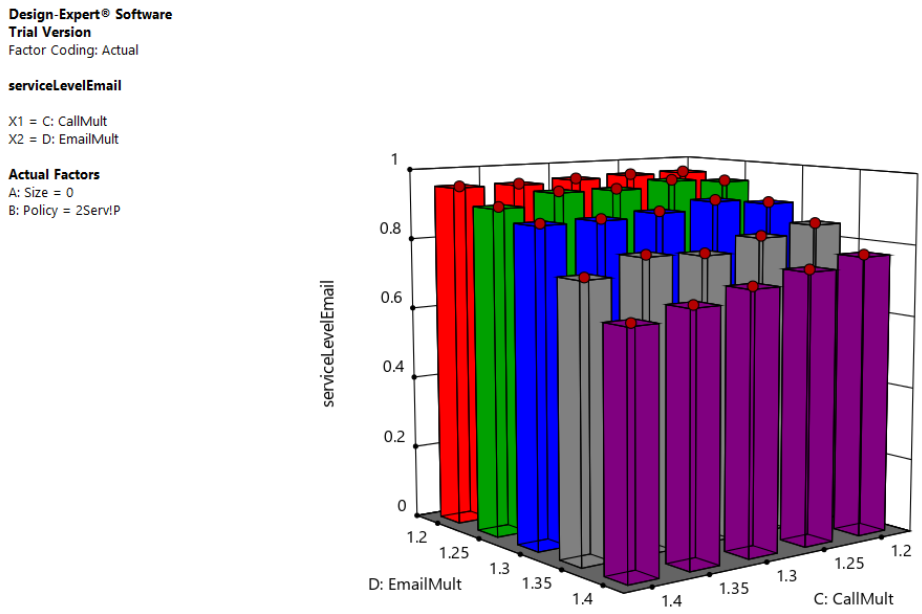

Figure 26. Service level email for 2 groups of servers. Where one group handles calls and emails without pre-emption (small call center).

Policy	CallMult	EmailMult	SL call	SL mail	SoA C	SoA E	Occ 1	Occ 2	abandon.	interrupt.
2Serv!P	1.2	1.2	0.827	0.964	0.222	52.002	0.542	0.911	35.39	0
2Serv!P	1.2	1.3	0.820	0.896	0.231	91.552	0.542	0.945	36.58	0
2Serv!P	1.2	1.4	0.820	0.777	0.230	132.299	0.542	0.963	36.18	0
2Serv!P	1.3	1.2	0.802	0.959	0.260	56.730	0.574	0.918	46.6	0
2Serv!P	1.3	1.3	0.794	0.890	0.269	96.430	0.576	0.954	48.64	0
2Serv!P	1.3	1.4	0.796	0.720	0.266	151.625	0.577	0.965	48.02	0
2Serv!P	1.4	1.2	0.776	0.954	0.300	63.827	0.604	0.923	58.95	0
2Serv!P	1.4	1.3	0.772	0.878	0.300	103.526	0.606	0.953	56.73	0
2Serv!P	1.4	1.4	0.772	0.658	0.302	163.227	0.605	0.967	58.19	0

Table 14. Shows the resulting output parameters over all arrival multipliers for 2Serv!P.

2ServSep

The last policy is the variant where there are two agent groups one for calls and one for emails. Two agent schedules where used for this policy. One is the same schedule used for the 2ServP policies. The other schedule has the agent group for emails decreased by one over the entire day. The results for both these agent groups with this policy show that simply putting together a roster(based on the original model) is too shortsighted.

Policy	CallMult	EmailMult	SL Call	SL Email	SoA C	SoA E	Occ1	Occ2	abandon.	interupt.
2ServSep	1.2	1.2	0.700	0.975	0.548	37.268	0.571	0.865	118.11	0
2ServSep	1.2	1.3	0.695	0.958	0.565	59.761	0.574	0.923	121.69	0
2ServSep	1.2	1.4	0.697	0.891	0.551	96.292	0.572	0.949	116.85	0
2ServSep	1.3	1.2	0.652	0.973	0.656	35.753	0.611	0.862	158.19	0
2ServSep	1.3	1.3	0.654	0.962	0.656	54.371	0.610	0.917	156.64	0
2ServSep	1.3	1.4	0.654	0.897	0.651	100.125	0.609	0.953	160.2	0
2ServSep	1.4	1.2	0.608	0.973	0.766	38.647	0.645	0.869	209.7	0
2ServSep	1.4	1.3	0.610	0.954	0.760	60.903	0.644	0.915	206.76	0
2ServSep	1.4	1.4	0.607	0.885	0.764	100.585	0.646	0.953	208.39	0
2ServSepL	1.2	1.2	0.866	0.209	0.197	406.364	0.473	0.974	29.59	0
2ServSepL	1.2	1.3	0.864	0.169	0.201	467.873	0.475	0.977	29.86	0
2ServSepL	1.2	1.4	0.864	0.140	0.198	510.074	0.476	0.979	30.72	0
2ServSepL	1.3	1.2	0.829	0.214	0.263	405.414	0.512	0.974	46.15	0
2ServSepL	1.3	1.3	0.830	0.162	0.264	469.085	0.513	0.977	45.78	0
2ServSepL	1.3	1.4	0.833	0.134	0.257	521.395	0.512	0.980	44.44	0
2ServSepL	1.4	1.2	0.796	0.214	0.331	405.105	0.547	0.975	63.15	0
2ServSepL	1.4	1.3	0.796	0.171	0.332	465.039	0.548	0.978	64.59	0
2ServSepL	1.4	1.4	0.796	0.138	0.330	512.870	0.550	0.979	63.37	0

Table 15. The combined results of 2 variants of 2 group of servers 1 for calls and one for emails.

2ServSepL is the group with a decreased amount of agents on email. This group performs average on calls and good on emails. Alternatively 2ServSep scores relative average on calls and bad on emails. It is expected that 2ServSep would score not as good as the blend policies. Although it seems similar to the 2ServP and 2Serv!P the schedule doesn't do this policy any justice.

There would be no fair comparison as long as none of these policies are performing optimally. As none of the two group variants improve on the 1 group variants.

Evaluation

First and foremost the second experiment did not give the expected result, hence large call center data has been omitted. As it turned out the agent rosters are not translatable to other policies. The results for 2ServSep (the two variants) make painstakingly clear that simply adding or subtracting an agent from one group to the other does not result in an acceptable output for both calls and emails. What was initially meant by this experiment was to give every policy the same (un)favorable conditions. This is clearly not possible as each policy benefits from a different sort of schedule with respect to the arrival rate. By simply multiplying the amount of agents for large call centers and assigning a seemingly arbitrary schedule, the effects of experiment 2 have only been enhanced. The results with confidence intervals are omitted from the appendix since no fair comparison can be made without proper schedules.

Important Notice

Due to the results of experiment 2 the decision has been made to discontinue the experiment as originally intended. This means that large call centers setups will not be evaluated. The reason for this is that generating an optimal schedule will take too much time, due to the increased number of agents. Instead the following chapters about the remaining experiments (2.2 & 3) will focus on schedule generation for the "small" call center format and comparing the results of each different policy.

This also means there will be no regression analysis. The factors chosen as input for the regression analysis (call center size, email arrival intensity, call arrival intensity, policy) were designed to see how different policies behave depending on the size of the call center. Without size as a factor, the regression model has lost its power.

In the following experiment 2.2 a Hill-climber algorithm is employed to generate a schedule. The reason for this is that a decision had to be made on how to continue the experiments in the final days of the project.

The choice was between:

1. Generating schedules for each policy with the Hill-climber that can't be proven to be optimal and evaluate the results of each policy.
2. Simulating all possible schedules for each policy (very time consuming).
    - Generating schedules with the initial amount of periods would be too time consuming (weeks). In order to generate schedules within an acceptable time frame the number of periods would have to be reduced, by increasing the length of each period from 60 to 90 minutes. Increasing the length of the periods is risky as it decreases the flexibility and this might lead to a decreased performance and losing valuable time.

While initially pursuing the first option to use local search (Experiment 2.2), a local search generated schedule showed that 2ServP is able to surpass 1ServP in performance. Proof that the two group policies could improve on one group policies was decisive to continue with the second option, simulating all possible schedules for each policy (Experiment 3).

Experiment 2.2

Experiment 2 gave insight to the conclusion that the problem at hand is far more complex than initially thought. Simply adding additional agents to accommodate emails according to a policy doesn't do any justice. In an attempt to show that a policy where two agent groups are used can improve upon its simpler variant a Hill-climber algorithm was used to generate schedules. The schedule is represented as an array, where each index represents a time slot and the value, the amount of agents assigned to a group.

Example schedule:

[0, 7, 7, 8, 8, 8, 8, 5, 5, 3, 3] – Total available agents, every index is a period.

[0, 1, 1, 5, 4, 1, 3, 1, 2, 2, 2] – Group 1

[0, 6, 6, 3, 4, 7, 5, 4, 3, 1, 1] – Group 2

The Hill-climber algorithm allows us to (not fully) explore the solutions space. Since there is no way of proving the optimal solution will be found policies can't be compared to each other, making it impossible to find a "winner".

Two hill-climbers were made to generate solution schedules. Both hill climbers iteratively look for neighbor schedules by increasing/decreasing the amount of agents in the group that strictly handles

calls, this process decreases/increases the number of agents in the group that handles both calls and emails.

The first Hill-climber is called random restart hill climber. It starts with a random generated solution. This schedule is simulated ten times to assign a score (composite of the service level of calls and emails). Then all its neighbors are simulated, the neighbor with the biggest score increase is chosen as the next candidate. Next all the new candidates solutions are evaluated, this process repeats until no better solutions are found. The algorithm creates a new random initial solution and starts over. All solutions are written to a CSV for manual evaluation. This is because the success of a solution depends on the two service levels instead of the combined score. The random restarts allow a broader (greedy) exploration of the solution space.

Policy	CMult.	EMult	Call Sl	Email SL	SoA Call	SoA Email	Occ1	Occ2	Abandon.	Interupt.
2ServP	1.2	1.2	0.940 (0.937,0.942)	0.953 (0.950, 0.956)	0.075 (0.072, 0.079)	58.465	0.424	0.862	11.05	729.7 (715.496, 743.904)
2ServP	1.2	1.3	0.936 (0.934,0.939)	0.928 (0.916, 0.941)	0.080 (0.076,0.084)	79.657	0.422	0.900	11.53	793.35 (779.144, 807.556)
2ServP	1.2	1.4	0.936 (0.934,0.939)	0.845 (0.816, 0.873)	0.080 (0.076, 0.085)	114.150	0.417	0.934	11.04	852.69 (841.524, 863.856)
2ServP	1.3	1.2	0.918 (0.915,0.920)	0.933 (0.925, 0.942)	0.109 (0.104,0.115)	77.711	0.434	0.897	16.99	847.22 (831.713, 862.727)
2ServP	1.3	1.3	0.919 (0.917,0.922)	0.864 (0.840, 0.888)	0.106 (0.101,0.111)	111.003	0.422	0.932	16.72	925.8 (912.498, 939.102)
2ServP	1.3	1.4	0.917 (0.914,0.920)	0.713 (0.674, 0.752)	0.112 (0.106, 0.117)	153.103	0.434	0.950	16.55	951.86 (942.059, 961.661)
2ServP	1.4	1.2	0.899 (0.896,0.902)	0.860 (0.834, 0.885)	0.139 (0.133,0.146)	111.814	0.441	0.929	23.77	977.16 (961.657, 992.663)
2ServP	1.4	1.3	0.896 (0.893,0.899)	0.695 (0.655, 0.735)	0.146 (0.139,0.152)	160.348	0.441	0.946	24.27	1019.91 (1008.604,1031.216)
2ServP	1.4	1.4	0.898 (0.895,0.901)	0.498 (0.461, 0.535)	0.143 (0.137,0.149)	216.024	0.446	0.958	24.21	1046.38 (1036.316,1056.444)

Table 16. The results of a 2ServP schedule found by the Random Restart hill climbing algorithm. Consisting of the following agent roster [0, 1, 1, 5, 4, 1, 3, 1, 2, 2, 2], [0, 6, 6, 3, 4, 7, 5, 4, 3, 1, 1]. The first vector represents the agent group handling only calls. The second the agent group that handles calls and emails. The position of each number represents the hour of the work day. The confidence interval for Speed of answer and Occupancy have been omitted due to spacing issues.

The second algorithm used was Simulated Annealing which is as a Hill-climber, similar to first approach but supposedly less greedy. The Hill-climber works with a variable called temperature, the temperature and the difference in score determines the possibility of accepting a candidate that is worse. As the temperature decreases every iteration so does the probability of selecting bad candidates. The idea behind this is that Initially a large amount of local optima are explored (á la random walks) until the algorithm settles on greedily climbing one optima. Simulated Annealing starts with a random initial solution.

Table 7 Shows the result of a roster found by the Hill-climbing algorithm. It shows crucial evidence that 2ServP is able to improve upon 1ServP (*Table 2*). With a call and email multiplier of 1.3 for

both, 2ServP achieves an improvement in email service level of roughly 9%. More notable is the reduction of interruptions which for that same row is roughly a 1000 less than 1ServP. This shows that 2ServP is able to half the amount of email interruptions. The occupancy ratio for both groups combined is also much lower in 2ServP, averaging around 0.7. While the occupancy of group 2 is always around 90% in a real environment agents can be rotated to even out the occupancy among agents.

Experiment 3: reduced shift exploration

Due to the observed complexity of the problem in the second experiment and the observed power of 2ServP in experiment 2.2 a new experiment will be performed. The simple schedules used for 2ServP and 2Serv!P in the first experiment did not improve on their simpler variants. Searching the entire solution space in order to see if there exists at least one solution would require multiple days as there are 15~ million different solutions. The amount of solutions is based on the number of agents assigned to a group per hour. In the original model there are 9 slots of an hour. By increasing the shift length from 60 to 90 minutes we reduce the model to 6 shifts. The result of this is that the solution space is decreased to roughly 60.00 -150,000 solutions (depending on the policy). This drastically reduces computation time to a few hours for calculating all possible configurations for 2ServP, 2Serv!P and 2ServSep (where agents handle strictly 1 job type). For computational reasons only call centers of the "small" size will be evaluated.

It is expected that longer shifts decrease flexibility (and therefore power) of the complex models. This is because the simple models have one pool of agents for calls and emails. This allows the simple models to have enough agents for emails when its busy, and in the case for 2Serv!P less agents working on emails when there are many incoming/waiting calls. The complex models rely on smart allocation of agents. By decreasing the number of shifts the flexibility of the policies decreases, hopefully this doesn't eliminate configurations that improve their simpler variants.

The arrival rates are also affected by the reduced schedule. As the amount of agents and arrival rates are part of periods within the call center. The new arrival rates consist of the original arrival rate + 0.5 of its neighbor's arrival rate. The same was done for the total amount of agents per hour while rounding down below .5.

To find the optimal schedule, each solution (schedule) is simulated 10 times with a call and email multiplier of 1.3 (medium arrival rate). To obtain the best solutions for each policy all solutions performing with a call and email service level above a certain percentage (depending on the success of the policy) where selected. All selected schedules were and simulated again, this time for 100 simulation runs to avoid any "Lucky Tickets" (A situation where 1 or more configurations benefits from lucky draws generated by the random number generators).

The best schedules are selected based on a common baseline for service level of calls & emails. For instance we filter all entries where the service level of calls and emails > 0.90, 0.80, 0.70. the highest possible. Then take the entry with the highest combined score. All the best solutions are contained in the attachments of this documents. Complete tables with confidence intervals can be found in the *Appendix* under Reduced Shift Results.

Policy	callMult	emailMult	Call SL	Email SL	SoA Call	SoA Email	Occ1	Occ2	Abandon.	Interpt.
1Serv!P	1.2	1.2	0.836	0.998	0.181	9.913	0.668	NaN	35.9	0
1Serv!P	1.2	1.3	0.819	0.998	0.200	10.373	0.690	NaN	41.69	0
1Serv!P	1.2	1.4	0.801	0.997	0.222	10.841	0.709	NaN	43.57	0
1Serv!P	1.3	1.2	0.824	0.998	0.193	10.157	0.684	NaN	41.14	0
1Serv!P	1.3	1.3	0.807	0.997	0.211	10.712	0.708	NaN	46.32	0
1Serv!P	1.3	1.4	0.793	0.996	0.229	11.114	0.725	NaN	51.13	0
1Serv!P	1.4	1.2	0.815	0.998	0.202	10.281	0.705	NaN	47.19	0
1Serv!P	1.4	1.3	0.796	0.997	0.223	11.119	0.724	NaN	51.49	0
1Serv!P	1.4	1.4	0.779	0.996	0.242	11.441	0.743	NaN	55.92	0

Table 17. The results of one group of servers handling calls and emails without pre-emption. The agent schedule used is {0, 7, 7, 8, 8, 5, 4, 3}.

Table 8 and 9, display the output of 1ServP and 1Serv!P. The results are similar to the earlier observed results in *Tables 1 and 2*. The results of these tables only serve as comparison for 2 group policies.

Policy	callMult	emailMult	Call SL	Email SL	SoA Call	SoA Email	Occ1	Occ2	Abandon.	Interrupt.
1ServP	1.2	1.2	0.943	0.939	0.068	65.263	0.860	NaN	9.69	1523.72
1ServP	1.2	1.3	0.942	0.898	0.072	89.869	0.899	NaN	9.73	1722.08
1ServP	1.2	1.4	0.942	0.784	0.069	130.502	0.931	NaN	9.28	1888.03
1ServP	1.3	1.2	0.927	0.838	0.094	109.008	0.906	NaN	14.89	1852.77
1ServP	1.3	1.3	0.925	0.701	0.096	154.535	0.935	NaN	14.5	2016.7
1ServP	1.3	1.4	0.927	0.531	0.092	210.210	0.951	NaN	13.92	2106.05
1ServP	1.4	1.2	0.903	0.646	0.130	169.342	0.940	NaN	21.38	2127.67
1ServP	1.4	1.3	0.909	0.471	0.120	228.146	0.951	NaN	19.44	2210.8
1ServP	1.4	1.4	0.905	0.375	0.130	282.138	0.957	NaN	20.78	2244.56

Table 18. The results of one group of servers handling calls and emails with pre-emption. The agent schedule used is {0, 7, 7, 8, 8, 5, 4, 3}

Table 10 displays the results of 2ServSep obtained using the optimal schedule. The divided agent groups seem to lack the power and flexibility observed in the 1Serv policies. Contrary to all other policies 2ServSep doesn't obtain a call service level of at least 0.8. However the results seem better than in experiment 2. The fluctuations in service levels in context with the arrival multipliers shows that the schedule is not resilient to changes in the arrival rates. Making this the least favorable option of all policies for small call centers. Serving calls and emails separate might only be interesting in the following cases:

- The speed of answer of emails is irrelevant.
- When there are only very few emails arriving.
- With greater granularity in shifts (which drastically increases the amount of schedules).

Policy	callMult	emailMult	Call SL	Email SL	SoA Call	SoA Email	Occ1	Occ2	Abandon	Interrupt.
2ServSep	1.2	1.2	0.788	0.922	0.304	90.103	0.565	0.895	47.92	0
2ServSep	1.2	1.3	0.788	0.786	0.308	136.229	0.568	0.931	47.96	0
2ServSep	1.2	1.4	0.787	0.644	0.305	173.530	0.568	0.944	48.2	0
2ServSep	1.3	1.2	0.737	0.929	0.398	87.785	0.607	0.895	68.06	0
2ServSep	1.3	1.3	0.736	0.841	0.401	124.818	0.611	0.931	68.09	0
2ServSep	1.3	1.4	0.738	0.606	0.395	183.494	0.611	0.946	68.55	0
2ServSep	1.4	1.2	0.687	0.922	0.492	90.516	0.647	0.895	91.63	0
2ServSep	1.4	1.3	0.688	0.809	0.493	132.319	0.648	0.929	93.43	0
2ServSep	1.4	1.3	0.687	0.815	0.490	130.392	0.648	0.928	94.3	0
2ServSep	1.4	1.4	0.684	0.624	0.499	178.028	0.649	0.944	94.86	0

Table 19. The results of two groups of servers, one handling strictly calls and the other strictly emails. The agent schedule used is { {0, 3, 5, 5, 4, 5, 3, 3},{0, 4, 2, 3, 4, 0, 1, 0}}

Table 11 displays the results of the optimal version for 2Serv!P. The schedule used performed best for a call and email arrival of 1.3. Just like its simpler variant 2Serv!P seems relatively (to the other policies) resilient against the increase of calls and emails. As the selected schedule is the optimal schedule in terms of call and email service level. The service level of calls remains steady around 0.8. The speed of answer for email is significantly higher than 1Serv!P, the result of this is that the service level of calls is also slightly better, around 3 %. There also seems to be a significant decrease in the amount of abandonments.

The results of 2Serv!P make this policy favorable over its 1 group variant. Favoring pre-emption over not pre-emptive still remains to be concluded in a simulation closer to reality. As the amount of interruptions visible in *Table 9* might take enormous toll on the sanity of the employees.

Compared to 2ServSep, 2Serv!P seems like a very attractive alternative with a stable promise in service levels for both calls and emails. The amount of abandonments is considerably less compared to 2ServSep and slightly better than 1Serv!P.

The reduce in stress on the agents is somewhat debatable as the amount of agents in each group differ. The stress on individual agents could be managed by swapping agents in and out of groups. This would require additional mechanisms that might not be present in a small call center. As a standalone solution 2ServP offers a more desirable outcome.

Policy	Call Mult	Email Mult	Call SL	Email SL	SoA Call	SoA Email	Occ1	Occ2	Abandon.	Interrupt.
2Serv!P	1.2	1.2	0.863	0.967	0.145	34.136	0.451	0.730	24.43	0
2Serv!P	1.2	1.3	0.848	0.966	0.160	39.056	0.452	0.772	28.32	0
2Serv!P	1.2	1.4	0.835	0.965	0.175	42.662	0.449	0.797	31.1	0
2Serv!P	1.3	1.2	0.842	0.967	0.169	34.885	0.475	0.745	31.19	0
2Serv!P	1.3	1.3	0.828	0.966	0.184	39.307	0.473	0.781	33.76	0
2Serv!P	1.3	1.4	0.813	0.965	0.200	44.869	0.473	0.810	37.97	0
2Serv!P	1.4	1.2	0.822	0.966	0.195	35.658	0.497	0.755	38.46	0
2Serv!P	1.4	1.3	0.809	0.966	0.206	39.988	0.497	0.785	39.44	0
2Serv!P	1.4	1.4	0.788	0.966	0.230	47.072	0.496	0.823	45.8	0

Table 20. The results of two groups of servers, one handling strictly calls and the other calls and emails without pre-emption. The agent schedule used is { {0, 3, 5, 4, 0, 0, 4, 3}, {0, 4, 2, 4, 8, 5, 0, 0} }

Policy	callMult	emailMult	Call SL	Email SL	SoA Call	SoA Email	Occ1	Occ2	Abandon.	Interrupt.
2ServP	1.2	1.2	0.944	0.957	0.068	60.919	0.579	0.821	8.89	608.54
2ServP	1.2	1.3	0.943	0.929	0.068	80.075	0.580	0.877	9.42	686.81
2ServP	1.2	1.4	0.947	0.885	0.064	101.892	0.572	0.911	9.36	728.21
2ServP	1.3	1.2	0.927	0.931	0.092	76.764	0.598	0.867	14.48	746.26
2ServP	1.3	1.3	0.928	0.884	0.091	101.770	0.597	0.918	13.9	823.78
2ServP	1.3	1.4	0.927	0.740	0.093	147.435	0.600	0.945	14.3	859.82
2ServP	1.4	1.2	0.909	0.836	0.122	118.032	0.616	0.913	19.68	913.63
2ServP	1.4	1.3	0.909	0.708	0.119	158.397	0.622	0.938	18.96	941.75
2ServP	1.4	1.4	0.908	0.506	0.123	219.947	0.620	0.959	20.37	983.1

Table 21. The results of two groups of servers, one handling strictly calls and the other calls and emails with pre-emption. The agent schedule used is { {0, 2, 3, 2, 2, 4, 1, 3} ,{0, 5, 4, 6, 6, 1, 3, 0}}

Just as in experiment 2.2 2ServP shows promising results. Up until a call and email arrival multipliers of 1,3 the output parameters are balanced around 90%. Combined with the lowest abandonment rates, this is arguably the best performing policy of all. It is able to decrease the amount of interruptions of 1ServP by more than half. This makes it interesting to pursue research in a more realistic setting. Depending on how many interruptions agents are mentally able to deal with.

Furthermore 2ServP seems quite resilient to the amount of incoming calls. This is not the case for the email service level which quickly declines starting arrival multiplier 1.3 for calls and 1.4 for emails. This is not necessarily as it is most likely the case that calls have priority over emails.

The occupancy ratios for group 2 are extremely high and the majority of the agents is assigned to group 2. Rotation of agents between group 1 and 2 might reduce stress however this is expected to be less compared to the other policies. In reality this means that agents might be under too much stress.

Evaluation

Experiment 3 shows promising results for call and email blending and verifies the hypothesis made in the experiment setup. Having a second agent group that handles strictly calls can protect the underperforming group (the underperforming group depends on the policy) and acts as a buffer, while maintaining flexibility. An example for this are the results of 2ServSep, the results in terms of service levels are worse than for any other policies. There simply aren't enough resources to maintain both parameters at levels that occur in ServP and !P.

Blending calls and emails while employing a second agent group improves performance of the call centers. With the right schedule 2ServP is able reduce the stress on agents that are handling emails and decreases the interruptions by more than half compared to 1ServP. Reducing the amount of interruptions improves the service level of emails as the time lost due to switching between calls and emails decreases. The amount of interruptions is expected to cause frustration with the agents, making a policy as 1ServP with 1500+ interruptions unfeasible. 2ServP reduces the amount of interruptions by more than a half, this might make blending with pre-emption feasible. If this remains a problem measures might be taken, such as as limiting the amount of interruptions, or introducing a third agent group that strictly handles emails.

Compared to 2ServP, 2Serv!P trades interruptions for an increased amount of abandonments, lower service level for calls and higher service levels for email. Performance parameters are better than for its one group variant and its service levels remain fairly stable under increasing pressure of incoming

calls and emails. Without interruptions this would possibly be the best approach. A possible downside of 2Serv!P and 1ServP is the decline of the call service level. Once the call service level declines to undesirable levels it will be difficult to redress the service level. This should be overcome by rule or by policy to halt the handling of new emails for part or for the entire email group.

While 2ServSep benefits from its improved schedule it doesn't compete with any of the blended policies. Blending calls and emails is in the scope of this experiment the preferred method. All of the blended policies improved over the separate handling of calls and emails.

Conclusion

Although This simulation study lacks important factors that occur in real call centers, this study shows the prospect of blending calls and emails. The features that are not included in the simulation and that discern this simulation from reality are stated below:

- Shrinkage, the unexpected absence of agents.
- Scientifically measured job switching cost. The time it takes to switch between jobs.
- Switching cost obtained in the situation where: An agent working on email is interrupted by an incoming call.

During this study two forms of call and email blending have been researched. The two forms can be discerned by either handling calls with (ServP) or without pre-emption (Serv!P). With preemption means that emails are handled within idle time and when needed an agent will interrupt its email to handle an incoming call. The other option is to wait until the email is finished or another agent becomes available. In this study the effects of the two policies are compared and improved by adding an additional agent group, that strictly handles calls.

Blending calls with or without pre-emption results in two different outcomes, pre-emption favoring calls and non-preemptive email. To choose one over the other depends on the priorities of the call center and on the realistic feasibility of handling calls with pre-emption.

Handling calls with pre-emption causes interruptions of the email which increases the total time spent on an email. The interruptions and resumptions of emails that occur when handling emails with pre-emption result in an increased workload on the agents. While this is undesirable, it might also be mentally infeasible, as a large amount of interruptions might cause frustration and dissatisfaction.

Undesirable or not this shows that blending calls and emails in call centers can (almost) eliminate idle time. Besides the interruptions and the high occupancy ratios being an obstacle 1ServP is able to score exceptionally well on calls with the ability to maintain emails at a decent service level (depending on the arrival rates) contrary to handling both call and email separately.

1Serv!P on the other hand doesn't favor calls as 1ServP does. This combined with an acceptable response time of 4 hours for email results in an opposite situation of 1ServP. The email service level is especially high while maintaining a reasonably steady call service level over a range of varying arrival rates. While this might not be the preferred order of priorities, 1Serv!P does not allow for interruptions which greatly reduces the occupancy ratio by 20%~. This allows for a steady flow of outgoing emails and eliminates the possibility of an ever growing backlog of emails.

Having calls and emails handled separately (2ServSep) is the least efficient method even with an (near) optimal schedule. Without any prior knowledge or optimal schedules 2ServSep should be avoided. As handling calls and emails separately loses flexibility over blending, resulting in low service levels.

Introducing a second group for both 1ServP and !P can increase the performance of their respective 1 group variants. 2Serv!P compared to 1Serv!P is able to reduce the speed of answer of calls and therefore the amount of abandonments. This increases the service level of calls while maintaining a high service level for emails. Especially 2ServP is able to make a difference, as it is able to drastically reduce the amount of interruptions and increase the email service level by a fair amount. This however requires an adequate agent schedule with respect to the blend policy. The impact the agent schedule has on each policy differs. As 2ServP benefits from fewer agents in the agent group that handles strictly calls and 2Serv!P requires more balance over both groups. Having less than adequate schedules may do more harm than good.

Discussion

In the proposal a couple of promises were made that eventually didn't make it to the simulation/experiment. First and foremost this would be large call centers. Having more than 20 agents would have been interesting and possibly lead to different outcomes. For instance 20+ agents would give a lot more flexibility in scheduling a couple agents for email. Alas after finding out in experiment 2 that the problem is more complex than initially thought, it became infeasible to generate schedules for large call centers.

Regression analysis is another topic didn't work out as planned. The factors chosen for the experiments and those that remained after the second experiment weren't the most insightful for a regression analysis (arrival intensities for call and email and varying policies), especially without large call centers. The input factors were designed to see how different policies in both large and small call centers compare. Without data on large call centers, only combinations of different policies with varying arrival rates were left to compare. Stating that there is some correlation between the arrival intensity and the service level is too obvious, this can easily be seen in the tables and images of experiment 1 and 2.

Furthermore I wanted to include shrinkage and doubly stochastic processes, both for realizing a more realistic simulation . These both required more time than available. Measuring the actual amount of shrinkage (enough for a tight validation) would require multiple days on-site measuring time agents spent away from their desk, and back and forth communication with a manager to verify my conclusions. Doubly stochastic processes, where one stochastic process determines a parameter in a subsequent probability distribution. Losing a lot of time to the hard to fit and later replaced distribution for email handling time. Left no time for doubly stochastic processes. Although the variance introduced by these double stochastic processes might give more realistic results, this would also make results harder to interpret. In my opinion doubly stochastic processes aren't missing in this simulation and would be more fitting in a situation where one is to improve/optimize a specific call center as this study was originally thought to be.

In experiment 2.2 the generation of schedules using hill-climbing algorithms is discussed. In the light of the results of experiment 3 this doesn't add much value to the research. Instead I could've moved on to experiment 3 directly. However, during the time of writing I was unsure that reducing the amount of shifts would lead to improving solutions (neither by using a hill-climber of which its success is to be debated). Even though the hill-climber succeeded in making a good schedule, comparing policies with optimal schedules would lead to more interesting results. The lucky draw obtained by one of the two hill-climbers served as a proof of 2ServP's power and as the motivator to compute the entire solution space, valuable time I would've otherwise used to pursue something else.

The absence of shrinkage is in my opinion the heaviest missing factor. This has made a tight validation of the call center impossible. The size of the call center certainly plays a role here, if you have only five agents and one is missing the workforce is reduced by 20%. This certainly has a high impact on a call center. However accurately measured shrinkage would've required several session onsite at the call center, which was not possible. A realistically modelled and validated call center simulation combined with actual switching cost would otherwise make my research directly usable. However this leaves the door open for future work on this topic, which in my opinion shows promising results.

Future Studies

Identifying the true switching cost that occur by blending call and emails is key to measuring the success of policies that employ pre-emption. Combining the results of this with a simulation more akin to reality will either prove or break the results of this research. However if agents are capable of handling the amount of interruptions that occur in 2ServP and the results of this research is translated to a real environment, this would result in a huge increase of performance and change the way workforce is managed.

An alternative to blending calls with emails is to blend webchat with emails. As these tasks involve the same type of action, typing messages. Helping a customer via webchat involves waiting for answers during which emails could be continued, possibly making interruptions less “definitive”. This would however require a considerable amount of agents handling webchat or vice versa a small amount of email arrivals (making the need for 2ServP and the likes obsolete).

The first problem encountered during the study of the 2 group policies was the lack of schedules for large call centers and the huge amount of time searching for a good schedules regarding small call centers. Splitting the day in more shifts gives more flexibility but also increases the solution space. A way to determine what granularity is needed for a schedule to perform better than a one group policy and a method of generating schedules is required, were the success of this research proven in real call center.

Experiment 1 where 1ServP is compared to 1Serv!P leads to the impression that if these two policies were to be combined the best of both worlds can be achieved. A mechanism that dynamically switches between policies or assigns a subset of agents not to be interrupted while working on email depending on service level or arrival rate. A mechanism like this would give more control over priorities in service level and eliminate the need for complex schedule generators, or at least limit the solution space for good schedules.

References

[1]. K. Kotiadis., and S. Robinson. 2008, Conceptual Modelling: Knowledge Acquisition and Model Abstraction, Winter Simulation Conference.

[2]. S. Akhtar., M. Latif. 2010, Exploiting Simulation for Call Centre Optimization, Proceedings of the World Congress on Engineering.

[3]. J. Pichitlamken., A. Deslauriers., P.L'Ecuyer., A.N. Avramidis. 2003, Modelling and Simulation of a Telephone Call Center, Winter Simulation Conference.

[4] V. Methora., J. Fama. (2003), Call Centre Simulation Modelling: Methods, Challenges and Opportunities, Proceedings of Winter Simulation Conference.

[5]. R. Ibrahim., H. Yen., P. L'Ecuyer., H. Shen. (2015). On the modelling and forecasting of call centre arrivals. International Journal of Forcasting.

[6]. E. Buist., P. L'Ecuyer. (2005) A Java Library for Simulating Contact Centers. Proceedings of the 2005 Winter Simulation Conference.

[7]. A. M. Law., W. D. Kelton. (2015). Simulation modelling and analysis (5th edition). NewYork: McGraw-Hill.

[8]. G. Koole., and A. Pot. (2006). An overview of routing and staffing algorithms, Department of Mathematics, Vrije Universiteit Amsterdam, The Netherlands.

[9]. B. Russel R. (2004) Designing Simulation Experiments. Proceedings of the 2004 Winter Simulation Conference.

[10]. T.A. Mazzuchi., R.B. Wallace. (2004), Analysing skill-based routing in call centers using discrete event simulation and Design Experiments. Proceeding of the 2004 Winter Simulation Conference.

[11]. S. Bhulai., G. Koole. and A. Pot. (2008), Simple Methods for Shift Scheduling in Multiskill Call Centers, Manufacturing & Service Operations Management

[12] T.R. Robinson. And D.J. Medeiros. (2010), Does the Erlanc C Model Fit in Real Call Centers? Proceedings of the 2010 Winter Simulation Conference.

[13] B. Legros., O. Jouini., G. Koole. (2018) Blended call center with idling times during the call service, IISE Transactions.

[14] Akaike, H. (1973), "Information theory and an extension of the maximum likelihood principle", in Petrov, B. N.; Csáki, F., 2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR, September 2-8, 1971, Budapest: Akadémiai Kiadó, pp. 267–281

[15] Delignette-Muller, M.L., Dutang, C. (2014). fitdistrplus: An R Package for Fitting Distributions. Journal of Statistical Software vol 65, issue 4.

[16] Neath, A,A., Cavanaugh, J.E. The Bayesian information criterion: background, derivation, and applications(2012). WIREs Comput Stat2012 , 4:199–203. doi: 10.1002/wics.199

[17] T.A.Koka , V.H.Badshah and R.A. Single and Multi Server Queuing Models: A Study(2017). International Journal of Mathematics And its Applications, Volume 5, Issue 4–D.