UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Flexibele capaciteit en wachttijden: een educationele tool

Egon Nuyts

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Inneke VAN NIEUWENHUYSE

2019

2020

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Flexibele capaciteit en wachttijden: een educationele tool

Egon Nuyts

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Inneke VAN NIEUWENHUYSE

Flexible versus dedicated resources in call centers: an educational tool

Egon Nuyts
Business Economics
Faculty of Business Economics, Hasselt University

Telephone call centers have grown world-wide during recent years and became a crucial contact point with customers. Streamlining the performance of these call centers has become an important aspect in order to maximize profit and customer satisfaction. Call center managers are on a permanent quest to reduce the average waiting time and the abandonment of their customers. The general perception seems to exist that combining multiple queues into one pool would improve the efficiency in a call center and more generally in a service environment. By pooling the appropriate demand on one merged agent group, it is assumed that waiting times will decline. In this paper we will use simulation models to prove this reasoning to be short-sighted and that there is no uniform answer. Furthermore, we will evaluate a simple overflow case and a scenario where overflow is combined with priority rules. Through simulation, it is illustrated that with the right threshold policy, the results of the pooled and unpooled scenario can be outdone. Nevertheless, the understanding of the possible unfavorable consequences of bad threshold policies are examined as well, in order to provide a complete view of the possible outcomes the decisions of call center managers can have on the system's performance. With this paper comes a template excel-file and all necessary simulation models so the reader himself can carry out the experiments and deduce the consequences and insights.

Keywords: Simulation, supply chain management, queueing theory, telephone call center, pooling demand, overflow, service operations, capacity flexibility

1. Introduction

Globalization has stressed the importance of operating practices if companies want to stay competitive. This is true not only in a manufacturing context but for service providers as well. The purpose of this paper is to provide insights with regard to demand pooling in such a service environment. Most companies divide service agents into departments according the type of customers they handle. To improve the performance, one could pool a set of departments into a larger one that handles all of the combined customer types (Tekin, Hopp, & Van Oyen, 2009). However, different customer types require different skills. In order to pool those departments, all agents involved have to be able to handle all of

Egon Nuyts
Flexible versus dedicated resources in call centers: an educational tool
Promoter: Prof. Dr. Inneke Van Nieuwenhuyse

This master thesis was written during the COVID-19 crisis in 2020. This global health crisis might have had an impact on the (writing) process, the research activities and the research results that are at the basis of this thesis.

the possible incoming calls, i.e., they have to be cross-trained. To elucidate the insights originating from this setting, we will use the example of a call center which focuses exclusively on inbound phone calls.

With applications in emergency dispatching, technical support and customer services among others, the call center industry is an interesting and important sector that keeps expanding. Companies acknowledge that call centers play an important role in their customer value delivery chain but often find it difficult to cope with the dynamic demand and operational complexity it entails. A substantial part of this complexity is related to finding the right balance between high employee satisfaction, excellent service quality and low operating costs, while dealing with multiple types of incoming calls. A possible partial solution which call center managers often intuitively consider, is pooling their demand over a set of cross-trained resources. If, and under which circumstances, this option will be an improvement of service levels in terms of abandonment and waiting times, shall be the main research question of this paper.

Generally, incoming calls represent various customer needs in several different languages. Providing operators with the skills to handle all of the possible requests in all possible languages is very expensive or even impossible in most cases. For this reason, operators mostly possess one or two skills at which they are specifically trained. If an operator has just one skill (for instance, he/she can only handle sales calls in Dutch), we call it a static of dedicated resource. On the other side, if they are cross-trained and have more than one skill (for instance, he/she can handle sales calls in Dutch and English), they are called flexible resources. They can be allocated dynamically when and where they are needed. Call centers are technologically sophisticated and automatically allocate the customer to a suitable operator when one becomes available, using skill-based-routing. If an operator with the right skillset is ready for use when the call enters the system, the customer will be served instantaneously. Otherwise they will be put in a queue.

Customers in a queue are impatient by nature and will leave if it takes too long. Restricting this abandonment is an important additional concern since leaving customers mean a possible loss of revenue and/or goodwill. Cross-trained operators also have an influence on the expenses of a call center. Learning an additional skill requires training while highly skilled employees expect a corresponding salary. In summary, more flexibility leads to better operational performance but there are costs associated with creating and maintaining this flexibility (Aksin & Karaesmen, 2002). Because call center managers strive for low operating cost in combination with a high quality of service, we will provide a comparison of waiting times as well as a cost-oriented overview of different scenarios.

Using the simulation software *Arena*, we will develop an educational tool which will demonstrate our findings in a comprehensive way. The used dataset contains all the incoming calls at the telephone call center of "Anonymous Bank" in Israel during January third 1999. The results of this simulation study confirm the call center operations management literature and augment it by providing understandable and clarifying awareness of the matter to laymen.

The remainder of this article is organized as follows: in section 2, we will discuss the relevant literature. Section 3 will focus on the data we used and the simulation models we built with it. The results originating from those simulations will be presented in section 4, followed by the conclusions and insights we can get out of them, in section 5.

2. Problem statement and literature review

This section provides a more detailed explanation of demand pooling, followed by a closer look at the related scientific literature. First, we will clarify some terminology:

- “Agents” or “operators” in this paper stands for the servers which handle the incoming calls. It refers to the call center employees. They are bundled in agent groups according to the call type they handle.
- “Cross-training” is a term that implies teaching an employee an additional skill, not belonging to his initial job description. It involves teaching new processes and improves the skills of employees.

2.1. Problem statement

Call center executives and managers seek the right combination between three components: relatively low operating costs, a sufficient job satisfaction for their employees and a satisfactory service quality for their customer. The latter is a combination of restricted waiting times and a service which actually helps the caller solve his problem. Pooling the demand is a frequently contrived working method to partially handle these obstacles. If we assume that our operators need only one skill to handle all calls, the unpooled situation is presented by Figure 1a and the pooled variant by Figure 1b..

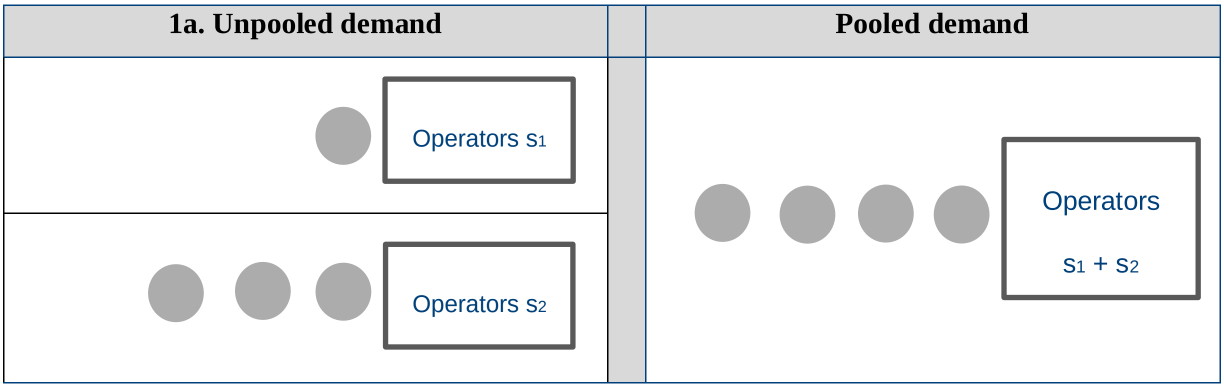

Figure 1: Pooling with 1 call type

The literature confirms the intuitive feeling that pooling with a single call type indeed is beneficial. It shortens the average waiting time, which leads to a better service quality. Being able to handle more calls in a given time period, means that call center managers will need less operators to achieve the same service level as before. In this way, pooling already offered a partial solution for two out of three objectives, only job satisfaction didn't improve explicitly.

So far, we have ignored the fact that in reality often multiple call types arrive at a call center. Multiple call types in one call center implies the need for multiple skills among the available agents. Simplified, the situation without pooling or cross-training would look like Figure 2a.

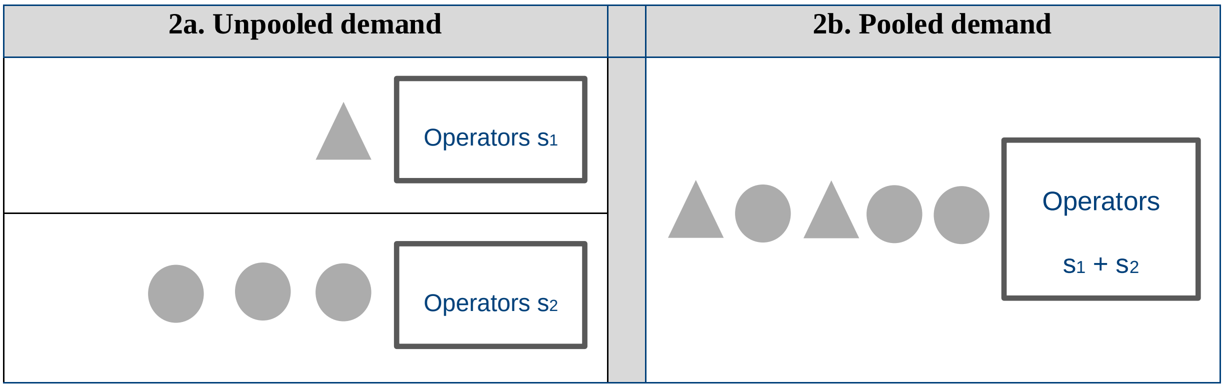

Figure 2: Pooling with 2 call types

If all operators with the required skill are busy, their queue can explode while other operators are having less work or even are idle. Pooling the incoming calls and cross-training the operators, Figure 2b, will provide call centers with the essential flexibility to assign idle or less busy operators where needed. Teaching operators various skills doesn't only affect the waiting time but often improves job satisfaction as well. It brings variation, makes a shift less tedious and can even influence career paths.

However, mixing multiple call types and cross-training operators can lead to negative results as well, i.e., it can cause waiting time to increase (van Dijk & van der Sluis, 2008). This paper will illustrate the advantages discussed above but will also investigate when pooling demand can be counter-productive and introduce some other options to improve the situation.

2.2. Literature review

The mathematical and operational complexity of call centers make it difficult for decision makers to fully grasp the dynamics. During the last two decades, modelling call center operations has been the subject of plenty of academic research. An early overview of the state of this research is provided in Gans, Koole, and Mandelbaum (2003). A more recent literature survey on call center operations management can be found in Aksin, Armony, and Mehrotra (2007). This survey mainly pays attention to management challenges caused by emerging technologies, behavioral issues of operators and customers, and to the interface between call center operations and sales and marketing. Both surveys show that not only conceptual models exist, but simulation and mathematical models are common as well.

The efficiency benefits achievable by combining specialist operators with a pool of cross-trained ones, is examined by Tekin et al. (2009). By means of standard queueing approximations and numerical analysis, they explore the impact of certain system parameters on the decision of which departments to pool. It demonstrates that if certain conditions regarding mean service times, the number of servers in a department and variation of service times are fulfilled, pooling queues effectively has operational advantages. The findings in this paper will be important reference points for our educational tool.

In an earlier study, Hopp et al. (2004) stated that partial cross-training can be nearly as effective as full pooling, i.e., cross-training only a fraction of the employees and keeping the others single-skilled can provide the desired improvement as well. Figuring out the appropriate degree of flexibility is one of the key questions an operations manager has to deal with. Finding the right combination between dedicated and flexible operators is therefore often treated in studies. In the light of this problem, Aksin and Karaesmen (2002) demonstrate that certain flexibility principles from a manufacturing environment also hold in a call center context. It is shown that smart limited flexibility is almost as good as full flexibility and that systems with smaller scale will benefit more from a careful design of their flexibility levels and structures, compared to systems that operate at large scale. Limiting the degree of flexibility will also decrease the loss of expertise of dedicated operators, i.e. specialists. Further research about the limited flexibility concluded that, if even possible, teaching all skills to every operator is not cost-effective. They stated that it suffices to teach every operator two different skills (Aksin, Karaesmen, & Lerza, 2006; Tekin et al., 2009; Wallace & Whitt, 2005). This will provide the majority of the benefits, while additional skills have a relatively low payoff. Using simulation, David and Bastian (2016) confirms that bi-skill call centers are economically better in the long run compared to full-skill or single skill call centers. Not only limiting the amount of skills is an improvement, also limiting the proportion of cross-trained operators seems beneficial (Robbins, Medeiros, & Harrison, 2010). Aksin, Karaesmen, and Örmeci (2005) as well as Chevalier, Shumsky, and Tabordon (2004) conclude that a 80/20 rule of thumb is near-optimal for a wide variety of parameters: unless training is for free, educating 20% of the operators to be flexible will result in the highest benefits while keeping costs to a minimum.

All of the above applies in situations where pooling is beneficial, i.e., when the setting is right so pooling improves average waiting times. Operator departments however, differ in parameters such as mean service times, arrival patterns of calls, the number of operators and the variability in service time.

All of these have an impact on the effect of pooling certain departments and whether it will provide an advantage or not. In another context, Smith and Whitt (1981) demonstrated that pooling could be counterproductive when service rates differ. We can extend this to call centers: if department A would have a very long service time and department B a very short one, FIFO would lead to longer waiting times for customers of department B instead of an improvement, if the two departments were pooled. Those customers would then have to wait occasionally in line behind others who occupy the servers substantially longer than their own kind of customers would. This will almost always disadvantage the customers with a shorter service time. Depending on the size of the customer segments, the average waiting time will increase or decrease. Pooling will always be favorable for two departments when the service time of one department is not larger than six times the service time of the second. Satisfying this condition, pooling departments with the highest squared coefficient of variation in service times will improve the waiting time the most (Tekin et al., 2009). Besides fully pooled or unpooled departments, other scenarios exist as well. Using an overflow of calls between departments and potentially giving them priority labels, managers can ensure a minimum service variability in the unpooled case while possibly exceeding the performance of both the pooled and unpooled scenario (van Dijk & van der Sluis, 2008). Figure 3a visualizes the one-way case where the overflow of cases can only occur from one queue while in Figure 3b, both queues can be served by both agent groups. The overflow in the one-way case as well as in the two-way scenario will only take place when the right conditions concerning idleness and priorities are fulfilled: with simple overflow, calls can only follow the dotted line when the other operators' queue is empty, i.e., when they are idle. When adding priorities calls can flow over when certain thresholds are met. This will be handled in detail further on in this paper.

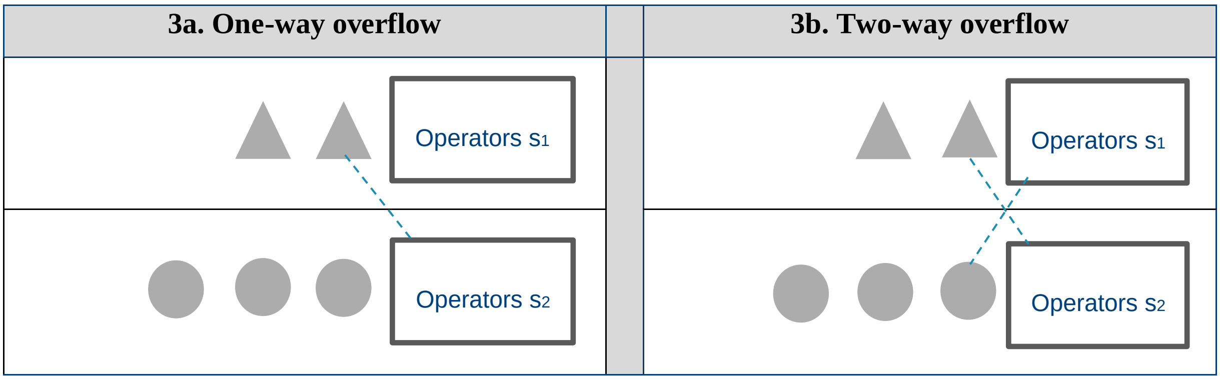

Figure 3: Overflow

As stated before, job satisfaction should be an important target in a service context since satisfied operators will generally do a better job. Call centers have fairly high employee turnovers and burnouts occur frequently. Emotional exhaustion, due to not having the time or right skill to complete a task, and diminished personal accomplishments , i.e., executing the same tasks repeatedly while rarely encountering and overcoming new problems, are two immediate causes for this problem (Aksin et al., 2007). Cross-training can be used as a remedy for both causes. Learning how to solve new problems and maybe discover new talents will strengthen personal accomplishments. With the help of efficient skill-based routing, call center managers can reap the benefits of flexibility: it allows them to route incoming calls to operators with suitable skills, while controlling the workload of their operators (Aksin et al., 2006). An excessive workload comes with a big risk of mental overload. It has been proven that when the operators' utilization rate exceeds 88% for a longer period of time, burnouts arise (Ord, 2016). To further reduce mental overload, Sisselman and Whitt (2007) propose an alternative method of routing. Instead of "simple" skill-based-routing, they suggest letting operators participate in the call-assigning process by expressing their preferences. This preference-based routing method would empower the operators but does not guarantee that all requests will be granted. In our simulation, we will disregard this alternative routing method and stick to regular skill-based routing as this is most used in real call centers.

A significant part of the relevant academic literature refers to or uses simulation models in their research. Because of the complexity of call center operations, simulation is becoming a popular tool to model the stochastic processes call centers are based on and analyze the performance of it under different circumstances. Mehrotra and Fama (2004) provide a framework, describing methods, challenges and opportunities all simulations of a call center environment must contain. Figure 4 represents this framework which will serve as a blueprint throughout this study.

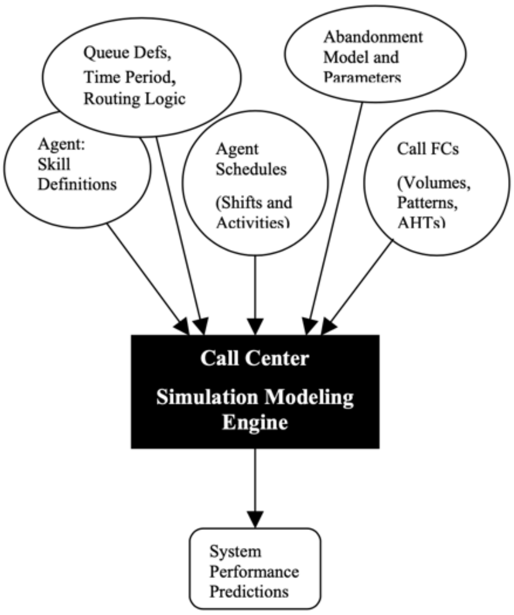

Figure 4: Call center simulation modeling framework (Mehrotra & Fama, 2004)

3. Methodology

3.1. The setting

In order to keep the simulation models and results as close to reality as possible, a dataset was used to forecast calls. The chosen dataset contains various data about the incoming telephone calls at the call center of "Anonymous Bank" in Israel during 1999. It's a smaller call center which provides several different services, handling up to 2000 calls daily. Using the framework in Figure 4, we will further unfold the setting.

3.1.1. Agent skill definition

Supplied services include providing information on and transactions of checking and saving, providing information for potential customers and supporting the website. To efficiently organize the operation, the center has split up these services up into six types of calls. Each call type represents a different skillset required from the agents. Agents with the same skillset will be gathered in an agent group, often called a department in large companies, and thus different call types demand a different agent group. According van Dijk and van der Sluis (2008), restricting the skills per agent to two will more often than not provide the best results. For this reason and to improve understandability, we will narrow the scope down to the two most frequent call types: regular activity and potential new customers. When taking care of regular activity, agents will serve existing customers in their native language. The "potential new customer" type implies providing prospective customers with the information they need, not making a distinction between languages. Both types are served on a First-Come-First-Served basis and are non-preemptive.

3.1.2. Queue definition, time period and routing logic

For our research, we have selected January 3th, the busiest day of the month. Despite the time period being 24 hours, no calls will arrive between midnight and 7:00 AM due to the opening hours of the call center. Although being a Sunday, January 3th 1999 is categorized as a weekday (Sunday to Thursday) thus the call center will be staffed between 7:00 and midnight. However, despite no calls being accepted after midnight, agents will finish the calls they are handling even if it means working beyond closing time.

Calls arrive at the center and when entering the system, they will be dealt with by the voice response unit. A VRU is an automated telephone answering system which uses prerecorded voice messages and the buttons on a telephone to interact with the customer. It will guide them through numerous options in order to get a first glimpse of the customer need and determine the required service. In our VRU the call type will be assessed, after which the call will be transferred to the corresponding operators, i.e., agent group. If there is no agent available from the appropriate group, the call will be held in a queue. Each agent group has its own separate queue. Calls will leave the queue if one of their operators becomes available, if they abandon the system due to long waiting times or in case of overflow, i.e., when an agent from the other agent group becomes accessible. This basic routing logic applies to all scenarios except the one where both types are fully pooled on one combined set of resources. In pooled situations, calls of both types will be put in one and the same queue if none of the agents out of the merged agent groups is directly available. The distribution we use to estimate the time spent using the VRU, is the same in all scenarios and for all cases and is presented in Table 1.

Table 1: VRU distribution

Voice Response Unit
-0.5 + ERLA (1.76, 5)

3.1.3. Agent schedule

For simplicity, we assume all stations are staffed non-stop during the opening hours. Taking into account employee breaks, lunch or shift changes will not offer added value to our educational purpose and will unnecessarily complicate the construction and understanding of our models. The same holds for shrinkage, when scheduled time is not worked because of unexpected absence from work due to causes such as sickness or holidays. Both kinds of lost agent time will not be considered in our simulation models.

To calculate the number of agents needed for both call types an online tool was used which combines the Erlang A and Erlang C formulas. Both are mathematical equations which use various parameters such as the time period, call load, average handling time, maximum utilization, shrinkage, the required service level and the maximum amount of waiting time you wish to accomplish. Erlang C uses those parameters to calculate the required number of agents according the specified parameters. Erlang C provides a good estimate but doesn’t include the number of people abandoning before reaching an agent. Erlang A on its turn, provides a good estimate of the abandonment but tends to underestimate the number of operators needed. It uses the “birth-and-death” process of Markov as basic principle of the formula. Customers entering the queue is seen as a “birth”, while people abandoning represent “deaths”. Combining both formulas will produce the most accurate results about the number of agents required and estimated abandonment. Taking into account the specifications of our call center environment, the online tool recommended to appoint seven operators to handle the regular calls and two operators to serve the prospective customers. In the pooled scenario, this will thus result in one pool of nine cross-trained agents.

3.1.4. Abandonment model and parameters

In our model, we will not include the likelihood of customers calling back after abandoning and only focus on abandonment itself. The moment at which customers will hang up and calls leave the queue differs across industries and even across companies. This makes it difficult to construct or find a uniform assessment method. Rules of thumb are no viable option either, they have a high chance of not being appropriate for the industry or company in our case study. In order to find an appropriate way of implementing the abandonment in our system we will use the available historical data.

Using the historical data of the month January, a life span or patience level could be deducted for both call types. For each call type separately we filtered our dataset, keeping only the calls which left the queue before being served. This leaves us with a range of waiting times of all customers which abandoned. Entering these times into the *Input Analyzer* of *Arena* provides us with the fitting exponential distributions in Table 2.

Table 2: Patience levels

Regular activity	Potential customers
0.9 + EXPO (67.2)	0.99 + EXPO (57.5)

These distributions will be used as a patience level which customer possess. Such a patience level or life span is assigned to each customer entering the queue. When the time spent in the system, a combination of time lost due to the queue as well as time lost to the VRU, exceeds the assigned value, the customer leaves the queue and an abandonment is registered. When recording abandonment, we will keep distinguishing both call types.

3.1.5. Call forecasts

Two major types of call forecasts are required for any basic call center simulation: call volumes and average handling or service times (Mehrotra & Fama, 2004). As stated earlier, historical data is used to determine the call volumes of both types. The data already has been organized at the time but still had some cleaning to do. After removing phantom calls and other irregularities, the dataset consists of 1903 calls which entered the call center on January 3th. 1458 of them are regular customers, which leaves 445 calls of potential customers.

Just as with patience levels, we used the cleaned data of the entire month of January to derive the service time distributions. For both call types separately, service times where extracted out of this cleaned data. According David and Bastian (2016) an exponential, Gamma or lognormal distribution would all offer a fitting solution. All three options were explored using *Input Analyzer*. All of the distributions possess corresponding p-values smaller than 0.005, meaning all three of them do not offer a statistically good fit. *Input Analyzer* does not specify further, as “< 0.005” is the smallest amount possible. This makes it impossible to pick the most optimal distribution based on the provided information. We will consider them equally and pick Gamma distributions for our models.

Table 3: Service time distributions

Regular activity	Potential customers
0.999 + GAMM (175,1.8)	0.999 + GAMM (108, 0.987)

3.2. Scenarios

In order to clarify and educate the consequences of certain decisions in a call center environment, our simulation tool consists of multiple scenarios. All scenarios will be simulated using 75 replications without a warm-up period. Using real data, a warm-up period in which the system should go towards steady state, is not necessary as the model starts equal to the underlying system. Using 75 replications reduces the half width, i.e., variance, adequately while containing the necessary computing time. In section 4 we will compare and clarify the output of the models. When using this research paper as a guideline for educating students about the subject matter, these are the simulation models which students will run themselves in order to obtain the necessary insights. To display all scenarios visually, basic flowcharts will be used. We will examine the flowcharts to get a first glimpse of the dissimilarities in flow between different process variants.

3.2.1. Scenario 1: Unpooled or status quo

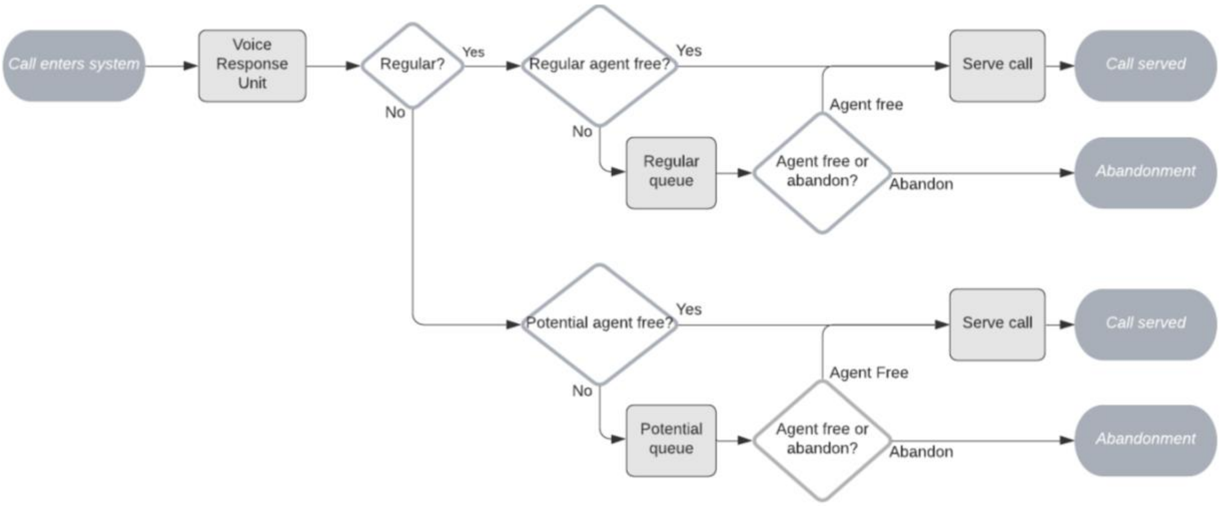

Figure 5: High-level unpooled scenario

Using a dummy at 7:00 AM and the interarrival times we deducted out of the cleaned historical data, calls enter the system as they did on January 3th 1999. The flow of all scenarios starts with this reading of data and a short encounter with the VRU. After this standard procedure, the unpooled situation splits in two parallel subsystems which are completely separated. Using skills-based routing, a call is transmitted to one of them. Skills-based routing allocates incoming phone calls to the right agent group based on the skill the agent groups possess. In our case, the skill needed to solve the customer's problem is reflected by the call type assigned by the VRU.

In both subsystems the call follows the same flow. When entering, an immediate check is performed if an operator of the appropriate agent group is available. If not, the customer is placed in the appropriate queue. "*Regular queue*" contains all customers in queue with call type "*regular*", while "*Potential queue*" consists only out of prospective customer. Both queues are taken care of according the FIFO principle. If no operator becomes available before the customer runs out of patience, he abandons the queue. Otherwise, the call is handled when all its predecessors are dealt with and an agent becomes available. Both paths result in the call leaving the system.

3.2.2. Scenario 2: Pooled

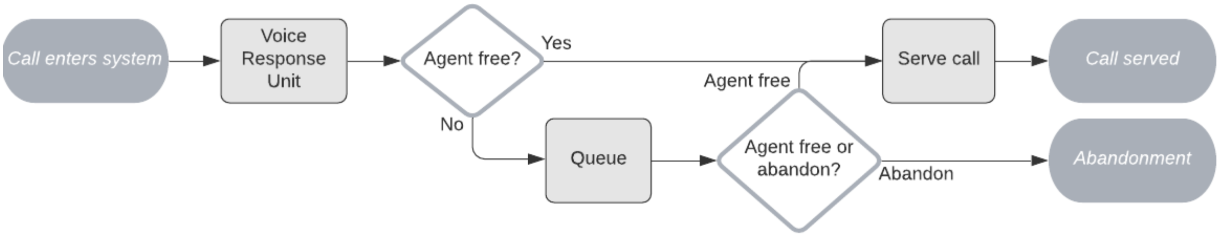

Figure 6: High-level pooled scenario

The pooled scenario resembles the unpooled scenario but lacks the separation into subsystems. In this case, all agents are cross-trained. Both agent groups are able to handle regular customers as well as providing information to potential customers. The relevant agent group in this situation, is therefore a merger of both agent groups out of the unpooled situation. Concretely for the flow, this means that when leaving the VRU, all calls are held in one queue if no suitable operator is available. In scenario 2a we will use the service times deducted from the historical data. Because the service times are quite similar, i.e., none of them is six times larger than the other one, this should improve the waiting time compared to the status quo (van Dijk & van der Sluis, 2008). To illustrate the other situation, when pooling does not provide an improvement, we will add a variant to the original pooled and unpooled scenario, scenarios 1b and 2b. In these variants, we will divide the service time and patience level of the potential customers by two while multiplying those of the regular customers with two. This results in the distributions displayed in Table 3, which fulfill the rule formulated by van Dijk and van der Sluis to become negative results when pooling: six times the potential customer service time is still smaller than the time needed to serve a regular customer.

In order to look at the effect of pooling and overflow at the utilization rate of resources, we will need alternative cases without abandonment. When examining the cases with abandonment, it seems logic that any improvement in waiting times would not lead to a better allocation of workload but will increase the total workload. The improvement in waiting times would lead to less abandonment, which means the same amount of resources would be less idle because they will need to handle more calls. For this reason, the unpooled and pooled cases without abandoning customers will be scenario 1c and 2c respectively. The flow in both cases will be the same as in the appropriate flowcharts, except the gateway for abandonment. Each call will be served by a resource, no matter how long the time spent in a queue will become.

Table 4: Distributions scenario 2b

Service time "Regular"	Patience level "Regular"
0.999 + GAMM (350, 3.6)	0.99 + EXPO (134.4)
Service time "Potential"	**Patience level "Potential"**
0.999 + GAMM (54, 0.493)	0.99 + EXPO (28.5)

3.2.3. Scenario 3: overflow

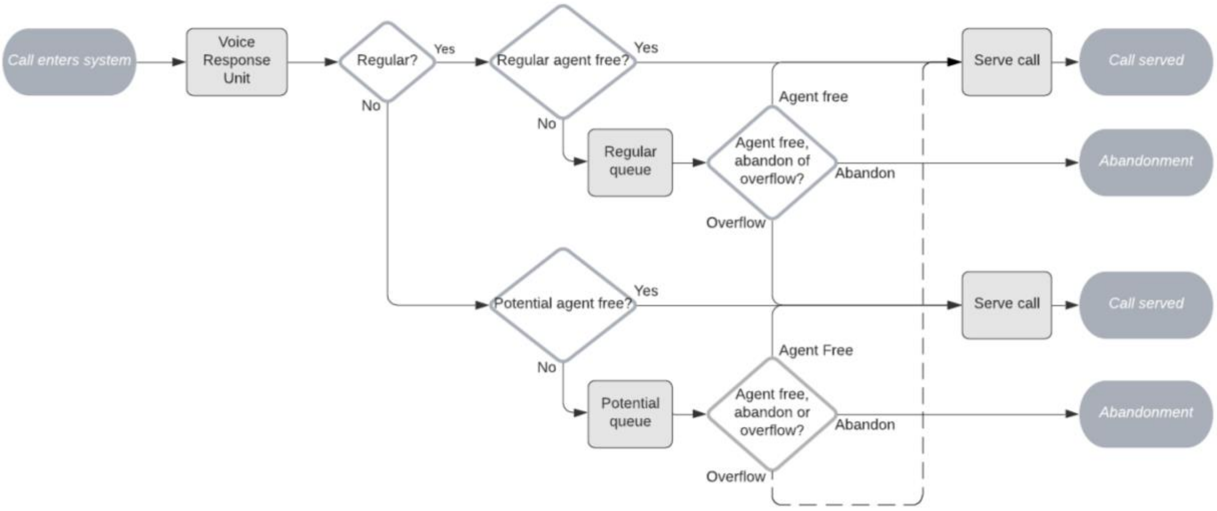

Figure 7: High-level overflow scenario

Pooling is not always a feasible or desirable option. For this reason, both subsystems in this scenario are kept strictly separated just as in the unpooled case. The distinction between both scenarios lies in the possibility of calls to switch between both subsystems. When a queue is empty, the appropriate agent group will serve calls from the other agent group's queue. The priority therefore remains to serve their own call type, but all agents are cross-trained and able to serve both call types. This will reduce the amount of time operators will be idle and will contribute to a better spread of the workload as well. The situation described is in case of a two-way overflow, when both agent groups can take upon calls from the other one. This is will be simulated in scenario 3a and is visualized by Figure 7 when taking into account the dotted line.

When overflow is one-way traffic, we talk of a one-way overflow system. Only one agent group is cross-trained and will receive calls from the other queue when idle. Because of the difference in amount of both call types, the potential customer agent group will be cross-trained in order to help serve the large number of regular customers calling the center. Scenario 3b is represented by Figure 7 when leaving out the dotted line. Equal to scenarios 1c and 2c, scenario 3c will be the overflow scenario modified to leave out abandonment.

3.2.4. Scenario 4: overflow + priority

Besides these cases with simple overflow, further improvements are expected to be accomplished when combining overflow with giving different priority levels to different call types. An agent group does not longer have to be idle in order to receive calls from the other agent group's queue when the latter has a higher priority. Prioritizing a certain call type in combination with allowing overflow forms the fundamental idea of scenario 4 and results in the same flowchart as scenario 3, Figure 7. In order to determine when calls can flow to the other agent group and when priority should be accounted for, a threshold will be introduced: **Thr**$(\theta_1, \theta_2, \Omega_1, \Omega_2)$. In accordance with van Dijk and van der Sluis (2008), this threshold will be defined as

Thr$(\theta_1, \theta_2, \Omega_1, \Omega_2)$: *when a server of server group j (j = 1,2) becomes available, it will give priority to a job of type i, where*

$$i = \begin{cases} 3-j & \left(m_{3-j} \geq \theta_{3-j} \wedge m_j < \theta_j\right) \vee \left(m_j = 0 \wedge m_{3-j} \geq \Omega_{3-j}\right) \\ j & otherwise. \end{cases}$$

with m_i *being the number of calls of type i waiting in the queue.*

Interpreting this definition, θ_i represents the threshold for giving priority, identifying the moment when the agent group will start processing the other call type, while Ω_1 stands for the overflow threshold, when customers will switch queue. When both call types have the same level of priority, the agent group will take the next call from its own queue.

- θ_i : threshold for giving priority: when the queue of type i reaches θ_i , agent group j will start serving customers of this queue by priority, even if there are type j calls waiting for these agents. This will only be the case when the number of waiting type j calls is lower than θ_j.
- Ω_i : threshold for overflow: when the queue of type i reaches Ω_i , calls of type i may flow to server j when there is no queue of call type j waiting to be served.

Applying this definition to scenario 3, yields $\mathbf{Thr}(\infty, \infty, 1,1)$ as threshold for the two-way overflow scenario and $\mathbf{Thr}(\infty, \infty, 1, \infty)$ for the one-way case.

Setting these thresholds right, can offer a substantial improvement over both the pooled and unpooled case. Choosing the wrong thresholds policy however, will lead to a performance worse than both of them. In scenario 4a, we will run a simulation model with the following threshold policy: **Thr**(1,3,1,1). We obtained these values by declaring them as variables in our simulation tools. Using *Process Analyzer*, one of the tools *Arena* offers, it is possible to quickly and easily simulate multiple combinations of the different variables. By means of trial-and-error, we discovered that this policy offers a sufficient improvement over the unpooled and pooled scenarios and even an improvement over the simple overflow cases. In the same way, we found **Thr**(5,2,5,3) to demonstrate the effect of a bad threshold policy. This will be used in scenario 4b. Using **Thr**(1,3,1,1), we will also build a case without abandonment, similar to the previous scenarios.

Ultimately, a variant is added where we will test the 80/20 rule of thumb. Instead of all nine agents being cross-trained, we will limit the amount to 20%, or two employees, in the best scoring scenario out of all the previous mentioned variants: the original scenario using **Thr**(1,3,1,1). Therefore, we are combining dedicated and flexible operators in this scenario.

4. Results

In this section different metrics will be given, visualized and explained for all scenarios, after which they will be compared to each other and the status quo. The following metrics will be used in our analysis:

Table 5: Metrics

Metric	Explanation
Average waiting time served "Regular/ Potential"	The average waiting time of customers split up according their call type. The metric will thus consist out of two subdivisions: - Average waiting time "Regular" - Average waiting time "Potential" The waiting time of customers which eventually abandon the queue will not be accounted for. Customers served immediately will be included with a waiting time of zero seconds.
Total average waiting time	The average waiting time of all customers served by an agent, independently of their call type. The same regulations concerning abandonment and immediately served customers as in the prior metric will hold.
Average waiting time of abandoning customers	The average amount of time abandoning customers have spent in a queue. Split up in subdivisions according call type.
Number of customers in queue	The number of customers which was not immediately served when leaving the VRU and thus had to spend time in a queue.
Utilization	The amount of time agents have been working in comparison with the total amount of time scheduled. The level of granularity in this metric is the measurement of the utilization rate of each agent separately.
Abandonment	Equivalent to the number of customers who ran out of patience and left the queue without being served.

4.1. Unpooled or status quo

The status quo is the scenario which all other variants should outdo. If we have a look at our waiting times in Figure 8, we notice that the total average waiting time as well as the average waiting time of both subdivisions is already rather acceptable. With an average waiting time of 12,95 seconds, most of the served customers will be satisfied. Keep in mind that this includes all customers with a waiting time of zero seconds, 939 customers in the unpooled case. The customers which pass the queue will thus have an average waiting time way higher than 13 seconds. This explains the problem which becomes clear when looking at Figure 9: more than a quarter of all incoming telephone calls will eventually leave the unpooled system without being served. Average waiting times of 92,45 and 96,36 seconds for respectively abandoning regular and potential customers illustrates the problem even better. For this reason, reducing abandonment becomes an important objective.

Figure 10 shows us the utilization rate of each server and reveals a clear difference between both agent groups. Bearing in mind the fact that the first and last hours of the day contain significantly fewer incoming calls and thus some time servers are idle, the agents serving regular customers are working almost non-stop during the majority of the day. On the other hand, agents serving potential customers are idle more than two-thirds of the time. Improving the difference between both agent groups and trying to reduce the workload on agent group "Regular", will be another additional goal and the subject of discussion in section 4.3.

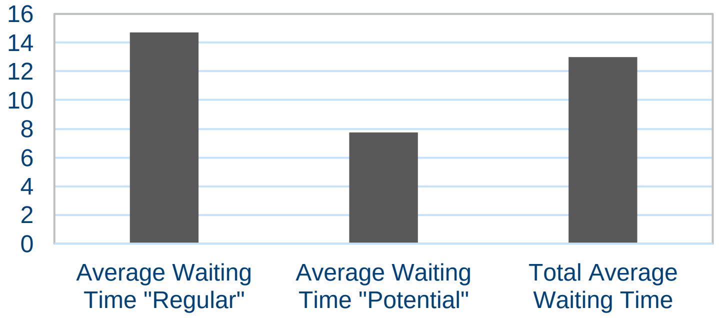

Figure 8: Unpooled waiting times

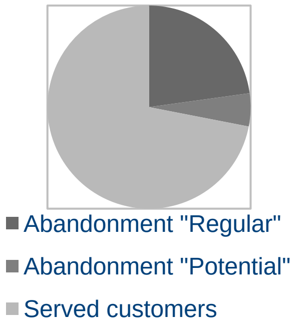

Figure 9: Unpooled abandonment

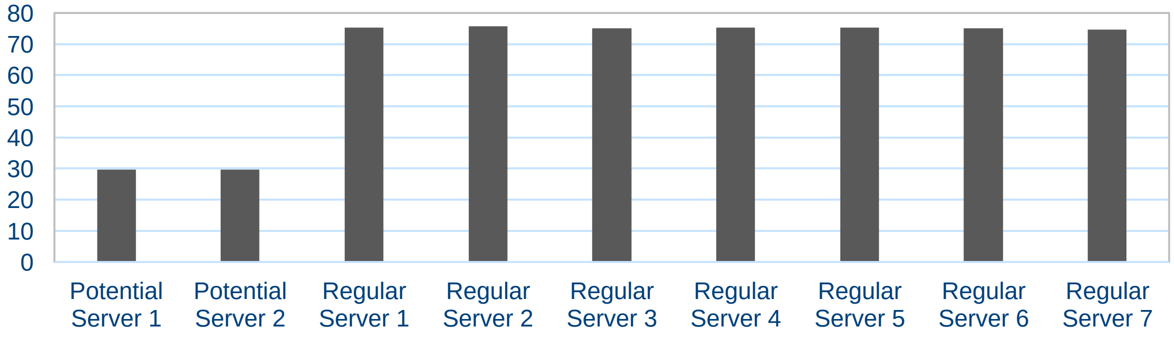

Figure 10: Unpooled utilization rate

4.2. Waiting times

4.2.1. Unpooled versus pooled

When looking for an improvement in waiting times, the most logic improvement for the status quo often is the pooled scenario. When evaluating the effect of pooling both call types on one resource pool, we will distinguish two different settings, as discussed in section 3.2.2. Figure 11 visualizes the waiting times when using the service time and patience level distributions derived from the dataset, while Figure 12 shows the same metrics obtained while using modified distributions. All graphs used in section 4.2 are expressed in seconds, unless stated otherwise.

In case of the original distributions, we observe a moderate improvement in waiting times for regular customers, while potential customers will have to wait a little longer before being helped. On average, this will lead to a reduction of almost three seconds of waiting time throughout the whole system. In terms of percentage, this is an improvement of 21% compared to the status quo. When looking at the simulation results with modified distributions, the large difference between the waiting time between both call types in the unpooled case, catches the eye. With service times of potential customers cut in half, their average waiting time becomes almost zero. Pooling both subsystems again yields a moderate improvement for regular customers, but significantly worsens the situation of prospective customers this time. With an increase of 41 to 390 potential customers forced to wait in the queue, it is no surprise that the appropriate average waiting time rises from almost non-existing to 94,69 seconds. This enormous escalation also leads to a gain in total average waiting time. The results of both cases illustrate the varying effect of pooling and confirms the rule of thumb used by van Dijk and van der Sluis (2008). From here on, the results of Figure 11 will be used as the default results for the pooled and unpooled scenario throughout our research.

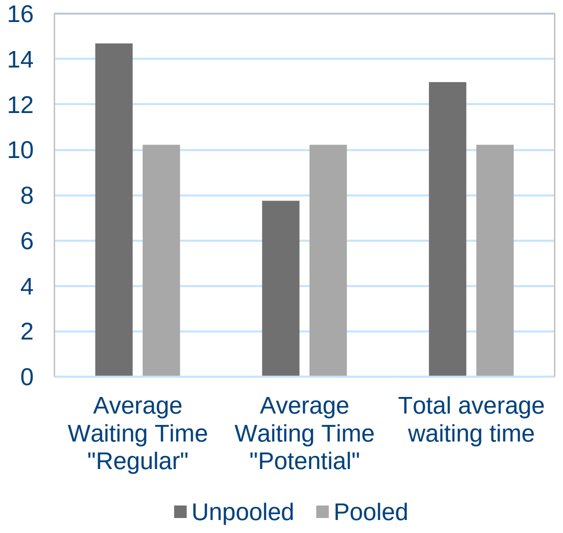

Figure 11: Original distributions

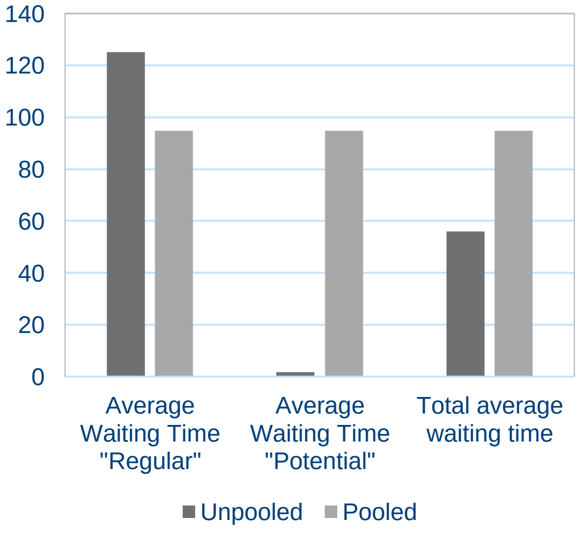

Figure 12: Modified distributions

4.2.2. Overflow

Starting with simple overflow without priorities, we will explore the outcomes of implementing both one-way and two-way overflow as stipulated in section 3.2.3. Figure 13 displays the relevant waiting times in comparison to the status quo. Looking at the weight of both call types and the utilization rate which will be analyzed further on, we opt for the logical one-way overflow of calls from the regular-queue to the agent group of potential customers. As one could expect, this drastically reduced the average waiting time of regular customers. The downside of this decrease of one-third for regular customers, is more than a doubling of the average waiting time of prospective customers. However, due to a favorable division of calls between both customer types, the one-way overflow situation does improve the total average waiting time in this example, despite the doubled waiting time for a quarter of the incoming telephone calls.

The two-way scenario offers the possibility to better spread the workload of the second agent group likewise. Even though the waiting time of potential customers increases here as well, it is an insignificant rise while the average waiting time of regular customers drops almost as low as with one-way overflow. This results in a total average waiting time lower than both the one-way scenario and the status quo, as shown in Figure 13.

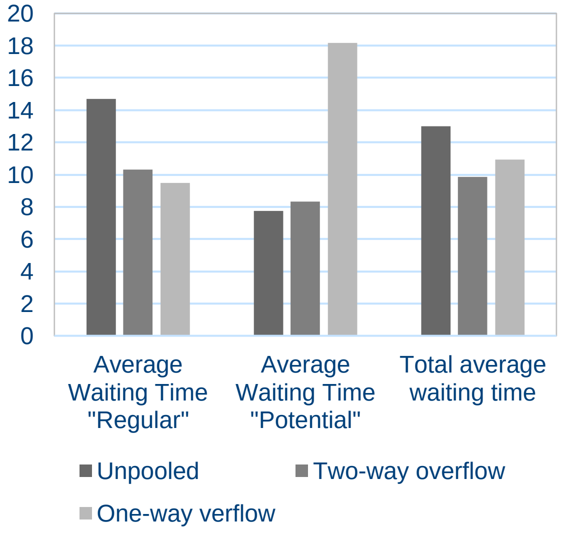

Figure 13: One-way vs. two-way overflow waiting times

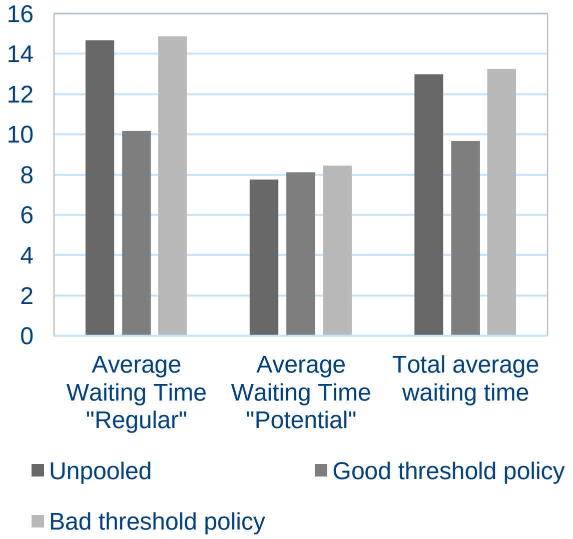

Figure 14: Effect of threshold policies

When implementing priority rules, picking the right threshold policy is essential in order to achieve good results. Simulation software offers the possibility to efficiently determine the effect of certain policies. To demonstrate the consequences this choice can have, Figure 14 visualizes the effect of a good and a bad policy choice, compared to the status quo. **Thr**(5,2,5,3) for example offers a deterioration of the waiting time of both customer types. It is self-evident that the total average waiting time is worse than the status quo as well. **Thr**(1,3,1,1) on the other side, offers a substantial improvement to the total average waiting time. When setting the best case of each scenario side by side with our status quo and comparing the resulting waiting times, we become Figure 15. It clearly displays the positive effect the different scenarios can have, compared to the unpooled case. Figure 15 tells us that, when pooling is not an option, overflow can achieve an improvement as well. Even when pooling is possible, a better result can still be achieved with simple overflow. When also taking into account priority rules, even the excellent results of a simple overflow scenario can be made better. In our case, the simple overflow case already was quite efficient, so the improvement is rather small. Nevertheless, the results prove that an amelioration of waiting times is possible when adding priority to the overflow scenario. Ultimately it is shown that waiting times of an unpooled case can be improved with 25% or more by making the right choices. These best case variants of each scenario, will also be used when comparing abandonment and utilization rates.

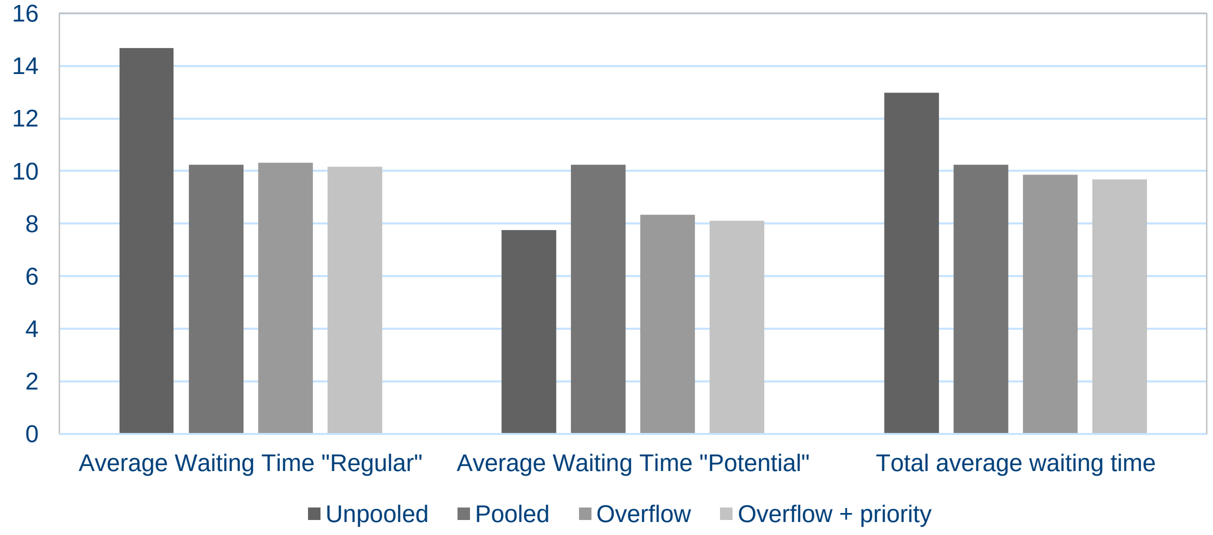

Figure 15: Waiting time overview

4.3. Abandonment

Rationally, reducing the time customers lose in the system will reduce the number of customers abandoning as well. Customers will be served faster and therefore spent less time waiting, causing fewer customers to run out of patience. In the unpooled scenario, 1369 out of the 1903 customers will be served. This leaves 28% of the customers who didn't got served because they abandoned the queue. Pooling our servers cuts down this portion of customers to 21,3%, that is a surplus of 129 customers receiving the help they demanded. When looking at the results of overflow with and without priorities, minor improvements can be seen compared to the pooled case. Using **Thr**(1,3,1,1) provides the best results, with an abandonment rate of 20,7%. Figure 6 visualizes the amount and composition of abandonment in all cases.

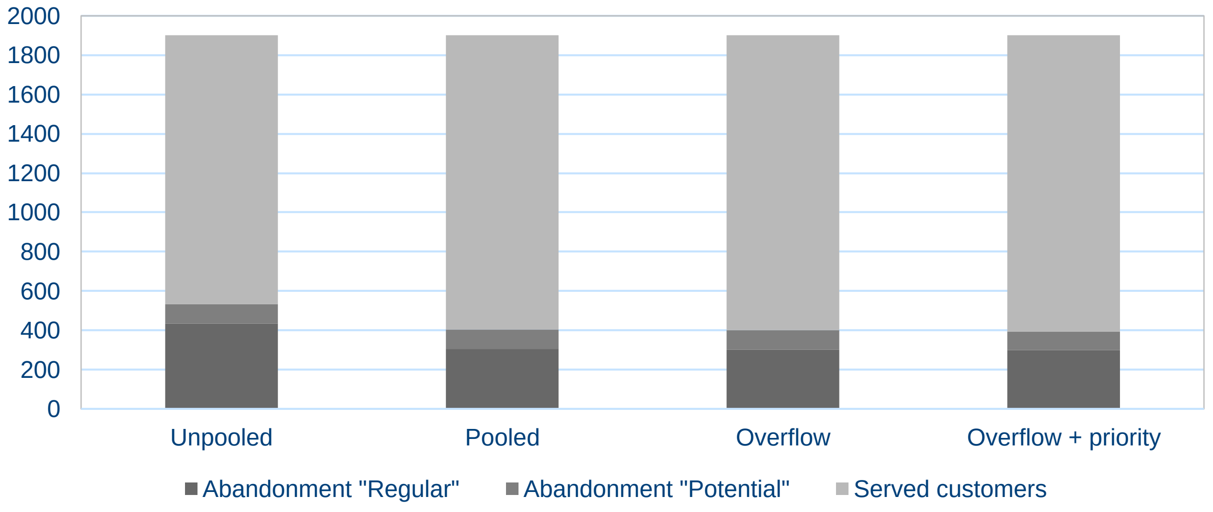

Figure 16: Abandonment overview

4.4. Utilization

When addressing utilization, we rationally expect the workload to be more equally spread in all scenarios when comparing to the unpooled case. This would mean an increase in the utilization rate of the two agents handling potential customers and a diminishing effect for all other agents. However, we must keep in mind the reduction in abandonment when using our simulation models. When working more efficiently, our first agent group will indeed receive more calls, but the other group won't see a significant decrease in calls. Instead, abandonment will decrease and their workload will remain approximately the same. Only the pooled scenario is an exception. The incoming calls will be divided uniformly among the agent pool. This is all illustrated by Figure 17.

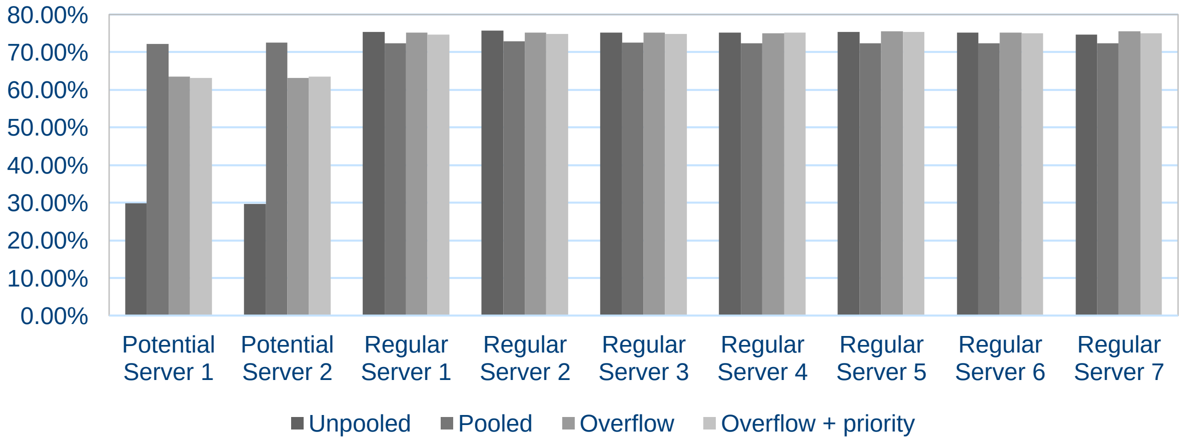

Figure 17: Utilization overview with abandonment included

In order to clearly illustrate the effect different scenarios have on utilization, we will build a variant of all scenarios where abandonment is left out. All customers will thus be served, no matter how long their waiting time has been. This will lead to enormous queues, which the student-version of *Arena* can't handle. To overcome this problem, the incoming calls will be limited to a couple of hours instead of an entire day. We will use the calls arriving between 7:00 AM and 11:00 AM January third, 1999. This results in 454 calls from which 376 calls or 82% are regular customers. The conforming utilization rates can be found in Figure 18. Although the shape of both figures looks the same at first sight, we now can clearly see that overflow, with and without priority, reduces the "Regular" agents' workload with approximately 5%.

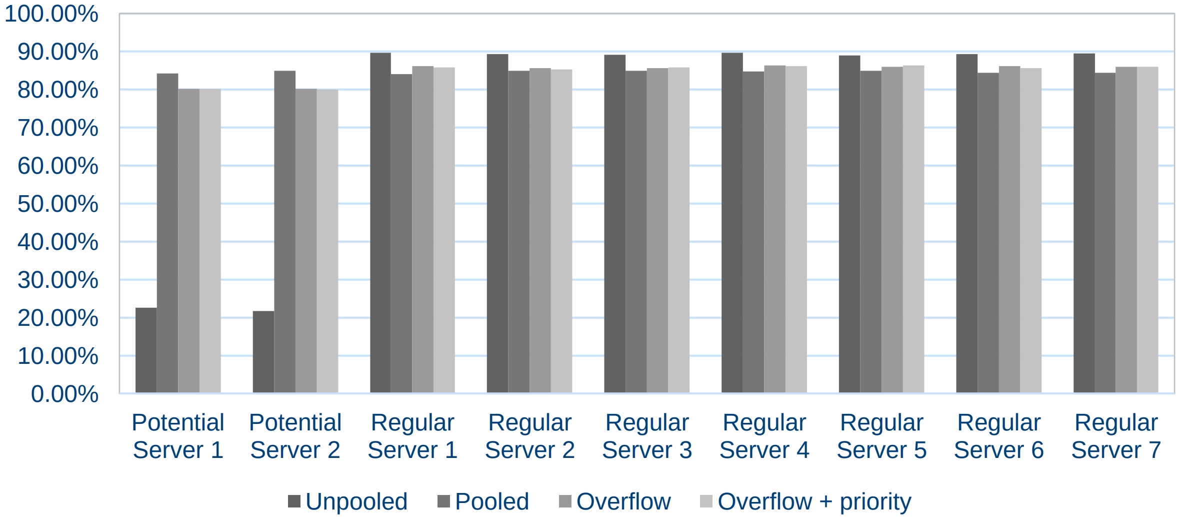

Figure 18: Utilization overview without abandonment

4.5. 20% Cross-trained agents

Besides limiting the amount of call types cross-trained agents should be able to handle, the literature review also made clear that limiting the proportion of cross-trained operators is important. Most research agree that cross-training 20% of the agents should approach the same results as cross-training the total workforce. When applying this rule to our most successful scenario, the combination of overflow with the right threshold policy, we obtain the average waiting times and abandonment as displayed in Figure 19 and 20 respectively. Cross-training two out of nine agents corresponds to 22% of the employees being cross-trained and is our best possible approximation of 20%. Although both results are still better than in the unpooled scenario, abandonment as well as the waiting times deteriorate significantly compared to the fully cross-trained case. Even when upscaling the percentage of cross-trained agents to four out of nine, i.e., 44% of the employees, the results are considerably worse.

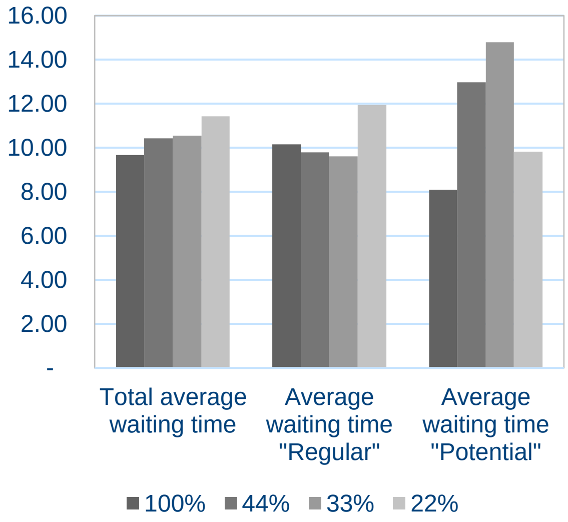

Figure 19: Cross-training rate & waiting times

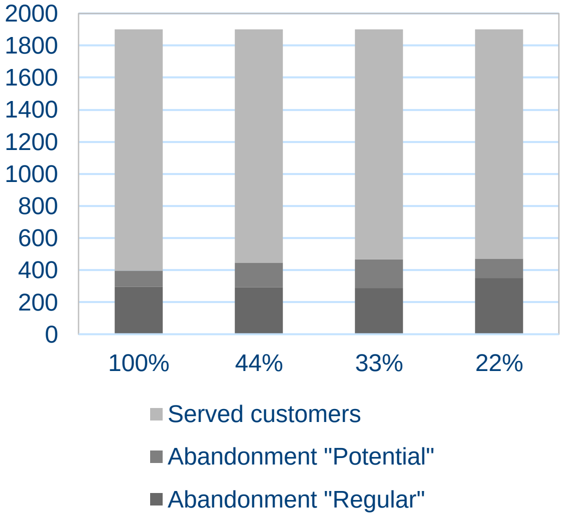

Figure 20: Cross-training rate & abandonment

5. Conclusions and insights

When starting with a system consisting out of separated queues for each type of customer problems, the general perception seems to exist that combining multiple queues would improve the efficiency. By pooling the appropriate demand on one merged agent group, call center managers assume the waiting times to decline and thus consequently a reduction in abandonment to take place. It is proved in other papers that this intuition is valid when the combined queues consist out of one call type. In this paper, we focused on the case when combining two queues which both handle different type of customer problems, i.e., a different set of skills is required by the fitting agent group. In order to investigate the consequences on waiting times and abandonment, we opted to construct an equivalent of a call center environment using the simulation software *Arena*. To keep the experiment as realistic as possible, the simulation models required data of a real call center. A free to use database was found online. It contained information about incoming telephone calls at the call center of a bank in Israel during 1999. The two most frequent call types where filtered out during a time span of 24 hours and served as input for our simulation models, resulting in 1903 incoming calls. Using data of an entire month, the necessary distributions where derived from the available data as well.

Using those acquired distributions, the results of the unpooled and pooled case have been compared. In this case, the intuitive feeling clearly became reality as both the waiting times and abandonment significantly improved. Despite a small rise in waiting time for the potential customers call type, the total average waiting time was 3 seconds less than in the unpooled case. That is equivalent to a reduction of 21%. Looking at abandonment, 129 less customers abandoned the queue when pooling the demand, reducing this fraction of customers to 405 instead of 534. The available literature warned us for the influence of the required service times when pooling different call types and provides a rule of thumb which states that as long as one of the service times is not larger than six times the other one, pooling would be beneficial. Both service times in our models where modified to not fulfill this rule of thumb. As expected, with almost a doubling of average total waiting time, the pooled case this time offers a less positive result. In conclusion, we can state that pooling demand in a service environment indeed can turn out to be beneficial concerning the total average waiting time and abandonment. Nevertheless, it is demonstrated that pooling can turn out to be negative as well and call center managers have to be aware of this possible result. The difference in required service times between the pooled call types proved to be an important metric when considering pooling. The rule of thumb Tekin et al. (2009) provided us with, turned out to be a convenient guideline: when mean service times differ considerably, pooling may lead to worse performance than the unpooled scenario. More specifically, if there exist two call types for which the average waiting time of one does not exceed approximately six times that of the other, pooling will be advantageous (Tekin et al., 2009).

Besides the unpooled and pooled scenario, two other variants where simulated and investigated: simple overflow and overflow combined with priority rules. When pooling demand is not an option due to certain practical circumstances, managers could still try to improve the performance by allowing calls to flow between both queues. When calls can only flow to the other agent group when these servers are idle, we are talking about simple overflow. When allowing overflow at certain threshold and giving priority to particular call types, we talk about combining overflow with priority rules. Trough simulation, it is shown that simple overflow can be as efficient as an advantageous pooled situation but the choice between one-way or two-way overflow has to be considered.

When adding priority rules and thresholds to the system, the challenge to become an even more productive system is to discover the right threshold policy. Using *Arena,* the effect of multiple policies on waiting times and abandonment could be examined. To demonstrate the importance of these thresholds, both a positive as well as a negative threshold’s consequences are discussed. It is proved that, even when simple overflow already is a significant improvement, a better result can become with the help of the right threshold policy. On the other side, it is also proved that the wrong choice of policy can result in a worse performance than the status quo. When applying the 80/20 rule of thumb concerning the cross-trained portion of agents, it turned out not to be true for our case study. Even with as much as 44% of the employees cross-trained, the results did not approach the optimal results when 100% was cross-trained. One of the possible explanations is the small scale of the call center in our case study. Dealing with a call center existing out of more than hundred agents, offers more margin and flexibility when limiting the amount of training.

Besides the average waiting times and abandonment, the utilization rates of call center employees are analyzed as well. Due to the unequal partition of call types in this case study, the agent group handling potential customers is idle environ 80% of the time in our status quo, while the other agent group is working 75% of the time during opening hours. Obviously, merging both agent groups into one in our pooled scenario, spreads the workload uniform across all agents. In the overflow cases, we would instinctively expect the idle agent group to become busier while reducing the workload on the busiest agents. Simulating both overflow scenarios without abandonment verifies our expectations. Be that as it may, abandonment is a factor that needs to be taken into consideration in reality. Looking at our overflow scenarios with abandonment, we conclude that when system performances improve, the busier agents do not become more idle. Instead, they will handle more incoming telephone calls, resulting in approximately the same utilization rate. In a system not handling 100% of the incoming calls, employing more agents thus seems to be the best way to reduce workload of the busiest operators.

In conclusion, we can state that the unpooled situation indeed can be improved in most cases, concerning average waiting times and abandonment. Pooling demand can be an efficient method of working but is not guaranteed to be an improvement. The difference in average service times is a valuable piece of information for managers when thinking over the consequences of pooling on their specific system. When pooling is practically not possible or managers want to ensure a minimum service variability in the unpooled case or a minimum idleness of the employees in the pooled case, the overflow of calls can offer a solution. Our simulation tool has proven that equally good results can be achieved with overflow and can even exceed them with the right threshold policy. Implementing the wrong threshold policy however, can lead to a worse performance than the status quo. Summarized, this simulation study has showed that the unpooled scenario can be ameliorated in various manners when the right choices are made. Amongst other things, decision makers have to be aware of the importance of the right threshold policy, the difference in average service times and the possible negative outcome when not thinking through their choices.

This paper could serve as a starting point for further research on multiple domains. A first extensions could be made by taking costs into consideration. Agents who are able to handle more than one call type, need training for all additional skills. Next to these training costs, highly skilled employees often expect a sufficient salary. All these extra costs could influence the manager’s choice when considering a change to the status quo and need to be balanced against the income surplus they will generate. A second extension could be the addition of a third or fourth call type. Although bi-skilled operators are thought of to be most efficient, it would be interesting to put this to the test using simulation. Determining which two call types to pool would result in interesting insights as well. Third, introducing shifts for employees could be an interesting extension. Dividing the opening hours into shifts and introducing a customized schedule based on the amount of call arriving during a shift, could lead to refreshing results and improvements. Finally, it would be interesting to find out why the 80/20 rule with regard to the ratio between dedicated and flexible operators does not apply to our case study. A first approach could be to magnify the scale of our call center.

Acknowledgement

The author would like to thank Ilan Guedj and Avi Mandelbaum to place their acquired and organized dataset at the disposal of the public at large. I also would like to express my gratitude towards my mentor Inneke van Nieuwenhuyse for her useful comments, helpful insights and guiding knowledge which significantly helped shaping and improving this paper.

Bibliography

Aksin, Armony, M., & Mehrotra, V. (2007). The Modern Call Center: A Multi-Disciplinary Perspective on Operations Management Research. *Production and Operations Management, 16*(6), 665-688. doi:10.1111/j.1937-5956.2007.tb00288.x

Aksin, & Karaesmen, F. (2002). Designing flexibility: Characterizing the value of cross-training practices.

Aksin, Karaesmen, F., & Lerza, E. (2006). A Review of Workforce Cross-Training in Call Centers From An Operations Management Perspective.

Aksin, Karaesmen, F., & Örmeci, L. (2005). ON THE INTERACTION BETWEEN RESOURCE FLEXIBILITY AND FLEXIBILITY STRUCTURES.

Chevalier, P., Shumsky, R., & Tabordon, N. (2004). Routing and staffing in large call centers with specialized and fully flexible servers.

David, A. M., & Bastian, N. D. (2016). ESTIMATING CROSS-TRAINING CALL CENTER CAPACITY THROUGH SIMULATION. *系统科学与系统工程学报：英文版, 25*(4), 448-468. doi:10.1007/s11518-015-5286-9

Erlang Calculator Retrieved from https://www.callcentrehelper.com/tools/erlang-calculator/

Gans, N., Koole, G., & Mandelbaum, A. (2003). Telephone Call Centers: Tutorial, Review, and Research Prospects. *Manufacturing & Service Operations Management, 5*(2), 79-141. doi:10.1287/msom.5.2.79.16071

Guedj Ilan, & Mandelbaum, A. (2000). *“Anonymous Bank” Call-Center Data*.

Hopp, W. J., Tekin, E., & Van Oyen, M. P. (2004). Benefits of Skill Chaining in Serial Production Lines with Cross-Trained Workers. *Management Science, 50*(1), 83-98. doi:10.1287/mnsc.1030.0166

Mehrotra, V., & Fama, J. (2004). *Call center simulation modeling: Methods, challenges, and opportunities* (Vol. 1).

Ord, D. (2016). What you need to know about the pooling principles in contact centers. Retrieved from www.omnitouchinternational.com

Robbins, T., Medeiros, D., & Harrison, T. (2010). Cross Training in Call Centers with Uncertain Arrivals and Global Service Level Agreements. *International Journal of Operations and Quantitative Management, 16*, 307 - 329.

Sisselman, M. E., & Whitt, W. (2007). Value-Based Routing and Preference-Based Routing in Customer Contact Centers. doi:10.7916/D8H70T92

Smith, D., & Whitt, W. (1981). Resource Sharing for Efficiency in Traffic Systems. *Bell System Technical Journal, 60*. doi:10.1002/j.1538-7305.1981.tb00221.x

Tekin, E., Hopp, W. J., & Van Oyen, M. P. (2009). Pooling strategies for call center agent cross-training. *IIE Transactions, 41*(6), 546-561. doi:10.1080/07408170802512586

van Dijk, N. M., & van der Sluis, E. (2008). To pool or not to pool in call centers. *Production and Operations Management, 17*(3), 296-305. doi:10.3401/poms.1080.0029

Wallace, R. B., & Whitt, W. (2005). A Staffing Algorithm for Call Centers with Skill-Based Routing. *Manufacturing & Service Operations Management, 7*(4), 276-294. doi:10.1287/msom.1050.0086