Call Center Mathematics

A scientific method for understanding and improving contact centers

Ger Koole

Version of 24th October 2003

Author: Ger Koole, Vrije Universiteit, Department of Mathematics, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands.
Tel. +31 20 4447755, email koole@cs.vu.nl, internet www.cs.vu.nl/~koole.
The latest version of this book can be downloaded from www.cs.vu.nl/~koole/ccmath.

Preface

This book is written for everybody who is dedicated to improving call center performance. It offers a scientific method to understanding and improving call centers. It explains all generic aspects of call and contact centers, from the basic Erlang formula to advanced topics such as skill-based routing and multi-channel environments. It does this without using complicated mathematical formulae, but by stressing the meaning of the mathematics. Moreover, there is a companion web page where all calculations can be executed. Next to understanding call center phenomena we show how to use this insight to improve call center performance in a systematic way. Keywords are data collection, scenario analysis, and decision support.

This book is also a bridge between call center management and those parts of mathematics that are useful for call centers. It shows the manager and consultant the benefits of mathematics, without having to go into the details of the mathematics. It also shows the mathematically educated reader an interesting application area of queueing theory and other fields of mathematics. As such, this book can also be used in an applied course for mathematics and industrial engineering students. Basic knowledge of call centers is assumed, although a glossary is added in case of omissions.

Ger Koole
Amstelveen/Sophia Antipolis, 2001–2003

Contents

Chapter 1

About this book

This book can be used on its own, but to get the most out of it it is advised to use the companion web site, www.cs.vu.nl/~koole/ccmath. This site contains updates of the book, a list of typos, and, most importantly, the pages in which you can do the calculations that illustrate the text.

1.1 No maths

This book is meant for call center managers without any knowledge of mathematics. This does not mean that they cannot use maths to improve their call center performance. Even without knowing the maths everybody can understand the Erlang formula. (And, vice versa, knowing the Erlang formula, does not mean that you understand it!) This book is directed towards understanding the quantitative aspects of call centers. Current computer-based systems allow us to gain this understanding without knowing the maths themselves. Only engineers that implement the math in our computer systems should know and understand the formulas. This is the second group for which this book might be helpful: engineers and mathematicians that know the mathematics, who want to be introduced to the field of call centers. For the mathematically interested readers some formulae are supplied in the appendices, where also an extensive list of references can be found to assist further study.

1.2 Why a web site?

Easy to update, everybody has access to internet.

1.3 Feedback

Current virtual books, small editions, constant updating.

Mail to koole@cs.vu.nl.

1.4 How to use this book

Section with "*" in the title contain side topics and need not be read for understanding the main text.

1.5 Overview

Here we give a short overview of what can be expected in the remaining chapters of this book. The first couple of chapters deal with the basic call center: a single-channel, only inbound traffic, no IVR, etc. For this basic call center we discuss first the standard models and tools. Then we move to more complex models that deal in-depth with the problems of managing even these relatively simple call centers. Then we move to the issues related to multiple skills and channels.

1.6 Acknowledgments

I would like to thank several people for their input and corrections: Theo Peek, Arnout Wattel, ... My thanks also go to the Mistral project at INRIA Sophia Antipolis for their hospitality during my visits in the last years that allowed my to write the bigger part of this book.

Chapter 2

What is Call Center Mathematics?

In this chapter we explain how call centers can benefit from a mathematical approach.

2.1 The subject of Call Center Mathematics

To manage call centers, or more generally, contact centers, effectively, one needs to have multiple skills. Roughly speaking there are those skills which are unique to the product that is delivered, and there are those skills that are needed in virtually any call center. Some of these latter skills are soft, such as training and motivating people. Other skills are of a more quantitative nature, and are related to service level and an efficient use of the human resources. Mathematics can play an important role in getting the best out of the service level/cost trade-off. In a simple single-skill call center we see that the Erlang formula is used to determine the occupation level at any time of day. Scheduling algorithms are then used to determine shifts and to assign employees to shifts. In more complex environments mathematics is used to route calls, to decide how and when call blending is done, etc. Mathematics are an essential part of call center management.

2.2 Why should a call center manager know about mathematics?

Let us first put you at ease by stating that we do not think that managers in call centers should know the mathematics themselves. We do think that managers should know about the implications of mathematical theory for call centers. But if the mathematics are already implemented in the software, why know anything about it? Why should we understand the Erlang formula if it is readily available in many decision support systems? The answer to this lies in the name decision *support* system. No computer-based system can completely automize the complex scheduling and planning tasks in a call center. Human interaction is always needed, and this is only possible if the user *understands* the software. In this way, learning about call center mathematics increases the effectiveness of the available software.

On the other hand, certain tasks within a call center, such as call routing, are completely automized. But here the crucial decision was taken at the moment the routing algorithm was implemented. Again, only an understanding of the dynamics of call centers can help us to implement the right routing machanisms. Thus again, an understanding of call center mathematics will help us make better decisions.

A better understanding will also improve the communication with other people, not in the least with the consultant who is trying to sell a model-based solution.

The first two parts of this book are directed towards a better understanding of call centers.

2.3 A scientific method to call center improvement

Any business change raises questions about the effectiveness of the proposed change. Will it really work out the way it is foreseen? Given our understanding of call centers we often have an idea what the type of effect will be of certain changes. Direct implementation of the proposed changes, *on-line experimentation*, has the advantage of simplicity and low costs. But these costs remain only low if the effects of the changes are positive! For this reason one often likes to experiment first in a "laboratory" setting. Mathematics offers such a "virtual laboratory". The important aspects of reality are described in a mathematical *model* and this model can be analyzed using mathematical techniques. This way different scenarios can be analyzed, hence the term *scenario analysis*. But mathematics can do more. It can generate solutions for you. This is what a workforce management tool does when it generates an agent schedule. This solution can be of varying quality, depending on the model that is implemented. In theory mathematics can generate solutions that are better than those that are thought of by a human, and in much less time.

Merging two call centers leads to economies-of-scale advantages. However, the physical costs of such an operation can be high! Calculations based on the Erlang model can quantify the expected cost reduction. This way a reasonably accurate cost trade-off can be made.

To be able to experiment with your call center in the mathematical laboratory certain issues have to be solved first:
- You have to know exactly how your call center operates, i.e., you have to be able to describe the relations between the entities in your model;
- You have to know the current and possible future input values.

Based on our understanding of call centers we present, in part III, the details of the steps that we have to go through when improving call center processes: model construction, data collection and analysis, running scenarios, implementation.

Often however it is not necessary to go through the full modeling cycle, certainly not if we have a good feeling for the consequences of changes, i.e., if we have a good understanding of call centers. This understanding is thus a central issue in call center management. Parts I and II are consecrated to the understanding of call centers.

2.4 What to expect from call center mathematics

Mathematics can help you manage your call center. However, you should not expect miracles. Every modeling exercise implies simplifying the real situation first to fit it in the framework of the model. With this modeling step certain approximations are introduced, requiring a careful use of the outcomes. Modeling everything simply isn't possible, because of time constraints and because for example human behaviour cannot be modeled in all details. This doesn't make modeling useless, but it requires an attitude in which outcomes of modeling studies are tested thoroughly before being implemented. In Part III this is discussed in detail.

Chapter 3

On call center management and its goals

In this chapter we discuss the overall goals of call center management. Starting from these overall goals, that hold for longer time periods, we formulate objectives for short periods. In the next chapters we make, in all detail, the translation back from short to long time intervals. We also discuss what types of decisions can be taken to fulfill these goals.

3.1 Cost versus service

A call center offers a product, delivered through telephone calls with clients. Service level can be defined as the degree of satisfaction of callers with the offered service. This service level consists of many different aspects, related to the quality of the answer, the waiting time of the customer, etc. Some of these are hard to quantify, such as friendliness of the agent, others are easily quantified.

A help desk tries to answer 90% of all question correctly during the first call. Next to that they require that 80% of the calls is answered within 20 seconds waiting, and that no more than 3% of the calls abandons before getting a representative.

The manager of a call center tries to satisfy the service levels set by higher management, given its budget, and other constraints such as the number of work places (often called seats), the ICT infrastructure, and the available workforce. Of course, the higher the budget, the higher the service level can be, due to better training and more available resources. The main resource is the call center agent or representative, although communication costs can also be high, certainly for toll-free services. The cost-service level trade-off has a central place in quantitative call center management. The advantage of a budget is that there is no discussion possible about its interpretation. Defining the required service level is more complicated. We discuss it in Section 3.3. In the next section we discuss the various types of decisions that influence the performance of a call center.

In certain situations the profit of each individual call can be measured in terms of money. In

such a situation the average profit per handled call can be calculated, and instead of balancing cost and service level, we just maximize profit. We will pay attention to this business model in Chapter 7.

3.2 A classification of management decisions

In this book we focus on decisions taken by call center managers, planners, and shift leaders. However, decisions relevant to call centers are also taken by other people and by software. In this section we discuss all relevant types of decisions.

Strategic decisions Strategic decisions are made by upper management. They concern the role of the contact center in the company, the type of service that is to be delivered, etc. It imposes the framework in which the call center management has to work. Upper management also decides on the budget that is available to the call center.

Tactical decisions Tactical decisions are typically taken by the call center management, They concern how the resources are to be used. These resources consist of the budget, the existing ICT equipment, and the (knowledge of) the people working in the call center.

Decisions about structure (e.g., skill-based routing) and organization are taken at this level, as well as decisions about the hiring and training of agents.

Planning decisions At the operational level we can still distinguish between the time-horizon in which decision take effect, ranging from weeks to milliseconds. Usually on a weekly basis new agent schedules are make by a planner at the call center. This is called workforce management.

Daily control Every day decisions have to be taken to react to the current situation in the call center. Usually shift leaders monitor service levels and productivity and can react to that.

Real-time control Finally, certain decisions are taken real-time by software, usually the ACD. This concerns for example decisions about the assignment of calls to available agents. Sometimes these decisions involve complex algorithms, for example in the case of skill-based routing.

3.3 Service level

We saw that the goal of call center management is to obtain the right cost-service level trade-off. We also saw by who and by what types of decisions cost and service level can be influenced. We now go into more detail how this service level is defined exactly.

The service level (SL) obtained by a call consists of several different aspects. Several are related to the handling of the calls themselves, such as the way in which the agents attend to the call, and the ratio of calls that need no need further calls, the first-time-fixed ratio. Others are related to the waiting process, notably the waiting times and the occurrence of abandonments. We focus on waiting times and abandonments, although other aspects of the service level can have a large impact on the waiting time and therefore also on the abandonments.

The help desk of an Internet Service Provider had a considerable rate of callers that phoned back after their call because the answer was not sufficiently clear to solve their problems. By improving scripts and documentation and by additional training this rate was reduced considerable. This not only improved the perceived service level, it also reduced the number of calls. This had a positive effect on the waiting times, and thus again on the service level.

The common way to define service level is by looking at the fraction of calls that exceeds a certain waiting time, which we will call the "acceptable waiting time" (AWT). The "industry standard" is that 80% of all calls should be answered in 20 seconds, but other numbers are possible as well. The service level can be calculated for all types of time intervals, from the very short (minutes) to years. Service levels of longer periods can be calculated be averaging in the right way service levels over shorter periods. When averaging over a number of intervals the number of calls in these intervals should be taken into account. Consider the table below. At first sight the average service level is 75%, by averaging the four percentages, but now the differences in calls per week are not taken into account. The right way of calculating is to compute the *fraction* of calls in each interval first. For example, the fraction of calls in the first interval is $\frac{2000}{17000}$, 17000 being the total number of calls over the four weeks. Using these fractions a *weighted average* is calculated in the following way:

$$\frac{2000}{17000} \times 95 + \frac{7000}{17000} \times 55 + \frac{5000}{17000} \times 70 + \frac{3000}{17000} \times 80\% = 68.5\%.$$

This way of calculating averages corresponds to the answer in case the service level was computed directly for the whole month. Indeed, out of a total of 17000 calls 11650 were answered in time, thus a $\frac{11650}{17000} \times 100 = 68.5\%$ service level.

Week	Number of calls	Answered within 20 s.	SL
1	2000	1900	95%
2	7000	3850	55%
3	5000	3500	70%
4	3000	2400	80%

The difference between 68.5 and 75% is not that dramatic. This is because the number of calls in the different weeks are roughly of the same order of magnitude. If the number of calls in the intervals over which we average are very different, then the way of averaging can have an even bigger impact on the result. These big fluctuations typically occur during days. At peak hours we can easily have ten or twenty times as many calls per hour as

during the night. Then the difference between ways of averaging can run into the tens of percents.

The percentage of calls that is answered in less than a certain fixed waiting time is sometimes called the *telephone service factor* (TSF). Another commonly used waiting time metric is the *average speed of answer* (ASA).

A phenomenon that occurs in every call center is that callers *abandon* (or *renege*) while waiting in the queue. In general, this is considered to be something to avoid, although some callers abandon in less than the AWT. One way to deal with abandonments is by setting a separate service level constraint on abandonments, e.g., on average not more than 3% abandonments.

If the TSF is used, then there is also the possibility to integrate the abandonments in this way of choosing the SL. For this, we first have to decide how to count abandonments. It is clear that callers who abandon after the AWT have received bad service, and therefore these calls are added to the number of calls for which the service requirement was not met. For callers that abandon before the AWT there are different possibilities. The most reasonable is perhaps not to count these calls at all. This leads to the following definition of service level:

$$\text{SL} = \frac{\text{Number of calls answered before AWT}}{\text{Number of calls answered} + \text{Number of calls abandoned after AWT}} \times 100\%.$$

Another possibility is to count them as calls for which the SL was met.

A call center receives 510 calls during an hour. The AWT is set equal to 20 seconds. A total of 460 receive service, of which 410 are answered before 20 seconds. Of the 50 abandoned calls 10 abandon before 20 seconds. Therefore the service level is $\frac{410}{460+40} \times 100\% = 82\%$*. Not taking abandonments into account when computing the SL would lead to a SL of* $\frac{410}{460} \times 100\% = 89\%$*!*

Incorporating abandonments in the ASA cannot be done in a simple way.

Service levels can be measured in two different scales: between 0 and 100 or between 0 and 1. We will use both. To go from one scale to the other we simply have to divide or multiply by 100. Mathematicians often prefer to measure between 0 and 1, because the results can be interpreted as fractions or probabilities. Although it will be clear usually, we will always use the "%" sign when using the percentage scale.

3.4 A discussion of service level metrics

When such a complex phenomenon as service level is reduced to a few numbers, then it is unavoidable that certain details are "averaged out". As an example, take the waiting times of just 4 calls: 0, 10, 30, and 100 seconds. When the service level is calculated as the fraction of calls that have a waiting time exceeding 20 seconds, than the service level is 50%. The same would hold if the waiting times were 0, 10, 24, and 30 seconds, although there is a clear difference between the situations! This difference shows up if we vary the AWT.

The difference between the two sequences directly shows up in the ASA, which is 35 seconds in the first case and 16 in the second. However, the waiting time sequences 0, 10, 30, 100 and 35, 35, 35, 35 would give the same ASA, showing that the ASA, by its proper definition, does not depend on the variability: is the ASA caused by many calls having a short waiting time or by a few calls having a very long waiting time? Both is possible!

A compromise between the two metrics is the expected waiting time in excess of the AWT, called the *average excess time* (AET). For the 0, 10, 30, 100 sequence the waiting times in excess of 20 seconds are 0, 0, 10, and 80, giving 22.5 seconds as AET. For 0, 10, 24, 30 it gives 3.5, and for 35, 35, 35, 35 the AET is equal to 15.

Chapter 4

The Erlang C formula

In the last chapter we saw that service levels, even for longer periods, could be derived from the service levels over shorter intervals. In this chapter we study call center performance over intervals that are short enough to assume that the characteristics do not change. The basic model for this situation is the Erlang model. This is what we study in this chapter in all detail.

4.1 The Erlang formula

In this section we introduce the famous Erlang C or Erlang delay formula, named after the Danish mathematician who derived the formula at the beginning of the 20th century. We have a call center with only one type of calls and no abandonments, thus every caller waits until he or she reaches an agent. The number of calls that enter on average per time unit (e.g., per minute) is denoted with the Greek letter λ. The average service time of calls or average holding time is denoted with β, measured in the same unit of time. We define the load a as $a = \lambda \times \beta$. The unit of load is called the *Erlang*.

Consider a call center with on average 1 call per minute, thus $\lambda = 1$, and a service time duration of 5 minutes on average, thus $\beta = 5$. The load is $a = \lambda \times \beta = 1 \times 5 = 5$ Erlang. Note that it does not matter in which time unit λ and β are measured, as long as they are the same: e.g., if we measure in hours, then we get again $a = \lambda \times \beta = 60 \times \frac{1}{12} = 5$ Erlang.

The offered traffic is dealt with by a group of s agents. We assume that the number of agents is higher than the load (thus $s > a$). Otherwise there are, on average, more arrivals than departures per time unit, and thus the number of waiting calls increases all the time, resulting in a TSF of 0%. (In reality this won't occur, as callers will abandon.) We can thus consider the difference between s and a as the overcapacity of the system. This overcapacity assures that variations in the offered load can be absorbed. These variations are not due to changes of λ or β, they originate in the intrinsic random behavior of call interarrival and call holding times. Remember that λ and β are averages: it occurs during short periods of time that there are so many arrivals or that service times are so long that undercapacity occurs. The strength of the Erlang formula is the capability to quantify the

TSF (and other waiting time measures) in this random environment with short periods of undercapacity and therefore queueing.

The Erlang C formula gives the TSF for given λ, β, s, and AWT. For the mathematically interested reader we give the exact formula, for $a < s$:

$$\text{TSF} = 1 - C(s,a)e^{-(s-a)\frac{\text{AWT}}{\beta}}.$$

Here e is a mathematical constant, approximately equal to 2.7; $C(s,a)$ is the probability that an arbitrary caller finds all agents occupied, the *probability of delay*. In case $a \geq s$ then TSF $= 0$. The formula itself is useful for those who implement it; for a call center manager it is more important to *understand* it, i.e., to have a feeling for the TSF as variables vary. For this reason we plotted the Erlang formula for some typical values in Figure 4.1. We fixed β, s, and AWT, and varied λ. In the figure we plotted λ on the horizontal axis, and the TSF on the vertical. The numbers in the figure can be verified using our Erlang calculator.

With the numbers of the example above, $\lambda = 1$ and $\beta = 5$, we got a load of 5 Erlang. Let us schedule 6 agents, and assume that a waiting time of 20 seconds is considered acceptable, i.e., AWT $= 20$ seconds. Filling in 1 and 5 and selecting "Number of agents" (20 is already filled in at start-up) gives after computation the TSF under "Service level". It is almost 72% (check this!). Increasing the number of agents to 8 already gives a TSF of 86%.

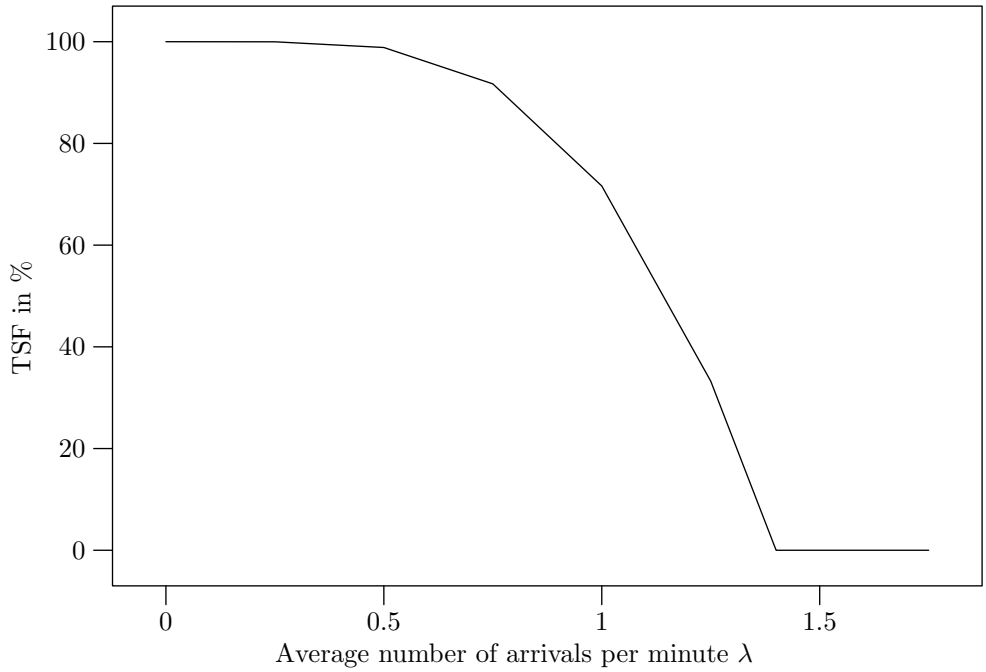

Figure 4.1: The TSF for $\beta = 5$, $s = 7$, AWT $= 0.33$, and varying λ.

We follow the curve of Figure 4.1 for increasing λ. Starting at 100%, the TSF remains close to this upper level until relatively high values of λ. As λ gets such that $a = \lambda \times \beta$

approaches s then the TSF starts to decrease more steeply until it reaches 0 at $\lambda = s/\beta = 7/5 = 1.4$. From that point on, as explained earlier, the TSF, as predicted by the Erlang formula, remains 0%.

Next to the SL in terms of the fraction of calls waiting longer than the AWT, the TSF, we can also derive the average speed of answer (ASA), the average amount of time that a caller spends waiting. The overcapacity assures that the average speed of answer remains limited. How they depend on each other is given by the Erlang formula for the ASA. This formula is given by:

$$\text{ASA} = \frac{\text{Probability of delay} \times \text{Av. service time}}{\text{Overcapacity}} = \frac{C(s,a) \times \beta}{s - a}.$$

For the same input parameters as in Figure 4.1 we plotted the ASA in Figure 4.2. We see clearly that as λ approaches the value of $s/\beta = 1.4$ then the waiting time increases dramatically.

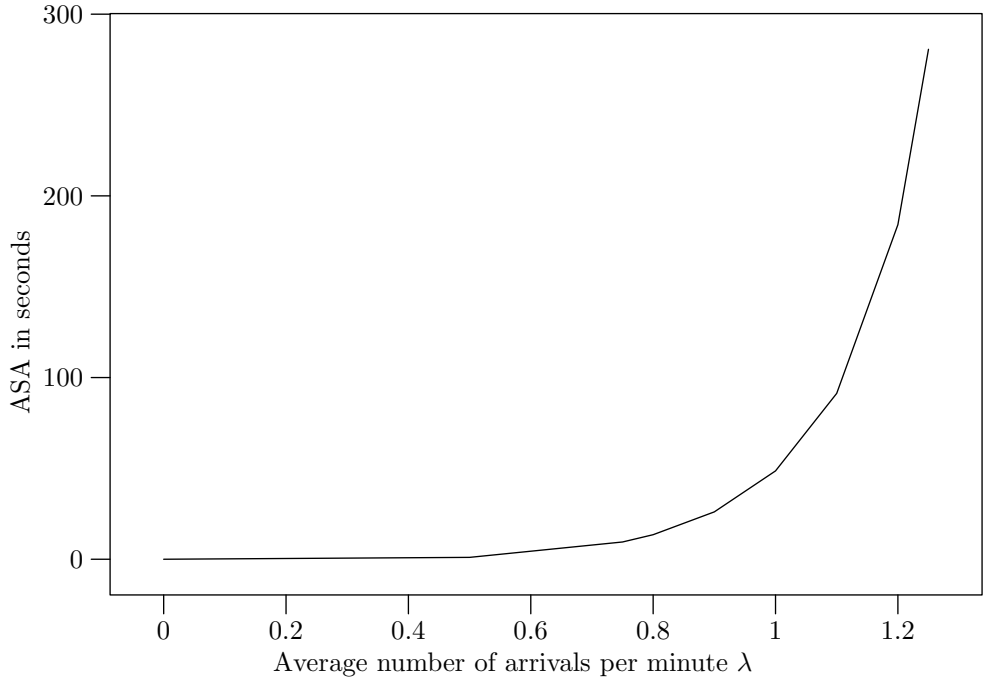

Figure 4.2: Values of the ASA for $\beta = 5$, $s = 7$, AWT $= 0.33$, and varying λ.

The probability of delay is not only an intermediate step in calculating TSF or ASA, it is also of independent interest: it tells us how many callers are put in the queue and how many find a free agent right away. The probability of delay can also be computed using an Erlang calculator, by filling in AWT $= 0$, and by noting that $100 \times$ probability of delay $=$ $100 -$ TSF.

Now we continue the example. We already saw that the load is 5 Erlang. Let us place 7 agents, then there is 2 Erlang overcapacity. Calculating the probability of delay $C(s,a)$ gives $C(7,5) \approx$

0.32. *Now we can fill in the formula for the average waiting time:*

$$ASA = \frac{C(s,a) \times \beta}{s-a} \approx \frac{0.32 \times 300}{2} = 48 \text{ seconds.}$$

This corresponds with the answers of Erlang calculator. Taking 8 agenten gives

$$ASA = \frac{C(s,a) \times \beta}{s-a} \approx \frac{0.17 \times 300}{3} = 17 \text{ seconds.}$$

Thus increasing the number of agents with 1 reduces the average waiting time with a factor 3.

Up to now we just discussed the service level aspects of the Erlang C system. Luckily, the agent side is relatively simple. Let us consider the case that $a < s$, thus $s - a$ is the overcapacity. Because every caller reaches an agent at some point in time, the whole offered load a is split between the s agents. This gives a productivity of $a/s \times 100\%$ to each one of them, if we assume that the load is equally distributed over the agents. If $a \geq s$ then saturation occurs, and agents get a call the moment they become available. In theory, this means a 100% productivity. In practice such a high productivity can only be maintained over short periods of time.

4.2 Using the Erlang formula

In the previous section we saw that the Erlang formula can be used to compute the average waiting time for a given number of agents, service times and traffic intensity. One would like to use the formula also for other types of questions, such as: for given β and s, and a maximal acceptable ASA or given SL, what is the maximal call volume per time unit λ that the call center can handle? Because of the complexity of $C(s,a)$ we cannot "reverse" the formula, but by trial-and-error we can answer these types of questions.

The question that is of course posed most often is to calculate the minimum number of agents needed for a given load and service level. This also can be done using trial-and-error, and software tools such as our Erlang calculator often do this automatically.

In our Erlang C calculator, fill in 1 and 5 at "Arrivals" and "Service time", fill in "80" and "20" at "Service level" and select "Service level" instead of "Number of agents". Computation shows that 8 agents are needed to reach this SL.

Most software tools will give you an integer number of agents as answer. This makes sense, as we cannot employ say half an agent. However, we can employ an agent half of the time. Thus when a software tool requires you to schedule 17.4 agents during a half an hour, then you should schedule 17 agents during 18 minutes, and 18 agents during 12 minutes. With 17 agents you are below the SL, with 18 you are above. Thus the "bad" SL during 18 minutes is compensated by the better than required SL due to using 18 agents. In our Erlang C calculator we decided not to implement this, because we assume that the time interval is so short that a constant number of agents is required.

Let us continue the example. Selecting "Number of agents" instead of "Service level" shows after computation that the actual service level is 86% instead of only 80% that was required.

"Garbage in = garbage out". This well-known phrase holds also for the Erlang formula: the input parameters should be determined with care. Especially with the value of the expected call durations β one can easily make mistakes. The reason for this is that the entire time the agent is not available for taking a new call should be counted. For the Erlang formula the service starts the moment the ACD assigns a call to an agent, and ends when the agents becomes available, i.e., if the telephone switch has again the possibility to assign a call to that agent. Thus β consists not only of the actual call duration, but also of the reaction time (that can be as long as 10 seconds!), plus the wrap-up time (that can be as long as the call itself). Note that the reaction time is seen by the caller as waiting time. This should be taken into account when calculating the service levels, by decreasing the acceptable waiting time with the average reaction time.

In a call center the reaction time is 3 seconds on average, the average call duration is 25 seconds and there is no finish time. On peak hours on average 200 calls per 15 minutes arrive. An average waiting time of 10 seconds is seen as an acceptable service level. We calculate first the load without reaction time. The number of calls per second is $200/(15 \times 60) \approx 0.2222$ *(*$\approx$ *means "approximately"), and the load is* $0.2222 \times 25 \approx 5.555$*. The Erlang formula shows that we need 7 agents, giving an expected waiting time of 8.2 seconds. This seems alright, but in reality there is an expected waiting time of no less than* 27.9 *seconds! This follows from the Erlang formula, with a service time of* $25+3 = 28$ *seconds (and thus a load of* $0.2222 \times 28 \approx 6.222$*), and 7 agents. The waiting time is then* 24.9 *seconds, to which the 3 seconds reaction time should be added. To calculate the right number of agents we start with a service time of 28 seconds, and we look for the number of agents needed to get a maximal waiting time of* $10 - 3 = 7$ *seconds. This is the case for 8 agents, with an average waiting time of 6.5 seconds. This way the average waiting time remains limited to 9.5 seconds.*

A possible conclusion of the last example could be that agents should be stimulated to react faster in order to avoid that an extra agent should be scheduled. However, these types of measures, aimed at improving the *quantitative* aspects of the call center, can lead to a decrease of the *quality* of the call center work, due to the increased work pressure. We will not deal with the human aspects of call center work; let it just be noted that 100% productivity is in no situation possible, and the overcapacity calculated by the Erlang formula is one of the means for the agents to get the necessary short breaks between calls.

4.3 Properties of the Erlang formula

Knowing the Erlang formula is one thing, *understanding it* is another. The Erlang formula has a number of properties with important managerial consequences. These we will discuss in this section.

Robustness One agent more or less can make a big difference in SL, even for big call centers. This is good news for call centers with a moderate SL: with a relatively limited effort the SL can be increased to an acceptable level. On the other hand it means that

a somewhat higher load, necessitating an additional agent, can deteriorate the SL considerably. In general we can say that the Erlang formula is very sensitive to small changes in the input parameters, which are λ, β en s. This is especially the case if a is close to s, as we can see in Figures 4.1 and 4.2. The figures get steeper when λ approaches s/β, and thus small changes in the value of the horizontal axis give big changes at the vertical axis. This sensitivity can make the task of a call center manager a very hard one: small unpredictable changes in arrival rate or unanticipated absence of a few agents can ruin the SL. In Chapter 6 we discuss in detail the consequences of this sensitivity.

In our small call center with $\lambda = 1$, β and $s = 8$ we expect an ASA of around 17 seconds. However, there are 10% more arrivals (i.e., $\lambda = 1.1$). The ASA almost doubles to over 30 seconds!

Stretching time A second property is related to the absolute and relative values of the call characteristics, i.e., β and λ. Recall that the load is defined by $a = \beta \times \lambda$. If either λ or β is doubled, and the other is divided by two, then the load remains the same. This does not mean that the same number of agents is needed to obtain a certain service level.

A manager is working in a call center that merely connects calls, thus call durations are short. As a rule she uses a load to agent ratio of 80%. From experience with the call center she knows that this gives a reasonable service level. For parameters equal to $\beta = 32$ seconds and 15 calls per minute the load is $a = 8$ Erlang. Indeed, with 10 agents the average speed of answer is 6.5 seconds. After a promotion she is responsible for a telephone help desk with also a load of 8 Erlang, but with β approximately 5 minutes, more than nine times as much. She uses the same rule of thumb, to find out that the average waiting is now around 60 seconds!

When λ is multiplied by the same number (bigger than 1) as β is divided with, then the load remains the same but it is like the system goes slower. Evidently, the waiting time also increases. If AWT is multiplied by the same number then the TSF remains the same. The relationship between the ASA and stretching time is more complicated.

It is like saying that the load is insensitive to the "stretching" of time. Certain performance measures depend only on s and a, but not on the separate values of λ and β. The probability of delay, $C(s, a)$, is a good example. It does not hold anymore for the TSF, here the actual value of β and λ play an important role. In fact, for given a and s, the service level depends only on AWT$/\beta$. Thus if time is stretched, and the acceptable waiting time is stretched with it, then the TSF remains the same. Of course, this is just theory, although we often see that the AWT is higher in call centers with long talk times compared to call centers with short talk times. For the ASA the effect of stretching time is simple: the ASA is stretched by the same factor.

Let us go back to the call center with $\lambda = 1$, $\beta = 5$, and $s = 8$. Then TSF $= 86\%$ for AWT $= 20$ seconds, and ASA $= 16.7$ seconds. Now stretch time by a factor 2, i.e., $\lambda = 0.5$ and $\beta = 10$. Then TSF $= 83\%$ for AWT $= 20$ seconds (a difference, but surprisingly small; the reason of this is explained below), TSF $= 86\%$ for AWT $= 2 \times 20 = 40$ seconds, and ASA $= 2 \times 16.7 = 33.4$ seconds.

Economies of scale Another well known property is that big call centers work more efficiently. This is the effect of the *economies of scale*: if we double s, then we can increase λ to *more* than twice its value while keeping the same service level, assuming that β and AWT remain constant.

A firm has two small decentralized call centers, each with the same parameters: $\lambda = 1$ and $\beta = 5$ minutes. With 8 agents the average waiting time is approximately 17 seconds in each call center. If we join these call centers "virtually", then we have a single call center with $\lambda = 2$ and 16 agents. The average waiting time is now less than 3 seconds, and employing only 14 agents gives a waiting time of only 13 seconds. An additional advantage is that there is more flexibility in the assignment of agents to call centers, as there is only a constraint on the total number of agents (although there will probably be physical constraints, such as the number of work places in a call center).

To give further insight in economies of scale, we plotted the two situations of the example above in a single figure, Figure 4.3. We consider the TSF, and take 7 and 14 agents. To make comparisons possible we put $\lambda \times \beta/s$ (the productivity) on the horizontal axis, and the TSF on the vertical axis. Because $\lambda \times \beta < s$, TSF gets 0 as soon as $\lambda \times \beta/s$ gets 1, no matter what call center we are considering.

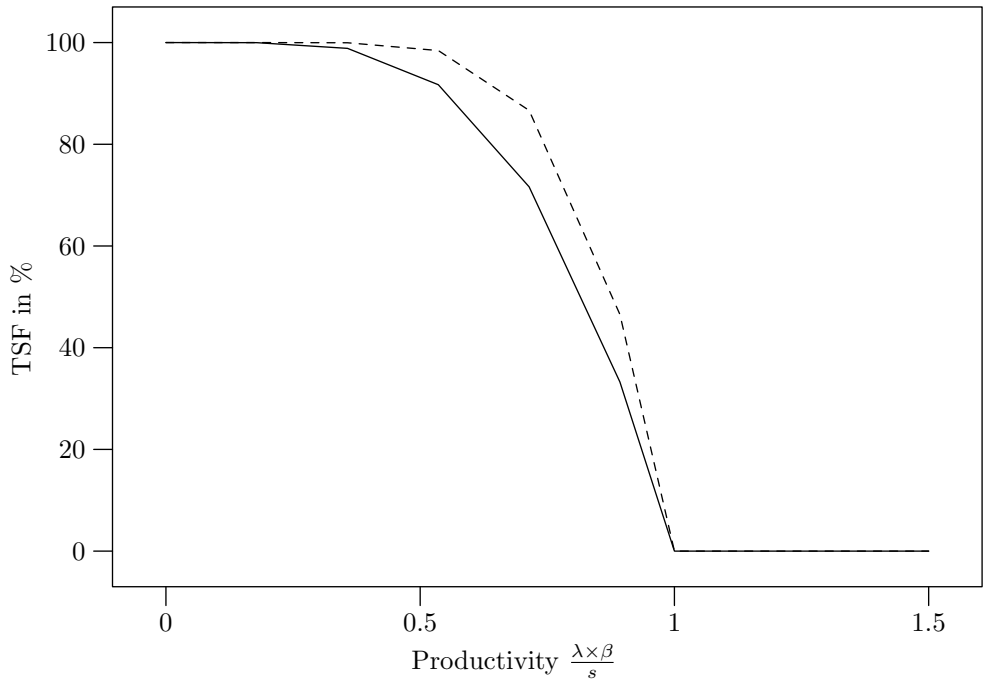

Figure 4.3: The TSF for $\beta = 5$, $s = 7$ (solid) and $s = 14$ (dashed), AWT $= 0.33$, and varying λ.

In Figure 4.3 we see that the dashed line is more to the right: for the same productivity we see that a bigger call center has a higher TSF. Stated otherwise: to obtain a target SL, a big call center obtains a higher productivity. This is related to the steepness of the curve

for productivity values close to 1, which is the sensitivity of the Erlang formula to small changes of the parameters, as discussed earlier in this section.

Variations in waiting times Consider two different call centers: one has parameters $\lambda = 1$, $\beta = 5$, and $s = 8$, the other has $\lambda = 20$, $\beta = 0.333$, and also $s = 8$. Both call centers have a TSF of around 86% for AWT = 20 seconds. Does this mean that the waiting times of both call centers are comparable? This is not the case. To make this clear, we plotted histograms of waiting times of both call centers in Figure 4.4. The level at the right of 100 denotes the percentage of callers that has a waiting time exceeding 100 seconds. We see that in the first call center, represented by the solid line, callers either do not wait at all or wait very long, there are hardly any callers that wait between 10 and 100 seconds. In the second call center (the dashed line) fewer calls get an agent right away, but very few have to wait very long.

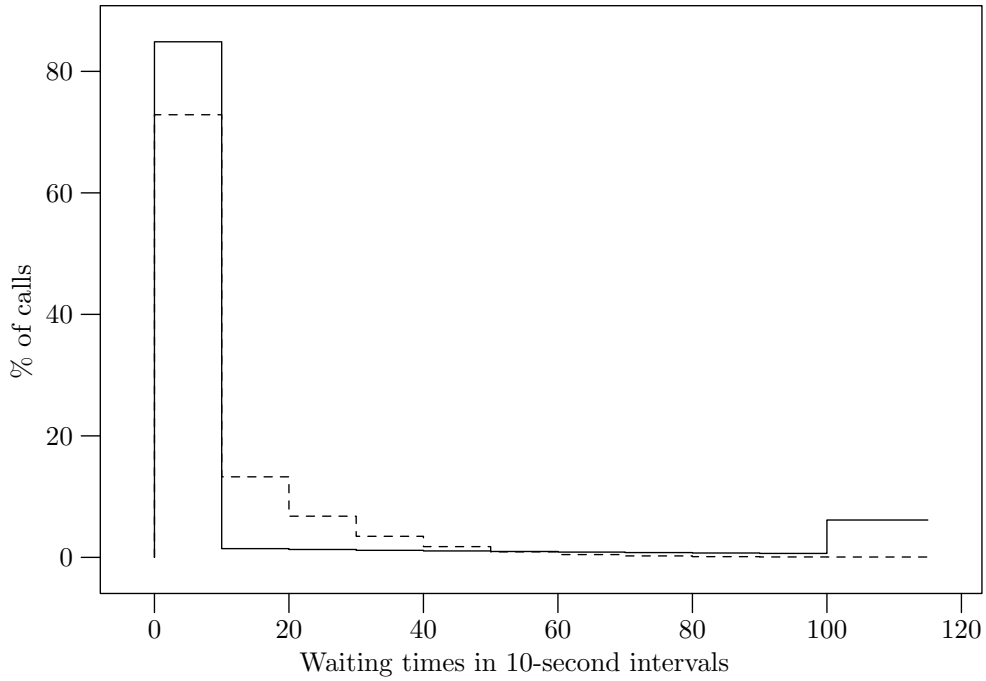

Figure 4.4: Histograms of waiting times for two different call centers.

There are two conclusions to be drawn from this example. In the first place: the TSF does not say everything. But more importantly, we see that depending on the characteristics of a call center there can be more or less variations in waiting times. Only a thorough investigation of for example the TSF for various AWTs can reveal the characteristics of a particular call center.

4.4 How good is the Erlang formula?

In this section we consider the weak points of the Erlang formula and its underlying assumptions. It will motivate some of the more sophisticated models that are discussed in Chapter 7.

It might come as a surprise that the ASA is bigger than 0 although there is overcapacity. The reason for this is the variability in arrival times and service durations. If all arrival times were equally spaced and if all call holding times were constant, then no waiting would occur. However, in the *random environment* of the call center undercapacity occurs during short periods of time. This is the reason why queueing occurs. The queue will always empty again if on average there is overcapacity. The Erlang formula quantifies the amount of waiting (in terms of ASA or TSF) for a particular type of random arrival and service times. The mathematical random processes that model the arrivals and departures are therefore nothing more than approximations. The quality of the approximation and the sensitivity of the formula to changes with respect to the different aspects of the model decide whether the Erlang formula gives acceptable results. We deal with the underlying assumptions one by one and discuss the consequences for the approximation.

Abandonments In a well-dimensioned call center there are few abandonments. Not modeling these abandonments is therefore not a gross simplification. However, there are call centers that show a completely different behavior than predicted by the Erlang formula because abandonments are not explicitly modeled. In general we can state that abandonments reduce the waiting time of other customers, thus it is good for the SL that abandonments occur! In call centers with a close to or even exceeding s it is crucial to model abandonments as well. Luckily this is possible. The corresponding model is discussed in Chapter 7.

Peaks in offered load Formally speaking, the Erlang formula allows no fluctuations in offered load. However, in every call center there are daily changes in load. As long as these changes remain limited, and, more importantly, if there are no periods with undercapacity, then the Erlang formula performs well for periods where there are little fluctuations in load and number of agents. By using the Erlang formula for different time intervals we can get the whole picture by averaging (as explained in Chapter 3). However, as soon as undercapacity occurs then the backlog of calls from one period is shifted to the next. This backlog should be explicitly modeled, which is not possible within the framework of the Erlang formula. Therefore the Erlang formula cannot be used in the case of undercapacity. For a short peak in offered traffic (e.g., reactions to a tv commercial) straightforward capacity calculations ignoring the random behavior can give quite good results. See also Chapter 7.

Type of call durations The Erlang formula is based on the assumption that the service times come from a so-called *exponential distribution*. In Appendix C we show what an

exponential distribution is. Here we just note that all positive values are possible as call durations, thus also very long or short ones, but that most of the durations are below the average. Certain measurements on standard telephone traffic show that call durations are approximately exponential, although the results in the literature do not completely agree on this subject. A typical case where call durations are not exponential is when there are multiple types of calls with different call length averages, or if a call always takes a certain minimum amount of time. In these cases one should wonder what the influence is of the different service time distributions on the Erlang formula. We can state that this influence decreases as the call center increases in size. With some care it can be concluded that only the average call duration is of major importance to the performance of the call center.

Human behavior Up to now we ignored the behavior of the agents, apart from the time it takes to take up to phone. However, agent behavior is not as simple as that. Employees take small breaks to get coffee, to discuss things, etc. Modeling explicitly the human behavior is a difficult task; describing and quantifying the behavior is even more difficult! In most situations these small breaks are taken when there are no calls in the queue. It can therefore be expected that they are of minor importance to the SL. In other situations it has a bigger impact, and it can seriously limit the possibilities of quantitative modeling.

Chapter 5

Workforce Management

The objective of *workforce management* (WFM) is to optimally trade off costs and service level, at a time scale of days and weeks. In this chapter we consider the basic call center whose architecture is simple enough that it can effectively be modeled by the Erlang C formula or one of its generalizations that will be discussed in Chapter 7. Extensions of the architecture and the corresponding steps of WFM are discussed in later chapters.

5.1 The general picture

Workforce management deals with the optimal use of the main resource in a call center, the agents. As input it uses historic call center data on traffic loads and information on agent availability; the output of WFM are agent schedules. WFM is usually done on a weekly or two-weekly basis, say four weeks in adavance.

WFM can be split into several more or less separate steps. The first is forecasting traffic load. The second is determining staffing levels for each interval. After that we have to turn the staffing levels into agent rosters. This is often split in two steps: determining shifts and assigning agents to shifts.

Once the schedule is made then over time changes have to be made as additional information comes in changing the underlying assumptions. Here we can think of changing forecasts, agent availability, etc. Finally the day comes that the schedule is executed. During it it might be necessary to take additional measures to ensure that SL and productivity requirements are met.

To support WFM many computer systems exist, that implement more or less effectively the different WFM steps. In this chapter we explain the mathematics underlying these WFM tools.

Given the scale and objectives of WFM it would perhaps be better to call it workforce *planning*. Workforce management would then involve, next to workforce planning, also other issues such as longer horizon problems related to hiring and training.

5.2 Forecasting

Forecasting is a mathematical activity that uses historic data to estimate future realizations. As such the underlying theory is part of statistics. Forecasting is used in call centers to predict future call volume.

Forecasting in call centers is not easy for a number of reasons:
- Forecasts have to be detailed, say one for every 15 or 30-munite period;
- Forecasts have to be precise;
- Forecasts depend on many known and unknown factors;
- There are many dependencies between call volume at different times;
- Business changes can have important consequences on call volume;
- Relevant data is often lacking.

Let us discuss these issues one by one.

Forecasts have to be detailed Calls usally have to be answered within less than a minute. To match the load with enough agent capacity as it is varying over the day we should not only estimate the daily call volume, but we should specify it up to the smallest interval that we distinguish in our schedules, often 15 or 30 minutes long.

Forecasts have to be precise In Figure 4.3 we saw that the TSF curve gets steeper and steeper as the productivity approaches one and as the size of the call center grows. This is equivalent to saying that the TSF is very sensitive to small changes in the forecast. Therefore the forecasts have to be very precise or other measures have to be taken to deal with unreliable forecast (see Chapter 6).

Forecasts depend on many factors Evidently, given what is said earlier, forecasts depend on the time of day. They also depend on the day of the week, and yearly fluctuations make that also the month plays a role. But that is not all: many other factors such as holidays, weather conditions, etc, can have a big impact. Some of these are known in advance, others are not. We will discuss this in detail.

There are many dependencies When call volume early on a day is high then experience shows that it will be high during the whole day. This means that there is a positive correlation between the call volume in different periods. This is an important observation that has to be exploited when reacting to changes deviations from forecasts occur.

Business changes have consequences on call volume This remark seems obvious, but it should that for example marketing decisions can have a major impact on call volume. Evidently the call center management should have the time to take measures; ideally they are involved in decisions that have a considerable impact on call volume.

Relevant data is often lacking Although in many call centers almost all transaction data is stored, data is sometimes less useful because business rules have changed since. Here we must think of the merging of call centers and the changing of scripts, changes in routing and skill groups in multi-skill call centers, or changes in products. Also changes in hardware and software play a role.

Given these observations, how should forecasts be made? We give an informal description, without going in the statistical details.

The goal of the forecast is estimating the weekly call volume. In the simplest case the daily call volume is calculated from the trend ("this year a 20% increase in call volume in comparison with last year") the seasonal fluctuations, and the distribution of call volume over the week. Based on this trend and the seasonal fluctuations of the previous years weekly call volume estimates can be made. Using the weekly and daily distributions or profiles (such as "on average 18% of the weekly call volume comes on Monday, of which again 8% between 10.00 and 10.30") estimates per day and per time interval can be made. This is possible because these profiles do hardly change, although the daily profiles depend on the day of the week (although Tuesday, Wednesday and Thursday are often similar).

It is currently week 47 in 2001. We would like to estimate the traffic in a call center in week 50. Week 46 showed an increase in call volume of 12% with respect to last year. Increasing last year's call volume in week 50 by 12% gives an estimate for week 50 this year. The quality of this estimation can be improved by computing the trend from more than just week 46, and by eliminating the influence of an exceptionally high or low traffic intensity in week 50 by considering also earlier years in the computation (as far as this is possible).

The quality of these estimates is sometimes doubtful, because there are many events that can influence call volume. They come in different categories. Events are either *internal* or *external* to the company, and either *predictable* or *unpredictable* in the sense that by the time the forecast is made the future occurrence of the event is known. Examples of external predictable events are holidays, implementation of new legislation, etc. Events such as marketing actions that generate calls are internal and (hopefully) predictable by the call center staff. Unpredictable internal events should be avoidable: they are often due to a lack of internal communication. Unpredictable external events are the hardest to deal with. Examples are bad weather generating calls to insurance companies and stock market crashes generating additional traffic to stock trading lines.

These events generate changes in offered load. We can forecast these changes as long as they are predictable. Predictable events are used twice for WFM. First to determine the expected call volume on a specific day or the week, and afterwards to update our knowledge concerning this type of event in order to improve future forecasts.

Unpredictable events need also be entered, although of course this knowledge cannot be used for forecasting these events. It is necessary however to "filter" them out of the data, if we want that the forecasts represent regular days. The question how to deal with unpredictable events remains. In Chapter 6 we will discuss this issue in detail.

We saw that predictable events can explain part of the offered traffic, while the rest of the expected load is directly extrapolated from the historical data. To estimate the

impact of the event historical data is used, but in an implicit way: the effect of the event is estimated using the data. Sometimes it is possible to base the entire estimation on the implicit use of historical data. This is for example the case if we base our estimation on the size of the customer base, and the types of customers. This can give more reliable forecasts: instead of estimating the load due to an increase in number of customers, the forecast can directly be based on the number and the average number of calls per customer. Further refinements can be introduced by diferentiating between new and old customers, the types of products that customers have, etc.

The question remains whether the forecasts are good enough for our purposes, keeping the steepness of the TSF as a function of the load in mind. It is a good habit to compare on a continuous basis the forecasts with the actual call volume. This shows whether the estimates that are generated are reliable or not, and the error that is made in the estimation can be quantified. This should be the starting point of a further analysis into the consequences of this error that should answer the question whether further measured are necessary. Some possible measures to deal with unpredictable deviations from forecasts are discussed in Chapter 6.

5.3 From forecast to schedule

Generating schedules on the basis of the call volume forecasts is a complicated task. To set the stage for the next sections we describe in this section the most basic method to generate these schedule. In later sections we discuss the drawbacks and present solutions for situations where the method to be described next cannot be applied.

The basic scheduling method consists of tree separate consecutive tasks: staffing, shift determination, and the actual scheduling of agents. We discuss them one by one.

Staffing The goal of WFM is to meet the required SL for minimal costs. In management reports the SL is often aggregated over days (and longer periods). The first step in staffing, that is usually taken implicitly, is assuming that the SL requirements should be met every interval of the day. Now the Erlang C formula can be used, with the detailed forecasts as input, to determine the number of agents that is needed in each interval. If the necessary input data is available then it can improve the accuracy of the results to replace the Erlang C formula by one of the extensions discussed in Chapter 7 (e.g., abandonments) and Chapter 8 (multiple skills). Using this approach means that the staffing levels are determined in such a way that the SL will be satisfied for every time interval, assuming that the traffic load will be as predicted and that nothing unexpected happens.

Shift determination Shifts consist of a number of consecutive time intervals, usually between 4 and 9 hours, often including one or more breaks. In a standard call center all shifts have the same length and breaks occur all after the same number of working hours. In this case it is mathematically speaking a relatively simple problem to find the optimal combination of shifts. This type of problem can already be solved with standard software

that is available on most PC's (often without the user realizing that it is there). The Excel Solver is indeed the right type of tool to solve these problems, and there are many more tools with similar possibilities on the market. Most WFM tools make calls to one of these solvers in a to the user transparant way and give the optimal mix back to the WFM tool. This is then passed on to the scheduling or rostering module.

Schedule determination In the current situation where the starting hour is the only difference between shifts a simple procedure in which agents can choose their own shifts is often best. Also in situations where homogeneous shifts are assigned to agents by call center management there is usually no need for mathematically advanced assignment methods, the computer usually serves only for administration purposes.

5.4 On staffing requirements

The standard way in call centers to determine staffing is as described above: the required daily or weekly SL is required for every 15 or 30-minute interval. An interesting question is whether this is really required, or whether one allows fluctuations during the day, as long as the average SL (as we learned to calculate it in Section 3.3) is as required.

We will not go into this discussion; instead, we show what can be gained if we only require the daily average. For a simple numerical example, consider only two intervals that have to be scheduled: one with $\lambda = 10$ per minute, $\beta = 1$ minute, and AWT = 20 seconds, and the second interval with $\lambda = 1$ per minute. The difference of a factor 10 is not atypical, sometimes the difference between the busiest and least busy intervals are even bigger. In Table 5.1 we show the results of different staffing levels per interval and over the average of both intervals. In the first line we see the numbers of agents needed to obtain the required TSF (80%) in both intervals, resulting in an overall SL of more than 90%! Because the second interval has a small impact on the overall SL, we see that reducing s in the second interval still keeps the overall SL above 80%. Reducing s in the first interval would lead to a TSF under 80%. Interesting enough, moving one agent from interval 2 to interval 1 improves the overall TSF, as is shown in fourth line. Moving one agent more does not increase the SL. We conclude that letting go of the requirement that the SL requirements should be met for each interval can improve overall SL and reduce costs. As far as we know no WFM tool has implemented this.

s in interval 1	s in interval 2	TSF in interval 1	TSF in interval 2	overall TSF
13	3	89.51	95.33	90.04
13	2	89.51	76.12	88.29
13	1	89.51	0	81.37
14	2	95.41	76.12	93.66

Table 5.1: TSF for two intervals and their weighted average using the Erlang C model

5.5 Integrating steps

There is another reason not to adher strictly to the interval requirements, but to consider only the daily SL requirements. This is the fact that when staffing is done based on a fixed staffing level for each interval, then sometimes considerable overcapacity occurs, because the length of shifts and relatively short peaks in traffic load cannot be matched. Thus the "best" shift mixture not only satisfies the staffing level at all times, but around peaks it exceeds this level because we cannot hire agents for the peaks only. A good solution is allowing a low SL for certain intervals, as long as the SL constraint is satisfied on average over the whole day.

More often than a restricted number of shifts with fixed length we see a multitude of different possible shifts with varying lengths. Then there are many different good solutions with different mixtures of shift lengths. Shift lengths are often part of the contracts that agents have. Which mixtures are possible depends therefore strongly on the preferences and contracts of the agents.

The other way around, the decision which type of contract to offer to an agent is an important decision with consequences for the scheduling step, but also consequences when it comes to costs. Small shift lengths make scheduling easier, and avoid unnecessary overcapacity. On the other hand, more agents have to be hired in total in the case of short shifts, and therefore overhead costs (such as training and monitoring costs) are higher. Very long shifts also reduce the efficiency of agents.

When two agents use car pooling to get to work they should have the same shifts. Thus we should already in the shift determination step take this into account: a shift, with the proper requirements, should be chosen at least twice. We see that agent preferences, that usually come into play while making rosters, already play a role in the shift determination step. This calls for an integration of both steps. Also if the roster requirements are highly personalized then an integration of scheduling and rostering is called for.

But in general it is again a complicated task, in which we often cannot focus on a single day, as contracts often specify the weekly number of working hours. Again, mathematical solvers are excellent tools to find feasible rosters.

simple example

We saw that, except for forecasting, there are sometimes good reasons to integrate all WFM steps. A drawback is the complexity of the resulting integrated problem. However, computers and mathematical software are nowadays powerful enough to handle also these problems.

5.6 Decision support systems

The different steps of WFM are implemented in a magnitude of WFM tools. Around their mathematical cores nice graphical user interfaces (GUI's) are built, adding many possibilities. We will not give web sites, they can easily be found using a search engine on

the web. Note however that the functionality of the tools varies enormously. In practice we see that many tools are only used partly, and that specially build tools for forecasting and scheduling are often used. WFM tools are mostly used for getting the data out of the PABX and for determining staffing levels. Other functionality (scheduling, rostering) is less used.

The main reason for this is that every call center is different. Of course, call centers have much in common, but every call center has its particularities which makes that a standard sofware solution does not fit. The choice is taking this for granted and buying a standard tool, or developing tailor-made software. As stated we often see compromises between the two, where standard tools are used partly.

5.7 Workforce planning

Section to add about long-term planning of workforce. Refer back to remark on shift lengths on page 5.5.

Chapter 6

Variations, uncertainty, and flexibility

In Chapters 4 and 5 the mathematical background is given for basic call center management. In this and the next chapters we discuss more advanced topics. This does not mean that they are less relevant: many call centers form a complicated environment that demands knowledge of the topics discussed here. We start with an in-depth discussion of the consequences of uncertainty for call centers.

6.1 Variations and the need for overcapacity

Every call center manager tries to combine a high service level with a high productivity. In the previous chapter we saw why this is not always possible, due to unavoidable variations in call holding times and interarrival intervals. The Erlang formula quantifies the influences of these variations, and shows what the variations cost in terms of additional personnel.

Consider a call center with $\lambda = 4$ and $\beta = 5$. The offered load is therefore 20 Erlang, and without any variations 20 agents would suffice to obtain a 100% SL. However, the Erlang formula shows that, under the usual variations, we need 5 additional agents to obtain a 20/80 SL, thus 25% overcapacity.

In the previous chapter we saw that increasing the scale smooths out these short-time variations. Indeed, doubling the number of offered calls in the example reduces the needed overcapacity from 25% to 15%, as can easily be verified using the Erlang calculator. Unfortunately, increasing the scale is not always possible, and even in large call centers some overcapacity is needed. Additionally, there are other types of variations that at first sight demand additional overcapacity.

Consider the average number of incoming calls λ. This number is the outcome of the forecasting procedure, and therefore an *estimation* of the real value. What if the estimation is 10% off? In the example it leads to a 65% SL. When the offered load is twice as high the SL under a 10% load increase is only 40%! We see: the bigger the call center, the more important the consequences of load changes.

There are more uncertain elements in call centers than just the offered load. An important one is the variation in agent availability, mainly due to illness. To assure the SL also under these circumstances we need to schedule additional overcapacity. E.g., we need 2 additional agents in the call center of the example to be able to cope with 10% illness.

Having scheduled this costly overcapacity, the following question remains: what to do if there is a 10% decrease in offered traffic? What if all scheduled agents are indeed available? In the example, a 10% decrease in offered load would require 23 agents, 2 less than the normal situation, and 4 less if we anticipated a 10% increase in traffic! In the next sections we discuss a way to deal with these unpredictable variations.

6.2 Averages versus distributions

The essence of entities such as the arrival rate and absence percentage is that they show fluctuations that cannot be predicted timely and entirely. When a heavy storm damages many houses it is too late for an insurance company to change the weekly schedule. The same holds when in the morning a larger than usual number of agents appears to be ill. And even for predictable events such as holidays, it is sometimes hard to estimate their influence on the load.

It is well possible to estimate averages. This is what forecasting is about, and every call center manager can tell the average absence percentage. However, we also know that fluctuations around this average occur frequently: now and then fewer than average agents are ill, and then again more than average. Similarly, in many call centers we also see that the offered load cannot be predicted accurately, no matter how much effort is put into it. Instead, we should accept that fluctuations around the average occur. Now the attention shifts to *quantifying* these fluctuations, and reacting accordingly.

The usual measure for the size of fluctuations is known from statistics as the variation. Based on the average and the variation a bell-shaped curve can be constructed representing the frequency table of for example the offered load. Under specific assumptions other approaches can be more appropriate. For example, if we assume that the illness of an agent has no relation with the illness of other agents, then it suffices to known the probability that an agent is ill. Given this it is straightforward mathematics to compute the fraction of days that a certain number of agents is ill. This can again be plotted in a frequency table. Such a frequency table is also known as a "distribution".

6.3 The need for flexibility

Having quantified the variability, we now have to take the appropriate measure. Our goal is to come to an acceptable cost-SL trade-off, in the presence of additional variations with which we are confronted after having scheduled the agents. Central are flexibility in the use of the workforce and reduction of the influence of variability. We start with the first.

By introducing flexibility at all time levels of the operation we can offer the required SL

while keeping a high productivity at the same time. At the highest level we have flexibility in contracts. With this we mean that for certain agents we can decide on a very short notice (e.g., at the beginning of the day) whether we require them to work or not. Of course they get paid for being available, and often they are guaranteed a minimum number of working hours per week. This is an excellent solution to deal with variability in arrival rate and absence. For the latter this is obvious; for the former we have to realize that the arrival rate during the first hours of the day often gives a good indication of the load during the rest of the day. Thus early in the morning it can already be decided whether additional agents are needed.

When trying to quantify this, we start with a minimum number of fixed contract agent. This minimum is based on some lower bound on the arrival rate and a minimal absence. Then we assure that there are enough agents with flexible contracts such that we can get the number of agents equal to the number required in the case of a maximal arrival rate and maximal absence.

A call center has an arrival rate falls between 4 and 4.8, with 90% probability. For the lower bound 50 agents are needed, for the upper bound 9 more. Out of these 50 agents between 1 and 6 agents are absent, on average 3. Thus we schedule at least 51 agents, and in the "worst" case we have to hire 14 more, on average 6.

Introducing flexible contracts gives us the possibility to handle days with a higher than usual traffic load. If the peaks are shorter, in the order of an hour, then we cannot require agents to come just for this short period of time. In this it is possible to mobilize extra workforce by having personnel from outside the office work into the call center.

example stock crash, everybody mobilized.

Although this seems a simple solution for emergency cases, one should realize that the extra agents should be trained and that the telephony and IT equipment should be in place to accommodate all agents.

A final type of flexibility is flexibility in task assignment. This is a method to react to load fluctuations that can even work at the finest level of fluctuations, that the Erlang formula accounts for. For this it is necessary that there are, next to the incoming calls, other tasks that have less strict service requirements. Examples are outgoing calls and faxes, and more recently emails and messages entered on Web pages. They have service requirements that range from hours to days, thus of a totally different scale than the requirements of incoming calls. To be able to satisfy the service requirements for these so-called channels it suffices to schedule just enough agents to do the work. Scheduling overcapacity, as for incoming calls, is not necessary. It also doesn't matter when outgoing calls or emails are handled, as long as they are handled in the required time interval. This makes it possible to use outgoing calls to fill in the gaps left by a low offered load, and allows in case of undercapacity agents originally scheduled for emails or outgoing calls to work on incoming calls. Thus instead of assigning in a fixed way agents to ingoing or outgoing calls, they are assigned dynamically (either by the supervisor or automatically) to a certain channel. This assignment should be done carefully. A free agent should obviously be assigned to a

waiting incoming call if any are present. A way to maximize productivity is by assigning free agents to outgoing calls if there are no waiting incoming calls. However, then every incoming call has to wait for a free agent. In most situations this will lead to a very low SL. The solution is to keep a number of agents free for incoming calls when none are waiting. This rule works when changing from ingoing to outgoing calls takes relatively little time. It is known as call blending, as it was originally intended for call center dealing with inbound and outbound traffic. Simply *blending* seems a more appropriate name given the recent focus on communication over the internet.

numerical example

6.4 Reducing the impact of variability

In the previous section we discussed ways to deal with variations by introducing flexibility in agent availability and task assignment. A different approach to dealing with variations is by reducing them or by reducing their impact. Consider the following example.

To make reservations for international travel the Dutch railways have two options that can be done from your home. The first is calling the contact center on a 0900-number, i.e., the caller pays for the call. The second is entering your travel data and the moment at which you want to be called back (a 4 hour interval) on a web page. Potential travellers are thus financially stimulated to enter their data on the web page, thereby turning an inbound call into an outbound call. This allows the contact center to contact you at some quiet moment during your preferred time interval. Often the call takes little time as the agent already known the travel options, based on the data that you entered.

The example clearly shows the advantage of outbound or email contact over inbound calls. Instead of having to answer within 20 seconds after the arrival of a call, you can take the moment in a long interval that suits you best. In general, the same amount of work is done with less agents and at a higher SL. This is a direct consequence of the fact that outbound calls have a less strict service level requirement. Then we assume of course that call blending is being used, that agents are not assigned in a fixed way to either incoming or outgoing.

Another way to turn inbound into outbound that is especially effective in reducing peak loads is by offering callers a call-back service: their telephone numbers are registered and they are promised to be called back as soon as possible. Take the following example.

A manager of a free 0800 service is complaining, and with reason. Her SL is lousy and her telecom costs are going over the top. Due to the bad SL customers abandon: the call center is paying the telecom costs of customers that not even reach the call center! The bad SL is the result of increased customer attention that went too fast for the call center to cope with by hiring new agents. Thus the increase in customers does not lead to an increase in income, only to an increase in costs!

The answer to this kind of situation is limiting the number of callers that can wait simultaneously in queue. This can be done by asking people to call back or by asking them to leave their number so that they can be called back. This way callers that get no service do not wait in queue. Not only the costs are reduced (in case the call center pays for the communication), it is also customer friendly. Certainly if the offered load is high then there is no sense in making callers wait, there will always be new callers for free agents to handle. Calculating how to set the number of lines that can be used simultaneously requires extending the Erlang formula. This is the subject of the next chapter.

Chapter 7

Extensions to the Erlang C model

In the previous chapter we discussed several measures to improve the performance of the call center. To quantify the impact if these improvements with respect to the standard situation, represented by the Erlang C formula, we have to extend the underlying Erlang model. We also saw that the Erlang C needed improvements to better model reality: abandonments is a good example. We indicate all these extensions with Erlang X. This Erlang X model is the subject of this Chapter.

7.1 Blocking

Theoretically speaking, the Erlang C model allows an unbounded number of queued customers. Not only will this never occur because callers abandon, it is also impossible because the number of lines available to connect to the call center is limited. Thus blocking can never be completely ignored. As we saw in the previous chapter it can even be advantageous to block customers, even if there is still capacity: they increase both abandonments and waiting times.

To determine the best number of lines (or, equivalently, the maximum number of customers in the system) we have to calculate productivity and waiting times for various numbers of lines. This allows us to see the trade-off between the two and make a justifiable choice. To calculate productivity and waiting times we have to make certain assumption about customer behavior in order to build the mathematical model. An important choice is related to the behavior of callers that are blocked. Either they are lost, they try to call again later, or they are called back as soon as the load permits. Each of these choices requires a different model and leads to a different performance.

an example

7.2 Abandonments

As soon as waiting occurs it is inevitable that callers abandon. Some callers abandon as soon as they enter the queue; most abandon after passing some time in the queue. Determining the patience of callers, i.e., the time that they accept to spend in the queue, is, mathematically speaking, a difficult task because most callers reach an agent before their patience is over. The average patience is not simply the average time abandoned calls spent waiting. As an extreme examples, assume that it never occurs that callers have to wait longer than 20 seconds; then a patience of longer than 20 seconds would never occur! Estimating the patience of callers from previous data is therefore a complicated task that requires sophisticated statistical analysis.

After obtaining the patience distribution we can include abandonments in our analysis. Under certain statistical assumptions concerning this distribution this is a relatively simple task. These assumptions boil down to two things:
- callers abandon the moment they are queued with a certain probability;
- callers in queue abandon within the next second with a probability that does not depend on the time they have already spent waiting.

Naturally, when doing numerical experiments we see the same behavior as in reality: abandonments decrease productivity somewhat (as less callers reach an agent), but also the waiting times decrease, it can even occur that the overall service level increases!

an example

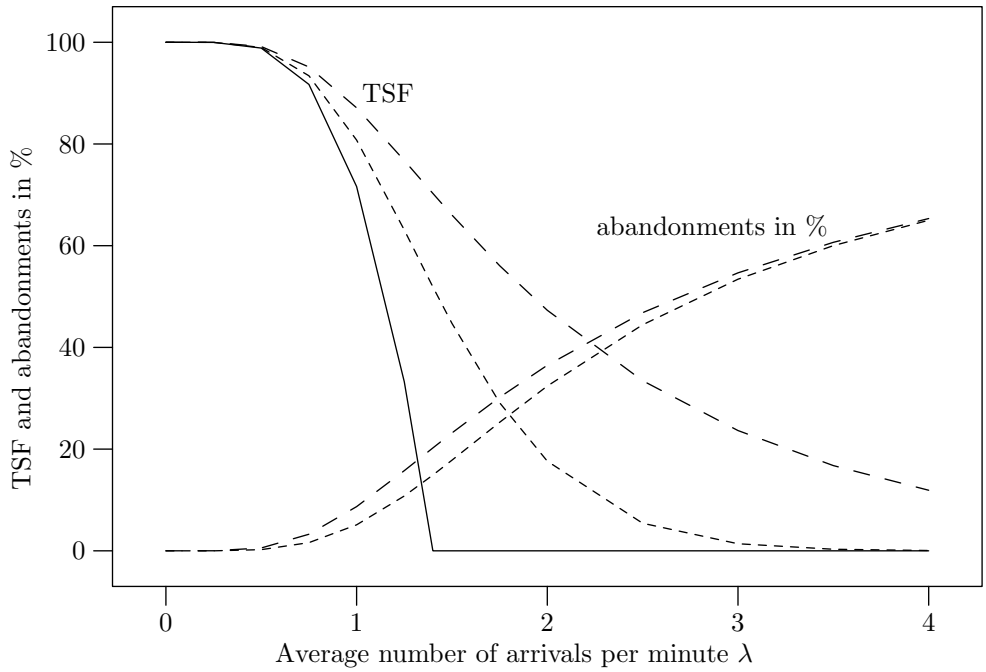

Figure 7.1: TSF and abandonment percentage for average patience ∞, 5 and 1 (from below), for $\beta = 5$, $s = 7$, AWT $= 0.33$, and varying λ.

An interesting subject is the psychology of abandonment and the ways to influence it. We measure the time at which people abandon, which we call their patience. It suggests that people base their decision only on the time spent in queue. This however is doubtful: why do people abandon at entering the queue? They have no patience at all? It also suggests that the caller's behavior cannot be influenced by given additional information such as expected waiting time which is not realistic. Another way to explain abandonments is based on the idea that people make their decision on the time that they expect that they still have to wait. It explains that some callers abandon at entering the queue: they expect that their waiting time surpasses the time that they accept to wait. At first sight it does not explain why people abandon while waiting in queue: as they wait the remaining waiting time decreases, so why abandon? The reason is that they did not know their own waiting time, while waiting callers learn about their own waiting time. Surprisingly enough it can be shown mathematically that the remaining waiting time in a standard call center, modeled by the Erlang formula, stays always the same, no matter how long a caller has waited already. So, mathematically speaking, one would say that the rational caller has no reason to abandon while waiting. However, a waiting caller does not only learn about his own waiting time, he also learns about the situation the call center was in the moment he or she arrived. In other words: if you have to wait long, then you are probably calling to a badly managed call centers meaning that you probably still have to wait for a long time. And this is a good reason to abandon while waiting for some time.

In some call centers there is a direct connection between each call and the profit a company makes: a call means on average a certain income to the firm. It is hard to imagine a company with a business model that simple (even a mail order firm likes happy customers that phone back again), but it is an interesting exercise to pursue it. Using the Erlang model extended with abandonments we see that every additional agents increase less the productivity, and therefore a break-even point is reached at some number of agents. What the corresponding service level is depends of course strongly on the parameters.

7.3 Blending

There are two types of questions related to blending:
- what are the productivity and waiting times given a specific way by which blending is implemented;
- how to implement blending in the best way?

As is already stated in Chapter 6 the waiting times are normally unacceptably high when every free agent is used for outbound calls or some other non-inbound task. Thus some capacity has to be kept free for inbound calls only. The best way, in terms of the productivity-waiting trade-off, to implement blending is as follows. Incoming calls get priority over other tasks, i.e., agents that become available are assigned to inbound calls if any are waiting, and arriving calls are assigned to an agent if any are free. Thus outbound calls (or other tasks) can only be scheduled if there are free agents and no waiting inbound calls. The optimal rule is to schedule an outbound call as soon as there are more than a

certain number of free agents. This number is called a *threshold*. Its optimal value depends on the trade-off between productivity and waiting times. If the threshold is relatively small, then productivity will be high and waiting long, and v.v.

7.4 Overload situations

In Section 3.3 we saw how to calculate the service level using the Erlang formula by averaging the service levels of each interval. This approximation is called the *pointwise stationary approximation* (PSA), because it does not take into account the transitions from one interval to another. This is not problematic, as long as the load a is smaller than the number of servers s, in other words, if the agents can, on average, handle the traffic. If this is not the case for one or more intervals, then the PSA will give wrong results. Over the interval(s) where undercapacity occurs the Erlang formula will give SL 0%, which is often not the case: the SL deteriorates over the interval, but it is not 0%. On the other hand, as soon as the parameters change and we find ourselves again in a situation of overcapacity, then the Erlang formula predicts right away the SL belonging to these parameters, not taking into account the backlog of customers from the previous interval. This leads to a SL prediction that is too high for this interval. It is tempting to state that both errors will cancel out, but unfortunately there is no ground to assume that this is always the case. An alternative method is needed. A simple and still quite accurate method is based on the idea that under undercapacity the random fluctuations of offered load play a lesser role. This means that the behavior in this situation is well predicted by the average behavior. We illustrate this with an example.

example fluid system, difference in workload and SL. Figure, PSA as well in it.

Chapter 8

Multiple skills

When call centers increase in size, we speak of economies of scale. Call centers also increase in the number of tasks that they execute, multiple skills is the rule, not the exception. The obtained advantages are called economies of *scope*. How to get the maximum out of multi-skill call centers is the subject of this chapter.

8.1 The framework

Multiple skills are only useful if the ACD can differentiate between the skills needed. There are different ways to obtain this differentiation. One way is installing an VRU (voice response unit) where callers have to choose; another way is communicating different numbers to the clients, depending on the required skill. The result is that there are different queues at the ACD for the different skills.

The term skill suggests that different agents can handle different skills. This is only partly true. In the first place, it is common for agents to be able to handle more than one skill. E.g., in a multi-lingual call center an agent might speak more than one language. Agents that have all the skill are called *generalists*; all other agents are called *specialists*. Although it often occurs that many agents have only one skill, it should not hold for all agents. In that case the agents for the different skills operate as separate independent call centers, so no economies of scope are obtained. Sometimes different call classes do not require different skills, but only require a different SL. This falls within the framework presented in this chapter.

Multi-skill call centers pose some challenging problems for the manager and the mathematician. The problems of single-skill call centers, as discussed in earlier chapters, play again a role, but become more complex. There are also problems that are unique to multi-skill call centers. Starting at the smallest time scale, a new problem of extreme complexity emerges: how to assign calls to agents in an optimal way? This problem has to be solved automatically thousands of times per day in every multi-skill call center, and a really satisfying solution to this problem does not yet exist. However, several special cases have properties that make optimal solutions possible, and also for the general problem solution

methods that perform reasonably well exist.

At a longer time scale we encounter the problem of scheduling agents. This also becomes extremely complex, due to the increase in possibilities that we have: which mixture of skills has the best cost-performance trade-off?

At a longer time scale, at the tactical level, the problem of hiring the right number of agents becomes one of hiring and educating the right number of agents. With educating we refer to the fact that in many call centers skills can be learned. This enables call center agents to have career paths in which one progressively acquires new skills. This is not always possible: in a multi-lingual call centers for example agents with the right language skills have to be hired, call centers usually cannot afford it to teach agents a language at the call center's expense. Education and hiring, while taking turn-over and fluctuating demand into account is a challenging task.

Finally, having multiple skills makes the organization of the call center more complex. Efficiency considerations should always be taken into account when making changes in the overall structure.

8.2 Routing calls

The complexity of the routing problem in multi-skill call centers can differ enormously. Sometimes we see only a few skills (e.g., different lines for B2C and B2B), sometimes we see tens of different skills (e.g., a call center have different product skills and language skills). Obviously, the routing problem is more difficult when there are more skills. However, good routing is not even always simple when there are only a few skills.

There are several aspects to good routing. They all play a role in the following example.

Consider the simplest situation possible, where there are only two skills and one requires less skills than the other. Evidently, when possible each call should be served by an agent belonging to the right group. But what if all agents of the .... (Assume that the SL requirements are equal.)

Appendix A

Definities

ACD *Automatic Call Distribution*, een onderdeel van een telefooncentrale dat ervoor zorgt dat calls die op een centraal nummer binnenkomen aan de agenten in een bepaalde groep worden toegekend.

afhaker Een call die door de beller wordt afgebroken *voordat* hij of zij een agent aan de lijn krijgt.

agent Een medewerker die inzetbaar is in het call center.

bedrijf Organisatie die een call center opereert.

beller Klant die het call center telefonisch probeert te bereiken.

bezetting Een groep agenten die het call center bemenst.

call Een telefoongesprek dat via het call center gevoerd wordt. We maken onderscheid tussen *inkomende* en *uitgaande* calls.

call blending Het door een meerdere agenten uit laten voeren van zowel ingaande als uitgaande calls.

call center Een verzameling middelen, waaronder een telefonische installatie en agenten, waarmee telefonische dienstverlening kan worden verricht.

CTI *Computer and Telephony Integration*, het proces dat communicatie tussen en integratie van telefonische en computerfaciliteiten mogelijk maakt.

ICT *Information and Communication Technology*, technologie die betrekking heeft op computers en telecommunicatie.

klant Persoon die zich al dan niet telefonisch met een bedrijf in contact wil stellen.

predictive dialer Functionaliteit van ACD t.b.v. uitgaand verkeer dat a.d.h.v. een nummerbestand automatische calls initieert.

wachttijd De tijd die een call doorbrengt tussen het moment van contact leggen met het call center en het moment dat een agent de call in behandeling neemt.

Appendix B

Annotated bibliography

This annotated bibliography tries to assist the reader in delving deeper into the subject of call center mathematics. By no means it is our objective to be complete.

- N. Gans, G.M. Koole, and A. Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management*, 2003. To appear. Electronically available at www.cs.vu.nl/obp/callcenters.
This paper gives the current state of the art concerning call center mathematics. It is written for academics, and assumes solid mathematical knowledge.

- A.N. Avramidis, A. Deslauriers, and P. l'Ecuyer. Modeling daily arrivals to a telephone call center. Working paper, 2003. Electronically available at www.iro.umontreal.ca/ lecuyer/p
This paper presents the state of the art in call center forecasting, especially when it comes to correlations between arrival counts in different intervals.

Appendix C

The mathematics

In this appendix we use a somewhat more involved mathematical notation. Knowledge of this is necessary to understand the formulas.

C.1 The Erlang C model

In Chapter 4 we gave a formula for the expected waiting time $\mathbb{E}W$ in the Erlang C model. The probability of delay played an important role in this formula. This probability is given by the following formula:

$$C(s,a) = \frac{a^s}{(s-1)!\,(s-a)} \left[\sum_{j=0}^{s-1} \frac{a^j}{j!} + \frac{a^s}{(s-1)!\,(s-a)} \right]^{-1}.$$

Of course this formula holds only if $s > a$, if $s \leq a$ then every caller is delayed and the probability of delay is thus equal to 1. We repeat the formula for $\mathbb{E}W$. All variables are now known.

$$\mathbb{E}W = \frac{C(s,a)\beta}{s-a}.$$

In earlier chapters we encountered several times the probability that the waiting time does not exceed a certain value t. We didn't give a formula for this expression in Chapter 4. It is as follows:

$$\mathbb{P}(W > t) = C(s,a)\mathrm{e}^{-(s-a)t/\beta}.$$

Here $\mathbb{P}$ should be read as "the probability that ...".

C.2 The Erlang blocking system

C.3 De exponentiële verdeling

Plaatje van de dichtheid en uitleggen Poisson proces, als aannames Erlang formule.

C.4 Geboorte-sterfte processen

Uitleggen geboorte-sterfte processen, afleiden Erlang formules.

C.5 Square-root staffing rule

Up to now we saw that an increase of scale leads to advantages with respect to productivity and/or service level. These advantages can always be quantified using the Erlang formula. To obtain a general understanding we formulate a rule of thumb that relates, for a fixed service level, call volume and the number of agents.[1] In a formula this relation can be formulated as follows:

$$\text{overcapacity in } \% \times \sqrt{s} = \text{constant}.$$

The constant in the formula is related to the service level, the formula therefore relates only overcapacity and the number of agents. the percentage overcapacity in the formula is given by $100 \times (1 - a/s)$. From the rule of thumb we obtain results such as: if the call center becomes four times as big, then the overcapacity becomes roughly halve as big. How we obtain this type type of results is illustrated by the following example.

A call center with 4 agents and $\lambda = 1$ and $\beta = 2$ minutes has an average waiting time of a little over 10 seconds. For this call center the associated constant is $100 \times (1-2/4) \times \sqrt{4} = 50 \times 2 = 100$. If we multiply s by 4, than $\sqrt{s}$ doubles. Thus to keep the same service level (the same constant), we halve the overcapacity to 25%. Thus the productivity becomes 75%, and thus with $s = 4 \times 4 = 16$ this gives $a = 12$ and $\lambda = 6$. If we verify these numbers with the Erlang formula, then we find an average waiting time of a little over 6 seconds. Closest to 10 seconds is $s = 15$, with approximately 12 seconds waiting time. If we multiply s again with 4, then the overcapacity can be reduced to 12.5%. This means $\lambda = 28$, with 3.2 seconds waiting time. Closest to 6 seconds is $s = 62$, from which we see that the rule of thumb works reasonably well.

From the example we see how simply we can get an impression of the allowable call volume if we change the occupation level. More often we prefer to calculate the number of agents needed under an increase in call volume. The calculation for this is more complex. If we denote with c the constant related to the service level divided by 100, then the formula for s is:

$$s = \left(\frac{c + \sqrt{c^2 + 4a}}{2} \right)^2 .$$

As in the previous example we start with 4 agents, $\lambda = 1$ and $\beta = 2$ minutes. The number c is the constant divided by 100, thus $c = 1$. Filling in $a = \lambda \times \beta = 2$ and $c = 1$, then we find indeed $s = 4$. Assume that λ doubles. Then $a = 4$, and with $c = 1$ we find $s \approx 6.6$. This is a good approximation: $s = 7$ gives a waiting time under the 10 seconds, $s = 6$ above. If $\lambda = 10$, the we

[1] The remainder of the paragraph is of a more mathematical nature and can be skipped without consequences.

find $s = 25$ as approximation. An agent less would give a waiting time of 9 seconds. If λ doubles again, then we get 47 as approximation, with 45 as best value according to the erlang formula. We see that for big values of λ doubling leads to doubling s.

If c is small with respect to a then we see that s is proportional to a. This means that the economies of scale become less for very big call centers, because it is already at a maximal level. What "big" is in this context depends on the service level.

Using this rule of thumb should be done with care. It is only useful to relate λ and s. Next to that, one should realize that it is only an approximation, the results need to be checked with the Erlang formule before use in practice. This point was illustrated in the example.

Appendix D

Other appendices

index, bibliographic notes ("further reading"), links