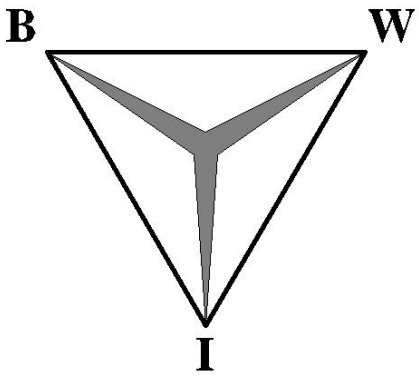
Skill-based routing in multi-skill call centers



Nancy Marengo nmarengo@few.vu.nl
BMI-paper
Vrije Universiteit
Faculty of Sciences
Business Mathematics and Informatics
1081 HV Amsterdam
The Netherlands
November 2004

Supervisor: Sandjai Bhulai S.bhulai@few.vu.nl

Preface

The paper that lies before you is the final result of a literature study that forms a compulsory element of the study Business Mathematics and Informatics at the Faculty of Sciences at the Vrije Universiteit in Amsterdam.

One of the subjects that are discussed during the course “Modeling of Business Processes” is call centers. During that course I became very interested in that subject and therefore I wanted to do some research on that subject. I talked with my supervisor Sandjai Bhulai about possible subjects and finally I chose for “skill-based routing” because there are a number of interesting questions related to it.

I would like to thank my supervisor Sandjai Bhulai for his time and advice.

Nancy Marengo
November 2004

Contents

1 Introduction

Nowadays, many companies such as banks, insurance companies and telecom providers provide their customers service via the telephone. A physical location where these services are delivered is called a call center. A call center constitutes personnel, computers and telecommunication equipment [3].

A well organized call center can enable a company to:
- improve the contact with the customers;
- enlarge the income by multiplying the number and kind of trade channels;
- improve the position of competition by making good use of the technology.

The overall challenge in designing and managing a call center is to achieve a balance between operational efficiency and service quality. This balance can be achieved by making use of workforce management (WFM) that deals with the optimal use of personnel.
Workforce management constitutes of the following four steps:

1. forecasting the load of the system (based on historical data);
2. determining the minimum number of employees (also called agents) needed in each time interval;
3. determining the number of shifts;
4. assigning shifts to agents.

1.1 Problem statement

Modern companies thrive to provide their customers service that is customized to customers needs. With the growing level of service customization the variety of services provided by those companies is highly increasing. This requires employees with a large set of skills.

For example, Compaq's call center can serve calls in eleven different languages [1]. This means that the agents must be able to speak those languages and for example, they also need to understand cultural differences between the different customers.
You can imagine that those agents cannot all speak those eleven languages: it may simply not be possible that an agent can speak so many languages and personnel costs constitute a significant amount of operation costs. Therefore it is most likely that agents should not be trained for being capable to offer service in all of the eleven languages.
A possible solution is to hire agents that speak a subset of the eleven languages, in different combinations. A consequence is that an agent that only speaks Dutch and English cannot serve a French speaking customer. To manage incoming calls to a so called multilingual call center a routing mechanism is used that routes incoming calls to agents. For example, incoming calls are first identified by language and then they are routed to agents who speak the same language.

The assignment of calls to agents having the required skills to handle those calls is called *skill-based routing*. The goal of skill-based routing is to route incoming calls in an intelligent way in order to achieve a high level of service and flexibility, against low costs.

Skill-based routing becomes a necessity in every multi-skill call center, and thus not only in multilingual call centers. A multi-skill call center is a call center that handles multiple types of calls that for example are classified according to the type of service requested, the spoken language and the perceived value of customers. In a multi-skill call center agents can handle different call-types, depending on the multiple skills that they have.

Nowadays, skill-based routing is receiving increasing attention. This paper gives an introduction to skill-based routing and its complexities.

Before going further on the subject *skill-based routing* chapter 2 gives an introduction to call centers in general. It explains how a basic call center works and how it is modeled. A basic call center is manned by identical agents who handle calls of one type. In practice the agents are not identical and call centers handle different types of calls.

Chapter 3 illustrates how skill-based routing works. There are two types of skill-based routing, static and dynamic skill-based routing. Both types are illustrated by giving examples.

Not only the routing of calls in multi-skill call centers can be very complex, but also the staffing problem: how many agents with a particular skill set are needed to meet given service level requirements? Chapter 4 illustrates how skill-based routing affects the staffing problem in a multi-skill call center. Finally, this paper ends in chapter 5 with a conclusion.

2 An introduction to call centers

Call centers can be categorized among different dimensions:

- *Multiple functions*
  Call centers can provide multiple functions: from help desk and customer service to order taking, telemarketing and emergency service.

- *Size*
  Call centers vary in size, from a small call center with 9 agents to big call centers in which tens or hundreds of agents can work at the same time.

- *Inbound or outbound calls*
  Two types of calls can be handled by call centers, namely: inbound and outbound calls. In the case of inbound calls, people are calling the call center to receive some service. With outbound calls, the call center (or the agents) contacts people by, e.g., phone, e-mail or fax. Today, in many call centers both types of calls are mixed. This is also called call blending.

- *Multiple skills*
  As indicated before, agents can have different skills and they can be distinguished by the number of skills they have. When the skill level required to handle calls is low, agents may be trained to handle every type of call. Agents that can handle every type of call are also called generalists. When the skill level required to handle calls is high, agents may be trained to handle a subset of the call-types that the call center serves.

2.1 What happens with incoming calls

This paragraph gives an overview of how a call center works. In what follows it is assumed that only inbound calls are handled.

Consider customers who want to contact a call center agent. The whole process starts by dialling a number that is often free (e.g., in the Netherlands it will be a 0800 number).
The call center has a switch, called a private automatic branch exchange (PABX). This PABX is connected through telephone lines to the public service telephone network (PSTN) company that provides the telephone service to the call center.

If there are one or more telephone lines free, then the call will be connected to the PABX. Otherwise, the caller receives a busy signal.
If the call is connected to the PABX, the call may be connected through the PABX to an interactive voice response (IVR) that distinguishes the customers based on their needs. Then the call is connected to an automatic call distributor (ACD), a switch that routes calls to the call

center agents. Modern ACD's can be programmed to route calls based on the qualifications of agents. The ACD has access to records that describe whether or not the agents are qualified or "skilled" to serve the different types of calls that arrive at the call center.

Given the type of an arriving call and the status of the idle agents, the ACD routes that call to an idle agent who is skilled to serve that call. If there is no idle agent with the right skill to serve that call, the call may be kept on hold by the ACD and then the customer may wait until a suitable agent becomes idle. Some customers become impatient and decide to abandon (hang up) before they are served. These impatient customers may try to call again.

When the call is connected with an agent, the customer receives the service he needs and then he hangs up. But sometimes it can be the case that the agent cannot completely serve the customer and therefore the call must be routed to another agent.

2.2 Modeling a call center

As we know, queueing is a common every day experience. Queues exist because resources are limited. A typical example is waiting in the supermarket or post office for service. Also in call centers queues can exist: calls that find all the agents busy on arrival are put on hold and then they wait in queue for an agent to become idle. Therefore call centers can be modeled as a set of multi-server queues.

First the notation that is used for describing queueing models will be introduced and then a queueing model for a call center is discussed.

A handy notation to describe queueing models is Kendall's notation. This notation has the following form: *a/b/c/d*. These symbols describe the arrival process, the service time distribution, the number of available agents and the system's capacity, respectively.

Most often used symbols for *a* are:

- M: the arrival process is Poisson (the interarrival time is exponentially distributed),
- D: the interarrival times are deterministic.

And those for *b* are:

- M: the service time is exponentially distributed,
- D: the service time is deterministic,
- G: the service time distribution is general.

If *d* is not specified then it is assumed that the system's capacity is infinite.

The M/M/s queue is the simplest queueing model of a call center and it will be described in the next paragraph. Before describing this model, some queueing characteristics will be specified.

2.2.1 Queueing characteristics

In the previous paragraph I showed how a call center works. This paragraph goes further on the modelling aspect. In this paragraph queueing characteristics, as the arrival process and service time, will be discussed.

The arrival process

Calls arrive randomly at times $t_1, t_2, \ldots, t_{r-1}, t_r$ at the system. The time between two arrivals t_{r-1} and t_r is called the interarrival time. It is assumed that these interarrival times are identical independently distributed. The simplest arrival process is one where the interarrival time is deterministic. The most used arrival process is the Poisson process with intensity λ. By a Poisson process the interarrival times are identical independently distributed and exponentially distributed.

The service time

The service time is the time a call spends in service. The distribution of the service time can be deterministic, exponential or general. The exponential distribution is most used for the service time.

Number of agents

The number of agents is crucial for the service level. First a description of the agents has to be given. This means specifying the number of calls that such an agent can handle and whether or not they have the same skills to handle them. If there are more than two agents it is important to describe whether each agent has a separate queue or if there is only one queue for all agents.

Capacity of the system

Besides the arrival process, service time and the number of agents the maximum number of calls that can stay in both the waiting queue and in service has to be specified. For a given number of agents it means that the bigger the total capacity the more calls can stay in the waiting queue. If the system's capacity is equal to the number of agents (there is no waiting queue) calls finding all the agents busy will be lost. Systems with no waiting queue are also called loss systems.
In systems with a waiting queue the room can be finite or infinite. In the first case only calls that find all agents busy will be lost. In the second case every call will enter the system and waits until he is served.

Queue characteristics

There are different rules by which the next call to be served can be selected. Common disciplines are First in, First out (the service is provided in the order of arrival) and Last in, First out (the service is provided in the reverse order of arrival). Another possibility is to randomly choose the next call. Another characteristic of the waiting queue is whether waiting calls can abandon if they have waited too long for service.

2.3 The basic call center model

As said in the previous paragraph, the M/M/s queue is the simplest queueing model for a call center. This paragraph gives a description of this model and some mathematical notation will be introduced.

The M/M/s queueing model assumes the following:

- calls arrive at the system according a Poisson process with a mean of λ calls per time unit (e.g., per hour, per minute, per second);
- the service time is exponentially distributed with mean $\frac{1}{\mu}$ time units;
- there are *s* agents and there is one waiting queue for all of them;
- the number of calls that can stay in the waiting queue is infinite;
- calls get service in order of arrival.

A schematic model of a basic call center is depicted in Figure 1.

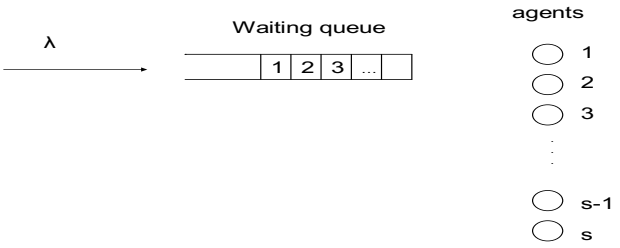


Figure 1: a schematic model of a call center

If a call is connected to the call center and fewer than *s* agents are busy, then the call is connected to one of the idle agents. Otherwise, the call waits in the waiting queue for an idle agent. As said before waiting calls are served in order of arrival.

The offered load is defined as $R := \frac{\lambda}{\mu}$ and the agent's utility is defined as $\rho := \frac{R}{s} = \frac{\lambda}{s * \mu}$.

Thus on average the agents are busy ρ x 100% of the time. If $\rho \geq 1$, then the offered load R exceeds the number of agents *s*. In that case the agents cannot cope with the arriving amount of work and then the number of calls in the system grows to infinity. To ensure that the system is stable it is assumed that $\rho < 1$, the stability condition.

The famous Erlang C formula

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

defines the probability that all agents are simultaneously busy.

A property that is often used in queueing theory is called the Poisson Arrivals See Time Averages (PASTA) property that is only true for Poisson arrivals. This property says that the fraction of customers finding on arrival the system in some state *j* is exactly the same as the fraction of time the system is in state *j*.

Based on the PASTA property something can be said about the fraction of arriving calls that must wait to be served. The fraction of arriving calls that must wait to be served is the same as the fraction of arriving calls that find all *s* agents busy, because customers only have to wait if they find all agents busy.
Because of the PASTA property the fraction of arriving calls that find all *s* agents busy is equal to $C(s,\rho)$, the fraction of time that all agents are busy.

The probability p_j that there are j calls in the system is calculated as [7]:

$$\begin{cases} \frac{(s*\rho)^j}{j!}\frac{s!*(1-\rho)*C(s,\rho)}{(s*\rho)^s}, j = 0,1,...,s-1 \\ \frac{(s*\rho)^s}{s!} * \rho^{j-s} * p_0, j = s, s+1,... \end{cases}.$$

Before giving some expressions for waiting times and queue lengths, the following notation for queue lengths and waiting times will be introduced:

- L: the mean number of calls in the system;
- L_q: the mean length of the waiting queue;
- W: the mean time that a call stays in the system, while waiting and while being served;
- W_q: the mean waiting time of a call in the queue, often referred as the Average Speed of Answer (ASA).

Because there can be calls in the waiting queue or in service, L is equal to the sum of the mean length of the waiting queue L_q and the mean number of calls in the system:

$$L = L_q + s * \rho .$$

The following expression can be given for the mean length of the waiting queue:

$$L_q = \sum_{j=s}^{\infty} (j - s) * p_j = \ldots = \frac{\rho * C(s, \rho)}{1 - \rho}.$$

The mean time that a call stays in the system is equal to the sum of the ASA and the mean service time:

$$W = ASA + \frac{1}{\mu}.$$

Little's formula describes the relation between the measures that are just introduced. It states that $L = \lambda * W$ and $L_q = \lambda * W_q$. With Little's formula and the expression for L_q the following expression for ASA can be given:

$$ASA = \frac{L_q}{\lambda} = \frac{C(s, \rho)}{s * \mu}.$$

A performance measure that is often used for call centers is the servicel level (SL), formulated as the percentage of calls that will be answered in less than *t* seconds:

$$SL = P(W_q \le t) * 100\% = [1 - C(s, \rho) * e^{-s*\mu*(1-\rho)*t}] * 100\% . \quad (2)$$

The industry standard is that 80% of all calls should be answered within 20 seconds, but other

numbers are possible as well [3].
Let SL^* be a lower bound for the acceptable service level. Given SL^* and the formula for SL (2), the minimum number of agents required s^* can be determined by doing trial and error. The goal is to find a number for s^* such that $s^* = \min\{s \mid SL(s) \geq SL^*\}$.

3 Modeling a multi-skill call center

The basic call center model described in the previous chapter is very simple. Blocking and customer impatience are ignored by the model. Furthermore, the model considers a call center with equally skilled agents and homogeneous calls. However, in practice agents may not all be equally trained and call centers handle different types of calls.

In this chapter a multi-skill call center is considered. As we saw at the beginning, agents in a multi-skill call center may not be capable to handle all the different call-types that are offered to the call center. A better solution is to hire agents that can handle a subset of the call-types that are served by the call center. For the ACD this means that it has to be programmed to route calls only to agents that are skilled for serving those calls.

Two types of problems are related to skill-based routing:

1. the agent selection problem: when a particular type of call arrives and there are two or more idle agents, then there has to be decided to which agent the call should be routed;
2. the call selection problem: when an agent becomes idle and one or more calls for which the agent has the required skills are waiting to be served, the agent has to choose which call to serve first.

For skill-based routing each agent is member of an agent group and the groups are characterized by one or more skills that agents within a group have. If a call of a certain type arrives, it is offered to one or more groups having the required skill to serve that call. The order in which this is done is determined by the routing scheme.

Garnet and Mandelbaum [4] introduced a number of designs for multi-skill call centers. Those designs represent building blocks for more complex systems. The six designs are illustrated in Figure 2. A circle represents a group of agents who have the same skills.

For example, in an “I” design a single group of agents handles only one call-type. This design is equivalent to the M/M/s queueing model.

In a “V” design two types of calls are both handled by one single group of agents and the agents are skilled to handle both types of calls. For the call selection problem a rule must be specified for the order in which the agents will handle those two call-types.
For example, type-one customers may be VIP and therefore they have higher priority than type-two customers; if calls of both call-types are in queue and an agent becomes idle, then that agent will “select” a type-one call to serve next.

Further, in an “N”-design two types of calls are handled by two groups of agents: group one only serves type-one calls and group two serves calls of both types.
This type of design can be used when type-one calls are VIP but there are not enough agents in

group one to serve them. So group two helps group one by giving priority to type-one calls over type-two calls. If a type-*i* call finds agents of both groups idle, then that call is routed to group *i*.

In an "X"-design two types of calls can be served by either of two groups of agents. Group one is assigned type-one calls as a primary skill and type-two calls as secondary: if there are waiting type-one calls, then group one gives priority to type-one calls over type-two calls. Otherwise, the group serves type-two calls, if they are.
In this design group two is assigned type-two calls as primary skill and type-one calls as secondary skill.

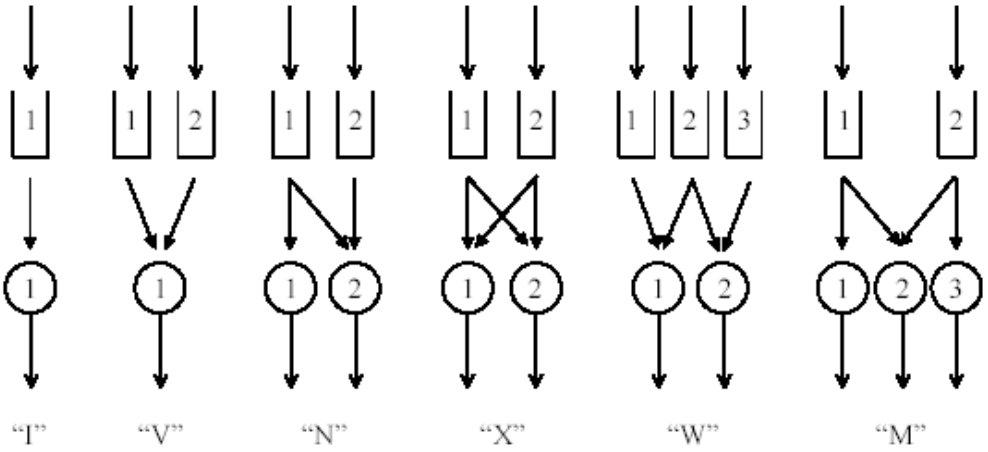


Figure 2: designs for skill-based routing

There are two types of skill-based routing: static routing and dynamic routing. Static routing means that the order in which calls are assigned to groups is fixed and only depends on the call-type. An agent selection rule might be: first agent group 1, then agent group 2 and then agent group 3, etc.
For dynamic routing the order depends on both the call-type and the system's state (e.g., the number of type-i calls that are in service and in queue). For this type of routing an online algorithm determines how a call should be routed.
In the remainder of this chapter the two types of skill-based routing are illustrated.

3.1 A Multi-skill call center using a static routing scheme

The following assumptions are made for the call center:

- there are *C* agents;
- *n* types of calls arrive at the call center according to *n* independent Poisson processes with rates λ_k, $1 \le k \le n$;

– the service time of call-type k is exponentially distributed with mean $\frac{1}{\mu_k}$;

– there is a separate waiting queue with w_k spaces for each call type k and it is assumed that $\sum_{k=1}^{n} w_k = K$.

Further it is assumed that an agent can answer any type-k call if and only if that agent has skill k. Thus an agent can have from one to n skills. Besides skills, agents also have priority levels for those skills. These priority levels determine how calls are routed to agents. Skills with a lower priority level number have a higher priority. Therefore calls, requiring skills with a low priority level number have high priority and thus are handled first. For clearness, Wallace and Whitt [8] suppose that a call center manager is the person who decides at which priority level an agent has a certain skill k.

The skills of agents and the priority levels of those skills are specified via a C x n agent-skill matrix A. The columns represent the priority levels and positive matrix entries represent the skills. Row i of A specifies the skills and their priority levels for the i^{th} agent. If $A_{ij}=k$, then agent i has skill k at priority level j; if $A_{ij}=0$, then agent i has no skill at priority level j; if there is no priority level j for which $A_{ij}=k$ then agent i does not have skill k.

The first column of A specifies the skills with priority level one (the primary skills), the second column specifies the skills with priority level two (the secondary skills), the third column specifies the skills with priority level 3 (the tertiary skills), etc.
Primary skills have the highest priority and it is assumed that every agent has a primary skill, such that $1 \le A_{i1} \le n$ for each agent i.

Further, it is assumed that each agent has at most one skill at any priority level and each agent can have each skill at only one priority level.

For illustration of the agent-skill matrix the following 6x4 matrix A is considered:

$$A = \begin{bmatrix} 3 & 4 & 1 & 0 \\ 1 & 4 & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 3 & 1 & 2 & 4 \\ 1 & 0 & 4 & 0 \end{bmatrix}.$$

The matrix specifies an agent profile for a call center manned by six agents, each having one to four skills. Agent 4 supports only call-type 4, while agent 5 supports all of the call types. Agent 2

can support 2 call-types: call-type 1 at the primary level and call type 4 at the secondary level. Note that agent 6 has a skill at priority level 3 but no skill at priority level 2. The difference between the second and last row of matrix A will become clear when I will discuss the assignment rules.

The agents are grouped by their primary skills, such that group G_k is the subset of all agents with primary skill k. Thus $G_k = \{i : 1 \le i \le C, A_{i,1} = k\}$, $1 \le k \le n$. Looking at the matrix A, four work groups can be distinguished, one for each call type. The work groups can be identified by looking at the first column of the matrix A. Work group G_1, for example, consists of agents 2 and 6, and work group G_2 consists of agent 3.

3.1.1 The agent selection rule

Arriving calls of type k that find one or more idle agents are first routed to idle agents in work group G_k, because those agents have primary skill k. Wallace and Whitt [8] use the longest-idle-routing policy to determine which of the idle agents in work group k will handle the call. This policy sends the call to the agent in work group k that has been idle the longest since the completion of his last call. However, other policies to select an agent could be used instead.
If all agents with call type k as a primary skill are busy, then the call is routed to idle agents having type k as a secondary skill. If all agents with call type k as a primary skill or a secondary skill are busy, then the call is routed to idle agents having call type k as a tertiary skill, and so on. If no available qualified agent can be found to handle the type-k call immediately upon arrival, then the type-k call is placed at the end of the queue associated with call type k.

Now the difference between the second and the last row of the agent-skill matrix A that is just illustrated becomes clear. The second row represents an agent with the skill profile 1400 and the last row represents an agent with skill profile 1040.
Consider the call center to which the 6 x 4 agent-skill matrix belongs. Suppose a type-4 call arrives at the call center. If there are no idle agents with skill 4 as primary skill, then the call is routed to one of the agents with skill 4 as secondary skill. In this case the agent with skill profile 1400 can be among the secondary-skilled agents to handle the call. The arriving type-4 call will only be routed to the agent with skill profile 1040 if there are no idle agents with skill 4 as primary or secondary skill. Thus in this case, the agent with skill profile 1400 has better chance of handling more type-4 calls than the agent with skill profile 1040.

3.1.2 The call selection rule

If an agent becomes idle and if there are no calls for which he has skills in one of the n queues, then the agent stays idle. Otherwise, the agent visits the queues of the call-types for which he has the required skill. The agent visits the queues in increasing order of the agent's priority level; the

agent starts with the queue of the call-type for which he has the lowest priority level number and ends with the queue of the call-type for which he has the highest priority level number.
If there are one or more waiting calls in a waiting queue, then the agent serves the calls in order of their arrival.

3.1.3 Case 1: a multi-skill call center with only specialists

Consider the case where each agent has only one skill. Agents having only one skill are also called specialists [5]. In this case type-*k* calls can only be handled by agents with skill *k* and agents with skill *k* can only handle type-*k* calls. Because each agent has only one skill and it is assumed that each agent has a primary skill, the agents are grouped by their primary skill and therefore the system decomposes into *n* independent $M/M/|G_k|/|G_k|+w_k$ queue's, $|G_k|$ is the number of agents in work group G_k. See figure 3.

Note that the *n* queues do not behave like the basic call center model: in this case the number of waiting spaces is finite, while in the M/M/s queue the number of waiting spaces is infinite.

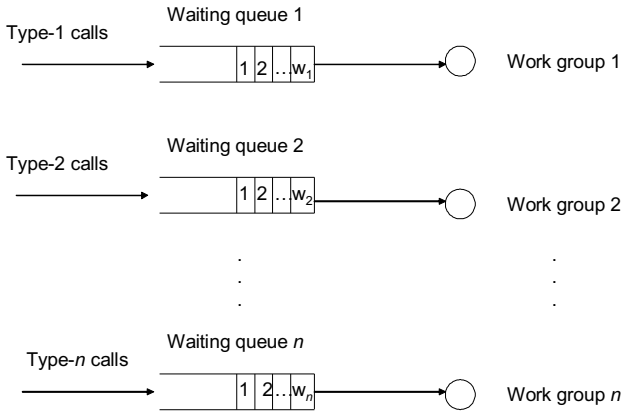


Figure 3: a call center with only specialists

3.1.4 Case 2: a multi-skill call center with only generalists

In this case a call center is considered where each agent has all the *n* skills. Agents having all skills are also called generalists. In this case a call can be handled by every agent and therefore the agents are fully flexible. See figure 4. Note that the number of call-types that an agent can

handle says something about the system's flexibility: a higher number denotes more flexibility.

Suppose a type-*k* call arrives. If there are one or more idle agents, then the call is routed to one of them. The order in which an arriving call is routed to the idle agents is determined by the agent selection rule: the call is first routed to idle agents with skill *k* as primary skill, then to idle agents with skill *k* as secondary skill, then to idle agents with skill *k* as tertiary skill, etc.

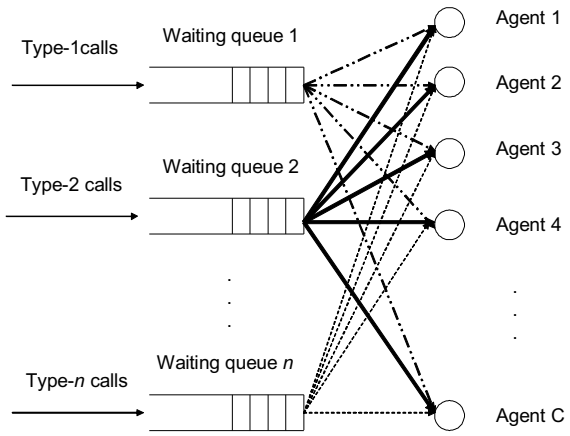


Figure 4: a call center with only generalists

3.1.5 Conclusion

So far I discussed a static routing scheme. In this routing scheme agents have priority levels for the different skills. Those priority levels determine how calls are routed to agents.
Arriving calls of type *k* that find one or more idle agents are first routed to idle agents who have skill *k* at priority level one. If all agents with call-type *k* at priority level one are busy, then the call is routed to idle agents having call-type *k* at priority level two. If all agents with call-type *k* at priority level one or priority level two are busy, then the call is routed to idle agents having call-type *k* at priority level three, etc.
If an agent becomes idle and there are one or more calls for which he has the right skills in the waiting queue, then the call selection problem is solved by visiting the waiting queues of the call-types for which the agent has the required skills. This is done in increasing order of the agent's priority levels; the agent starts with the queue of the call-type for which he has the lowest priority level number and ends with the queue of the call-type for which he has the highest priority level number.

In the previous two paragraphs two special cases of a multi-skill call center are discussed: a call center with only specialists and a call center with only generalists. The second case represents full

flexibility because each call-type can be served by either of the *C* agents. Therefore it is investigated in [8] how many skills agents need in order for the performance to be the same as if all agents would have all skills.

In their simulation experiments Wallace and Whitt considered a call center with 90 agents and 30 available waiting spaces for all six call-types. The number of agents per work group is 15 and the service times are exponentially distributed with mean 10 minutes, e.g., $\frac{1}{\mu_1} = \frac{1}{\mu_2} = ... = \frac{1}{\mu_6} = 10$ minutes. Wallace and Whitt showed that the performance of a call center where each agent has exactly two skills is nearly the same as if each agent would have all six skills.
Table 1 shows the performance for a multi-skill call center with all agents having one, two and all six skills.

# skills per agent	$P(blocking)$	$E(W_q)$	$P(W_q \leq 0.5)$	ρ
1	0.0336	2.85	0.478	0.897
2	0.0044	0.59	0.716	0.928
6	0.0038	0.46	0.781	0.929

Table 1: a comparison of the performances of a call center with agents having all one, two and six skills

3.2 A multi-skill call center using a dynamic routing scheme

In the previous paragraph I illustrated the static form of skill-based routing by describing a static routing scheme. In this paragraph I will give an illustration of a dynamic routing scheme.

The following assumptions are made for the multi-skill call center considered in this paragraph:

- there are C agents;
- n types of calls arrive at the call center according to n independent Poisson processes with rates λ_k, $1 \le k \le n$;
- the service time of call-type k is exponentially distributed with mean $\frac{1}{\mu_k}$;
- there is a separate waiting queue for each call type k.

In this case a call center with M different agent types is considered. Each agent type represents agents having a particular set of skills that an agent can have. This set of skills is also called a skill set and it is defined as $S_i = \{ k \in n \mid \text{agent type } i \text{ has skill } k \}$, $1 \le i \le M$.

The idea is that the C agents are grouped by their skill set; all agents within a group have the same skill set and thus the same skills. So it can be said that there are no groups with the same skill set. The skills of the different groups are represented by an M x n agent-skill matrix A. Let A_{ik} indicates whether or not agents in group i have skill k ($1 \le i \le M, 1 \le k \le n$):
$A_{ik} = 1$ if agents of group i have skill k and $A_{ik} = 0$ otherwise.
The rows of the agent-skill matrix represent the M groups and the columns represent the different skills. For illustration the following 4 x 4-matrix is considered:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

The agents of group one have only skill one, the agents of group two have skills one and three and the agents of the last group have all four skills.

3.2.1 Case 3: a call center with calls served by a private group of agents

Consider the case where each call-type *k* is served by a private group of agents that only handles type-*k* calls. The call center then decomposes into *n* independent M/M/s queues, one for each call-type. See figure 5. The call center looks like a call center with only specialists (see paragraph 3.1.3). The difference is that in this case an agent handles only one skill, while he may be skilled for handling several call-types.

Denote by C_k the minimum required number of agents with skill *k*, $1 \le k \le n$. This number can be calculated with the Erlang C formula (1), given the service level constraints for call-type *k*. The number C_k plays an import role in both the agent selection rule and the call selection rule.

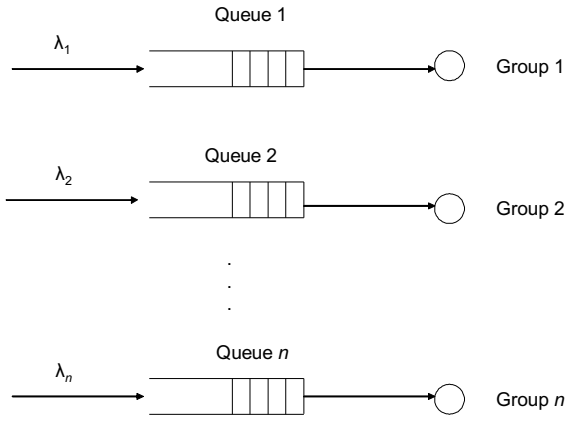


Figure 5: a call center with each call-type being served by a private group of agents

3.2.2 The agent selection rule

The idea of the agent selection rule is that each agent type *i* is assigned a value P_i. This value can be seen as the contribution of type-*i* agents to an arbitrary type of call. The higher the value P_i, the higher the contribution of a type-*i* agent.

The value P_i of type-*i* agents is defined as:

$$P_i = \sum_{k=1}^{n} A_{ik} * Q_k \qquad (1 \le i \le M),$$

with $Q_k := \dfrac{C_k}{\sum_{i=1}^{M} A_{ik} * B_i}$ and B_i, the total number of type-*i* agents.

Looking at the expression for Q_k, the denumerator $\sum_{i=1}^{M} A_{ik} * B_i$ is equal to the total available number of agents with skill *k*. So, Q_k is the minimum required number of agents with skill *k* as a fraction of the total available number of agents with skill *k*.

Table 2 shows an agent-skill matrix and the value P_i for three groups of agents. Looking at that table you can see that group 2 has a higher value for P_i and therefore it is more valuable (or flexible) than group 1, while group 3 is more valuable than group 2.
Thus, if there are two groups *i* and *j* with skill sets S_i and S_j respectively and $S_i \subset S_j$, then it can be said that group *j* is more valuable or flexible than group *i*.

	Skill 1	**Skill 2**	**Skill 3**	**P_i**
Group 1	1	0	0	Q_1
Group 2	1	1	0	$Q_1 + Q_2$
Group 3	1	1	1	$Q_1 + Q_2 + Q_3$

Table 2: the agent-skill matrix and value P_i for three agent groups

Suppose a type-*k* call arrives and there are idle agents with skill *k*. Then the call is routed to an agent within the group with the minimum value of P_i among all the groups having skill *k*.
If there are no idle agents with skill *k* upon arrival of a type-*k* call, then the call is put in the waiting queue.

3.2.3 The call selection rule

The idea of the call selection rule is that each call-type is assigned a specific amount of credit. The call selection rule ensures that each call-type receives a service level that is equal to or better than it would receive when served by a private group of agents. Therefore, the number of type-*k* calls that are in service at time *t* is compared to the nominal capacity C_k.

The amount of credit U_k for call-type *k* is defined as:

$$U_k(t) := \frac{C_k - N_k(t)}{C_k - \rho_k}, \tag{3}$$

where $N_k(t)$ is the number of type-*k* calls that are in service at time *t*.

In the description of the M/M/s queue I illustrated that because of the system's stability the offered load ρ doesn't exceed the number of agents *s*. So for each call-type *k* this means that $C_k > \rho_k$, and thus the denominator in (3) is always positive. The credit U_k can be positive or negative depending on whether or not $N_k(t)$ is exceeding the nominal capacity C_k.
If U_k is positive, then call-type *k* receives a less service as if it would be served by a private group of agents, because for the last case there are C_k agents needed to meet the given service level constraints. A negative value of U_k means that $N_k(t)$ exceeds C_k, what means that call-type *k* receives more service than if it would be served by a private group of agents.
Now, the call selection rule can be described. Suppose a type-i agent becomes idle. If there are no waiting calls for which the agent has the right skill, then the agent stays idle. If there are only type-*k* calls in queue and the agent has skill *k*, then the first arrived type-*k* call will be selected to serve next. If there are waiting calls of different types for which the agent has the right skill, then a call from the call-type with the minimum amount of credit among those call-types is routed to the agent.
So, the call selection rule is:

An agent of group i that becomes idle selects a call from the call-type with maximum amount of credit among the call-types k with waiting calls and $A_{ik} = 1$.

3.2.4 Conclusion

In the described routing scheme agents are divided in *M* groups, such that all agents within a group have the same skills to serve the same call-types. Those different groups are assigned a value that can be seen as the contribution to an arbitrary type of call. The agent selection problem is solved by routing an arriving type-*k* call to an idle agent within the group with the minimum value among all the groups having that skill *k*. If there are no idle agents with skill *k* upon arrival of a type-*k* call, then the call is put in the waiting queue.
An import aspect of the dynamic routing scheme is that the call selection rule ensures that each call-type receives a service level that is equal to or better than it would have experienced when served by a private group of agents. This is done by comparing the number of calls of each call-type that is in service to the number of agents that would be minimal needed if each call-type

would be served by a private group of agents.
In contrary to the static routing scheme the call selection rule depends on both, the traffic parameters (the arrival and service rate) and the number of calls that are in service. Fluctuations in the arrival and service rates may have an undesired impact on the service level. Therefore accurate estimations of the traffic parameters are required.

3.3 Comparison

In the previous paragraphs I gave a brief illustration of the two types of skill-based routing (static and dynamic routing) by describing both a static routing scheme and a dynamic routing scheme.

The static routing scheme uses priorities (skill levels) that determine how calls are routed to agents. For each agent it is determined via an agent-skill matrix which skills he has and each of those skills are assigned a skill level number. Skills with low skill level numbers have higher priority than skills with high skill level numbers. For the solution of the call selection problem this means that agents give priority to call-types requiring skills with low skill level numbers over call-types requiring skills with high skill level numbers.

The use of those priorities can be very handy. When an agent acquires a new skill, then the call center manager can decide to give that new acquired skill a low priority such that (relatively) a few calls of the call-type that requires the new skill are routed to the agent. The priority of the new acquired skill may be changed after some experience is gained and then more calls may be routed to the agent.

The static routing scheme is not very complex, in comparison to the dynamic routing scheme: the priorities are fixed and how the priorities are calculated is not taken into account by the static routing scheme.

The dynamic routing scheme also makes use of priorities, but then in another way as the static routing scheme does. For the solution of the call selection problem each call-type is assigned an amount of credit. The higher the amount of credit, the higher priority that type of call has. The amount of credit depends on the number of calls that is service. The number of calls that are in service changes over time and therefore the amount of credit and thus also the priorities are changing over time.

In contrary to the static routing scheme the dynamic routing scheme makes use of the reservation of flexible agents. Suppose there are two agent group *i* and *j* with skill sets S_i and S_j respectively, and $S_i \subset S_j$. As said before, agent group *j* is more flexible than agent group *i*. Further, suppose that an arriving call can be handled by either of those two agent groups and that there are idle agents within both groups. Then the routing scheme ensures that the arriving call is routed to group *i*, such that the idle agents within group *j* are reserved.

4 Conclusion

The basic call center is a call center at a single location with identical agents who handle only homogeneous inbound calls. For given arrival rates, service times and service level requirements, the minimum number of agents needed may then be determined using the famous Erlang formula.

In practice call centers serve heterogeneous calls requiring different skills and agents may have different skills determining which call-types they can serve. In such more complex call centers extra attention has to be paid to the routing mechanism and skill-based routing becomes a necessity. The goal of skill-based routing is to route incoming calls in an intelligent way, in order to achieve a high service level and flexibility against low costs.

Two types of problems are related with skill-based routing:

- the agent selection problem: when a particular type of call arrives and there are two or more idle agents, then there has to be decided to which agent the call should be routed;
- the call selection problem: when an agent becomes idle and one or more calls for which the agent has the required skills are waiting to be served, the agent has to choose which call to serve first.

The main idea of skill-based routing is that the agents are divided into groups, such that all agents within a group can serve the same call-types. For each type of call there exists an ordered list of groups with agents having the right skill to handle calls of that type. The agent selection problem is solved by routing an arriving call to the first group in the list that has an idle agent.
If there are no idle agents for handling that call, then the call is put in the waiting queue.
For each agent group there is a list of call-types for which the agents within that group are skilled. When an agent becomes idle and there are waiting calls for which the agent is skilled, then the call of the type with the highest priority is routed to the agent.

There are two types of skill-based routing: static routing and dynamic routing. Static routing means that the order in which calls are assigned to groups is fixed and only depends on the call-type. By dynamic routing the order depends on both the call-type and the system's state (e.g., the number of type-i calls that are in service and in queue). By this type of routing an online algorithm determines how a call should be routed.

In this paper I have illustrated the two types of skill-based routing. For the illustration of the two types of skill-based routing I gave a brief description of a relatively simple static routing scheme and a more complex dynamic routing scheme.

Besides the routing of calls, the staffing problem becomes very complex. Skill-based routing makes it difficult to determine how many agents with which skills are needed to meet given service level requirements: the famous Erlang C formula is not accurate anymore when agent agents have multiple skills.

Skill-based routing is receiving increasing attention. Note that the benefit of skill-based routing depends on how it is implemented. Different ways of routing calls will imply different performances.

5 Literature

[1] Aksin, O.Z. and Karaesmen, F. (2002), *Designing Flexibility: Characterizing the Value of Cross-Training Practices,* http://home.ku.edu.tr/~fkaraesmen/pdfs/flex2102.pdf

[2] Borst, S. and Seri, P. (2000), *Robust Algorithms for Sharing Agents with Multiple Skills*, Working paper, Bell Laboratories.

[3] Gans, N. , Koole, G. and Mandelbaum, A. (2003), *Telephone Call Centers: Turorial, Review and Research Prospects*, Manufacturing & Service Operations Management, vol. 5, no. 2.

[4] Garnett, O. and Mandelbaum (2000), *An introduction to skills-based routing and its operational complexities*, Teaching note, Technion,

http://fic.wharton.upenn.edu/fic/f0503mandelbaum.pdf

[5] Koole, G., Pot, A. and Talim, J. *Routing heuristics for multi-skill call centers*, Proceedings of the 2003 Winter Simulation Conference.

[6] Koole, G. (2004), *Call center mathematics*, http://www.math.vu.nl/~koole/ccmath/book.pdf

[7] Tijms, H.C. (2002), *Stochastische methoden en simulatie voor BWI* College dictaat

[8] Wallace, R. and Whitt, W. (2004), *A staffing algorithm for call centers with skill-based routing*, http://pages.stern.nyu.edu/~gjanakir/WhittPaper.pdf