Manitz, Michael; Piehl, Marc-Philip

A fast staffing algorithm for multistage call centers with impatient customers and time-dependent overflow

Central European Journal of Operations Research

Provided in Cooperation with:
Springer Nature

Suggested Citation: Manitz, Michael; Piehl, Marc-Philip (2023) : A fast staffing algorithm for multistage call centers with impatient customers and time-dependent overflow, Central European Journal of Operations Research, ISSN 1613-9178, Springer, Berlin, Heidelberg, Vol. 32, Iss. 3, pp. 763-791,
https://doi.org/10.1007/s10100-023-00883-z

This Version is available at:
https://hdl.handle.net/10419/306379

Central European Journal of Operations Research (2024) 32:763–791
https://doi.org/10.1007/s10100-023-00883-z

ORIGINAL PAPER

A fast staffing algorithm for multistage call centers with impatient customers and time-dependent overflow

Michael Manitz[1] · **Marc-Philip Piehl**[1]

Accepted: 5 September 2023 / Published online: 13 October 2023

Abstract
Ensuring customer satisfaction is one of the main objectives of a call center. We focus on the question of how many agents are necessary and how they should be allocated to maintain a service level threshold and reduce the expected waiting time of the customers. In this paper, we consider a multistage call center that consists of a front and a back office, impatient customers, and an overflow mechanism. Based on the performance evaluation of such a system using a continuous-time Markov chain, a configuration of agents is determined using a binary search algorithm. We focus on structural insights, e.g., convexity conditions, to obtain a quick solution for the staffing problem. Since monotonicity does not always hold, the approach is heuristic. The numerical results show the performance of the algorithm. The influence of the fraction requiring second-level service in the back office and the impatience rate for the minimum number of agents is shown.

Keywords Call center · Queueing · Impatient customer · Staffing

1 Introduction

A call center is usually organized in multiple stages where most customers are served in the front office, but a fraction need additional service in the back office. The customers arrive at the call center randomly over time, the service time expended by agents is stochastic, and the patience time of the waiting customers is random. Due to this stochasticity, it is difficult to predict the number of agents needed. Having too

✉ Marc-Philip Piehl
marc.piehl@uni-due.de

Michael Manitz
michael.manitz@uni-due.de

[1] Chair of Production and Supply Chain Management, Mercator School of Management, University of Duisburg/Essen, Duisburg, Germany

few agents can lead to unsatisfying service, such as a long wait time. If a caller's patience is exceeded, the call is lost due to abandonment. Therefore, the minimum number of agents and their allocation to the front and back office is determined to meet one or more performance measures.

To the best of our knowledge, this paper is the first to solve the staffing problem in a serial call center with an overflow mechanism based on a waiting-time threshold, impatient customers and a limited capacity. We show that the monotonicity of the performance indicators does not always hold and propose a heuristic approach that addresses the problem.

The aim of the work is to develop a fast algorithm that uses these effects to determine the minimum number of agents. Due to the fast computation time, different scenarios can be analyzed, such as different values for the required performance measures. This algorithm can also be applied to other performance measures not mentioned in this paper.

In Sect. 2, the analyzed serial system is presented, and the relevant performance measures used in the optimization problem introduced in Sect. 5 are described. Section 3 provides an overview of the literature, and Sect. 4 explains the influence of agent allocation on the service measures. In Sect. 6, we present the solution methodology based on applying a double binary search. The numerical results of the staffing algorithm are shown in Sect. 7 for one period under the assumption that the system is in a steady state. Finally, Sect. 8 summarizes our conclusions and offers suggestions for further research.

2 Problem description

We analyze a two-stage call center consisting of a front office and a back office with a time-dependent overflow mechanism and impatient customers. This is an equivalent queuing system as presented in Stolletz and Manitz (2013), see Fig. 1.

The customers arrive at the front office according to a Poisson call-arrival process with an arrival rate λ_F to receive first-level service by a front office agent. The front office consists of C_F agents, and C_B agents are working in the back office. A

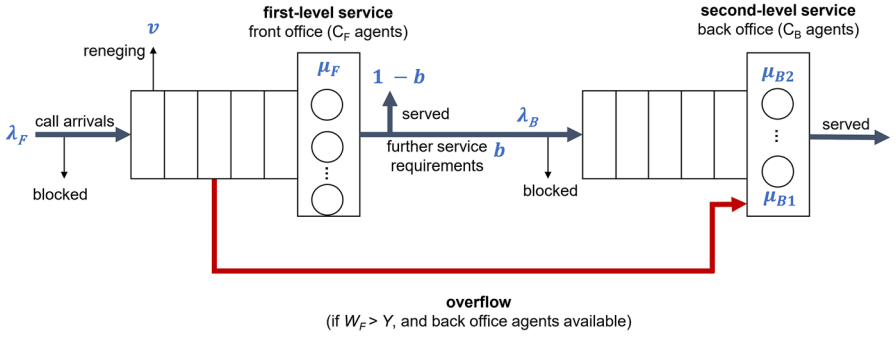

Fig. 1 Serial call center with time-dependent overflow and impatience

fraction b of these calls need second-level service and are routed to the back office with arrival rate $\lambda_B = \lambda_F \cdot b$. If all agents C_F or C_B are busy, the next customer enters a queue. The call center is considered to be a loss-delay system. The length of the queue in terms of the number of simultaneous calls that are in the system is limited by the number of trunks. It is possible to determine the size of the infrastructure, but that is beyond the scope of this paper. The capacities of the front office and back office are limited to K_F and K_B customers. Hence, the number of waiting positions is also limited. A customer is queued if all agents are busy and there is at least one free waiting position. A customer gets blocked if either K_F or K_B or both are exhausted. If the waiting time W_F in the front office exceeds a threshold t and at least one back office agent is available, the customer flows over to the back office. An additional queue for overflow calls does not exist, which results in relief for the front office queue. An overflow is not possible if no agent C_B is available and the customers must wait longer than t minutes. A customer leaves the system if the service is finished in the front office with $(1 - b)$ or in the back office or because of limited patience in the front office queue without service. The limited patience is assumed to be exponentially distributed with rate ν. If the capacity of the back office K_B is exhausted, the customer is blocked and leaves the system without second-level service. The service times are exponentially distributed random variables. Here, μ_F denotes the service rate of agent C_F in the front office. The service rate of agent C_B in the back office is μ_{B1} in the case of an overflow call and μ_{B2} in the case of second-level support.

For the performance evaluation, the queueing system is modeled as a continuous-time Markov chain (CTMC), as presented in Stolletz and Manitz (2013). The Markov property is satisfied by approximating the overflow rule with a fixed threshold t for the waiting time using a direct overflow. For simplicity for the Markov property, we set the threshold t equal to the service level requirement Y because we can then measure the amount of overflow. $P_n^{(Y)}$ represents the probability that an arriving customer overflows immediately and depends on the number n of customers queueing in the line ahead of the arriving customer and on Y. We use the queue-length based overflow (QLBO) for $P_n^{(Y)}$ rather than the waiting-time based overflow (WLBO) because there is no relevant difference in the results, as mentioned in Stolletz and Manitz (2013). In this approach, an overflow to the back office is possible if the expected number of customers in queue $\bar{n}$ that can be served during Y minutes is reached. For further information, see Stolletz and Manitz (2013). The determination of the performance measures is presented in Barth et al. (2010), and the extension made for impatience customers is presented in Stolletz and Manitz (2013). We focus on the performance measures that are influenced by the threshold of the waiting time t, the X/Y service level with $Y = t$, and the expected waiting time. We define the service level for all calling customers as follows:

The probability X that a calling customer receives service in a time span of Y minutes is called the X/Y service level, which is the product of the conditional service level as described in Barth et al. (2010), and the counter probability that a calling customer is blocked $(1 - P(blocking))$ and the counter probability that a randomly selected customer eventually reneges $(1 - P(reneging))$. The service level refers to the front office because all customers first arrive in the front office or join the front office queue. Therefore, only the waiting time of the front office

customers is accounted for in the service level. Similar to Stolletz and Manitz (2013), reneging is also only considered for customers that enter the front office queue and are then potentially served or routed to the back office by overflow. Therefore, we define the X/Y service level for all calling customers with respect to the blocking probability $P(blocking_F)$ of the front office customers as follows:

$$SL_F = P(W_F \leq Y \mid served) \cdot (1 - P(reneging)) \cdot (1 - P(blocking_F)). \tag{1}$$

We see one possibility for the indirect consideration of back office customers at the service level in the blocking probability. The blocking probability is divided into the blocking probability of front office $P(blocking_F)$ and back office customers $P(blocking_B)$. The number of calling customers in the front office is usually significantly higher than the number of back office customers. For this reason, it makes sense to weight the probability of blocking with the arrival rate. The larger the fraction of the arrival rate in the total rate $\lambda_g = \lambda_F + \lambda_{eff_B}$, the more significant it is to consider the corresponding weighted blocking probability. Thus, the weighted blocking probability $P(blocking_{F+B})$ is defined as:

$$\begin{aligned} P(blocking_{F+B}) &= \frac{\sum_{g \in G} \lambda_g \cdot P(blocking_g)}{\sum_{g \in G} \lambda_g} \\ &= \frac{\lambda_F \cdot P(blocking_F) + \lambda_{eff_B} \cdot P(blocking_B)}{\lambda_F + \lambda_{eff_B}} \end{aligned} \tag{2}$$

with

$$P(blocking_B) = \sum_{n_F=0}^{K_F} \sum_{n_{B_1}=0}^{c_B} P\{n_F, n_{B_1}, K_B - n_{B_1}\}. \tag{3}$$

Hence, we define the X/Y service level for all calling customers with regard to front and back office customers as follows:

$$SL_{F+B} = P(W_F \leq Y \mid served) \cdot (1 - P(reneging)) \cdot (1 - P(blocking_{F+B})). \tag{4}$$

Using only the service level as a performance measure may result in a high waiting time for customers waiting longer than Y. In our numerical experiments, we observe that in addition, the performance in the back office could be low if only the service level is used. For this reason, we consider the expected waiting time of a customer in the front office (served and reneged) as a second performance measure in addition to the service level, which is defined as follows:

$$E[W_F] = \frac{E[Q_F]}{\lambda_{eff_F}} + \pi \cdot Y. \tag{5}$$

π describes the probability that an arriving customer is routed directly to a back office agent by overflow. For more details, see Stolletz and Manitz (2013).

The mean waiting time for back office customers is:

$$E[W_B] = \frac{E[Q_B]}{\lambda_{eff_B}}. \tag{6}$$

$E[Q_F]$ and $E[Q_B]$ describe the expected queue lengths in the front and back offices, respectively. The rate at which a call is not blocked is described by λ_{eff_F} in the front office and λ_{eff_B} in the back office; see Barth et al. (2010). The total expected waiting time is then the sum of (5) and (6). Similar to the service level, a weighted average of the expected waiting time is relevant because the numbers of customers and agents in the front and back offices are different. Thus, we define:

$$\begin{aligned} E[W_{F+B}] &= \frac{\sum_{g\in G} \lambda_g \cdot E[W_g]}{\sum_{g\in G} \lambda_g} = \frac{\lambda_F \cdot E[W_F] + \lambda_{eff_B} \cdot E[W_B]}{\lambda_F + \lambda_{eff_B}} \\ &= \frac{\lambda_F \cdot E[W_F] + E[Q_B]}{\lambda_F + \lambda_{eff_B}}. \end{aligned} \tag{7}$$

Other possible disaggregated service measures can be considered. In the first step, we focus on an aggregated service measure comprising all relevant aspects.

3 Literature

The literature on the topic of call center management is vast. For an overview of research on operational call center issues, see Aksin et al. (2007), Gans et al. (2003), Grossman et al. (2001), and Pinedo et al. (2000). Stolletz (2003) and Koole and Mandelbaum (2002) provide a review of the literature on various queueing models in the context of call center modeling. Liao et al. (2012), Chevalier and van den Schrieck (2008), Pot et al. (2008), Wallace and Whitt (2005) discuss related analytical approaches to staffing in call centers.

There are two parts to our problem. First, we evaluate the system behavior of the call center and thus determine performance measures, which in this case are the service level and the waiting time. This is used as input to our optimization problem in which the staffing requirements are minimized. In this chapter, we therefore first compare evaluation models and then optimization approaches. We choose only publications that analyze call centers that consider either a single-stage or multistage system and an overflow mechanism with a fixed value for the waiting time or no overflow. For this, we use the following classification scheme:

- **Performance evaluation method:** The system behavior can be analyzed in two ways, by simulation or analytically. For the papers considered, the performance evaluation is analytical. An example of evaluation by simulation is provided in Wallace and Whitt (2005).
- **System:** The system design can be a single-stage (sis) or multistage (ms) system. In this case, a multistage system is a serial system with two stages.

- **Capacity:** The total number of customers in the system is infinite (∞) as well as finite according to the number of agents (C) or the limit of capacity (K, with $K > C$).
- **Abandonments:** Customers abandon due to impatience (Y = Yes, N = No).
- **Overflow:** In the literature, there exist three types of overflows in call centers. A state-dependent overflow depends on the number of customers in the system. Furthermore, it can depend on a random threshold value for the waiting time. Another possibility is an overflow that depends on a fixed value for the waiting time. In the selected papers, only such an overflow mechanism is considered. More detailed descriptions can be found in Stolletz and Manitz (2013) and the references therein. An overflow mechanism occurs (Y = Yes, N = No).
- **Staffing:** The evaluation is used to solve a staffing model (Y = Yes, N = No).

Staffing method:

- **Objective function:** In the selected articles, the following two objective functions are found. First, the minimization of the total costs of agents, and second, the minimization of the number of agents.
- **Constraints:** In addition to the most common measure, the service level (SL), the expected waiting time (EW) are used as performance constraints in the articles studied.

All approaches under consideration assume a Poisson call-arrival process, exponentially distributed service times and consider multiple parallel servers ($C > 1$).

3.1 Performance evaluation

Table 1 summarizes the articles that are considered to relate to classification by performance evaluation.

Kim and Park (2010) considered a two-stage call center and proposed an analytical solution on the basis of queueing theory. To solve a staffing problem, this approach is applied. The capacity of the call center is limited to K. However, no overflow or reneging is considered.

Table 1 Classification by performance evaluation

Reference	System	Capacity	Abandonments	Overflow	Staffing
Barth et al. (2010)	ms	K	N	Y	N
Kim and Park (2010)	ms	K	N	N	Y
Bekker et al. (2011)	sis	∞	N	Y	N
Koole et al. (2012)	sis	∞	N	Y	N
Stolletz and Manitz (2013)	ms	K	Y	Y	N
Koole et al. (2015)	sis	∞	N	Y	N
This paper	ms	K	Y	Y	Y

Bekker et al. (2011), Koole et al. (2012) and Koole et al. (2015) analyzed an overflow mechanism with a fixed threshold by using a CTMC. In this approach, Koole et al. (2012) and Koole et al. (2015) use an Erlang approximation to model the waiting time of the first customer in the queue. However, they consider an infinite single-stage system with parallel queues. They do not consider abandonments, and no use of the evaluation is applied to solve a staffing problem.

For the performance evaluation, we used an analytical approach, and as a baseline for our analysis, we used the call center system first introduced in Barth et al. (2010) and extended by Stolletz and Manitz (2013). Therefore, the evaluation is performed by using a CTMC. We consider a serial call center with impatience and an overflow mechanism with a fixed threshold on the waiting time and back-office agent availability. We further consider a finite system in which the next calling customer is blocked when all slots in the queue are occupied. Our contribution integrates this performance evaluation when solving staffing problems using a heuristic approach.

3.2 Optimization methods

Kim and Park (2010) solved the staffing problem by using numerical tests. Their objective is to minimize the total costs of agents under the condition that the "80/20 standard service level" must be fulfilled, i.e., that 80 % of the calling customers are served within 20 s.

In this paper, we minimize the total number of agents under the condition that the service level and the expected waiting time must be fulfilled. We use an algorithm based on binary search to solve the staffing problem.

An overview of routing and staffing algorithms in multi-skill call centers can be found in Koole and Pot (2006). As mentioned in Koole and Pot (2006) and outlined in Koole and van der Sluis (2003), a staffing algorithm using local search is efficient when it is assumed that the service level is concave with respect to the minimum number of agents. Since we consider the impatience of callers, the service level can no longer be concave. Contrary to Koole and van der Sluis (2003), we consider a serial call center and solve the staffing problem for only one period. Therefore, we do not consider constraints on global performance measures. Our staffing algorithm can be used to extend the staffing problem to several periods, e.g., one day. In this case, the resulting problem can be solved using a local search. Therefore, it is important to study the properties of the performance measures.

4 Influence of agent allocation on performance measures

In this section, we provide insights into how agent allocation influences service measures. For that purpose, we use an example with corresponding parameters $K_F = 50, K_B = 20, Y = 8, \nu = 0.1, \mu_F = 0.25, \mu_{B1} = 0.2, \mu_{B2} = 0.125, \lambda_F = 10, b = 0.6,$ so $\lambda_B = 6$. Due to the high arrival rate and fraction b, the call center is overloaded.

4.1 Service level

When considering the service level that only considers the blocking probability of the front office customers SL_F, the function shows the typical S-shape. The S-shaped curve of this service level is a typical observation and was already noted by Henderson and Mason (1998); it implies that having only a few agents in service results in low service. By adding an additional agent, the service does not improve strongly. After a certain point, the service increases more. If the number of agents is high enough, then an increase has only a small impact on the service. This phenomenon implies that for each allocation, the service level increases monotonically.

In our numerical experiments, the monotonicity of SL_F and SL_{F+B} in C^{tot} does not hold if the proportion b of callers requiring second-level service is too high, for instance, if $b = 1$. For this reason, we analyze its influence in Sect. 7.2.1.

Related to the previous example, the service level SL_F is not always monotonic for $b = 1$. In this case, for example, with $C_F = 1$ and increasing C_B, SL_F increases monotonically. On the other hand, for $C_B = 1$, SL_F first increases and then decreases from $C_F = 1$ to $C_F = 17$. After that, SL_F decreases again until $C_F = 50$.

The service level SL_{F+B}, which accounts for the weighted blocking probabilities, can be found in Fig. 2.

It can be observed that the course is similar to an S-shape, and if, for instance, $C_F = 5$ and C_B are running, the service level initially increases monotonically in C^{tot}. From $C_B = 17$, however, the service level decreases again. With $C_F = 17$ and increasing C_B, the service level increases monotonically all the time.

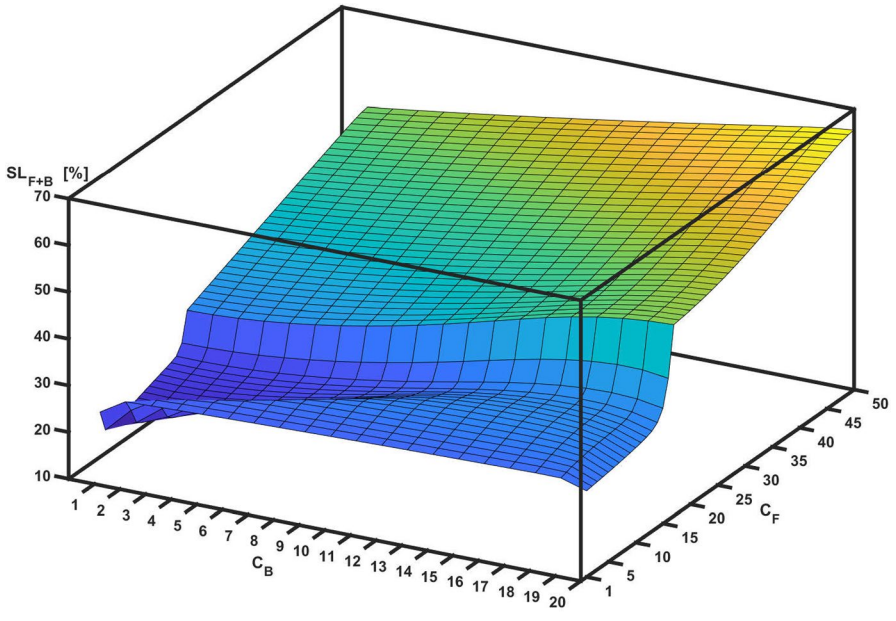

Fig. 2 SL_{F+B} as a function of the number of agents C_F and C_B

For a fixed $C_B = 2$ and running C_F, the service level initially decreases from $C_F = 1$ to $C_F = 5$ and then increases for $C_F = 6$ to $C_F = 50$. The service level increases monotonically from $C_B = 16$ to $C_B = 20$ and C_F running. It can be clearly seen from the figure that the service level function has concave and nonconcave areas. Furthermore, it can be observed that monotonicity does not hold in all cases, as one might expect with increasing C^{tot}.

For further detail, we would like to give the following example, which reflects the observations from our numerical experiments, to demonstrate the reasons for a decrease in the service level even when the total number of agents C^{tot} increased:

We apply the same instance that was mentioned before for $C_B = 10$ and $C_F = 1$ to 20. Figure 3 shows the progression of the service level SL_F and SL_{F+B}, the blocking probabilities for the front and back office, and the weighted blocking probability. Furthermore, the reneging probability is plotted.

For $C_F = 7, C_B = 10$, the service level is $SL_F = 26.17$ % and $SL_{F+B} = 31.75$ %. The reneging probability $P(reneging)$ is 41.46 %. For the blocking probabilities, we obtain $P(blocking_F) = 38.03$ % and $P(blocking_B) = 2.29$ %. As a result, the weighted blocking probability is 24.68 %. Due to the high load, the utilization in the front office is 100 %, and in the back office, it is 98.31 %.

Increasing C^{tot} by 1, i.e., $C_F = 8, C_B = 10$, SL_F increases to 26.50 %. SL_{F+B}, on the other hand, decreases to 31.58 %. $P(reneging)$ decreases to 40.49 %. The mean number of calls $E[N]$ increases from 62.50 to 63.30, of which the mean number of customers in the front office $E[N_F]$ increases slightly from 48.46 to 48.49. The

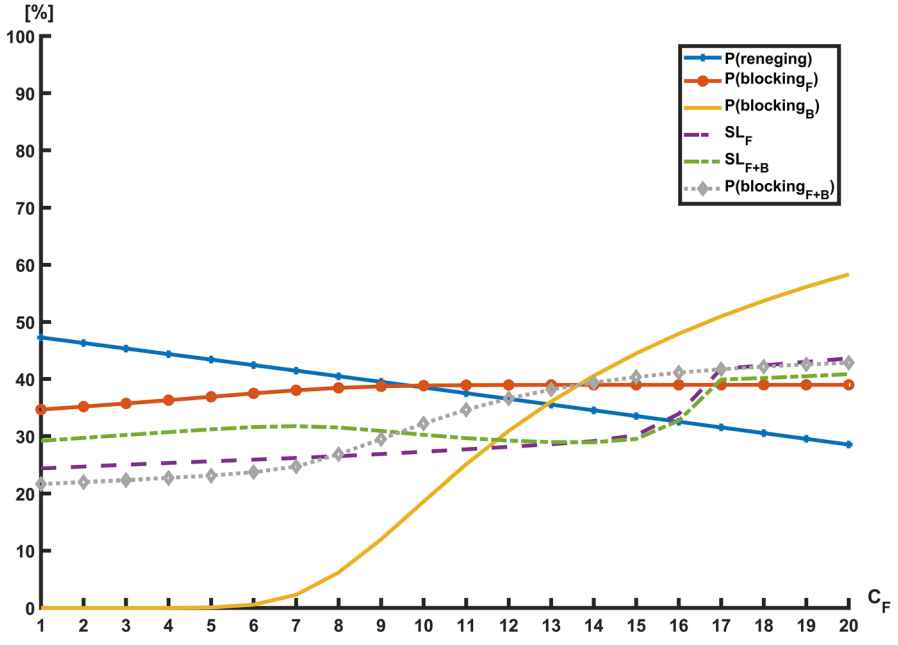

Fig. 3 Comparison of SL_F, SL_{F+B} and reneging and blocking probabilities as a function of the number of agents C_F and $C_B = 10$

mean number of customers in the back office with second-level service requirements $E[N_{B_2}]$ increases from 10.80 to 13.03. Front office utilization remains at 100 %. In the back office, utilization increases to 99.08 %. As a result, $P(blocking_F)$ also increases slightly to 38.45 %. In the back office, $P(blocking_B)$ increases to 6.17 %. As a result, the weighted blocking probability $P(blocking_{F+B})$ is 26.82 %.

The service level SL_{F+B} initially increases. While $P(blocking_{F+B})$ increases slightly, $P(reneging)$ decreases. As the blocking probability in the back office $P(blocking_B)$ increases more strongly, $P(blocking_{F+B})$ also increases more strongly. As a result, SL_{F+B} decreases.

The jump from $C_F = 16$ to $C_F = 17$ can be explained as the probability that a calling customer will wait at least Y time units $P(W_F > Y)$ drops from 17.68 to 0 %, which occurs when the number of customers served within Y exceeds the average number of customers in the front office queue ($\bar{n} > n$). As a result, the probability $P(W_F \leq Y \mid served)$ that a customer waits at most Y time units under the condition that he is eventually served increases since $P(W_F \leq Y \mid served) = 1 - P(W_F > Y)$. As $P(reneging)$ decreases and although the blocking probability increases, SL_F and SL_{F+B} increase as well.

We would like to note that the monotonicity of SL_F holds when considering realistic scenarios. We consider a call center from the financial sector as it is used in Barth et al. (2010). Here, it is a realistic assumption that 10 % of customers need additional service in the back office.

4.2 Expected waiting time

Figure 4 shows the curve of the weighted expected waiting time $E[W_{F+B}]$ relating to the example. For $C_B = 1$ and C_F running, the waiting time increases significantly initially. Here, the waiting time in the front office decreases with increasing C_F. However, more customers are routed to the back office, which increases the waiting time for customers there. After that, $E[W_{F+B}]$ decreases again. For instance, with $C_F = 1$ and increasing C_B, $E[W_{F+B}]$ initially decreases. From $C_B = 5$, the waiting time increases again. In this case, we observe a reverse effect since the waiting time in the front office increases and the waiting time for customers in the back office decreases. As the number of agents C^{tot} increases, it can be observed that the waiting time function has areas where convexity does not hold. Notably, monotonicity does not hold in all cases.

The nonmonotonicity complicates the finding of an optimal solution. We propose a heuristic that determines a feasible solution in Sect. 6.

5 Optimization model

We propose a decision support model that quantifies the minimum number of agents and their allocation across both offices while meeting a given X/Y level of service and a maximum value for the expected waiting time. This problem is similar to the buffer allocation problem in which a decision is made about the buffer capacities and their

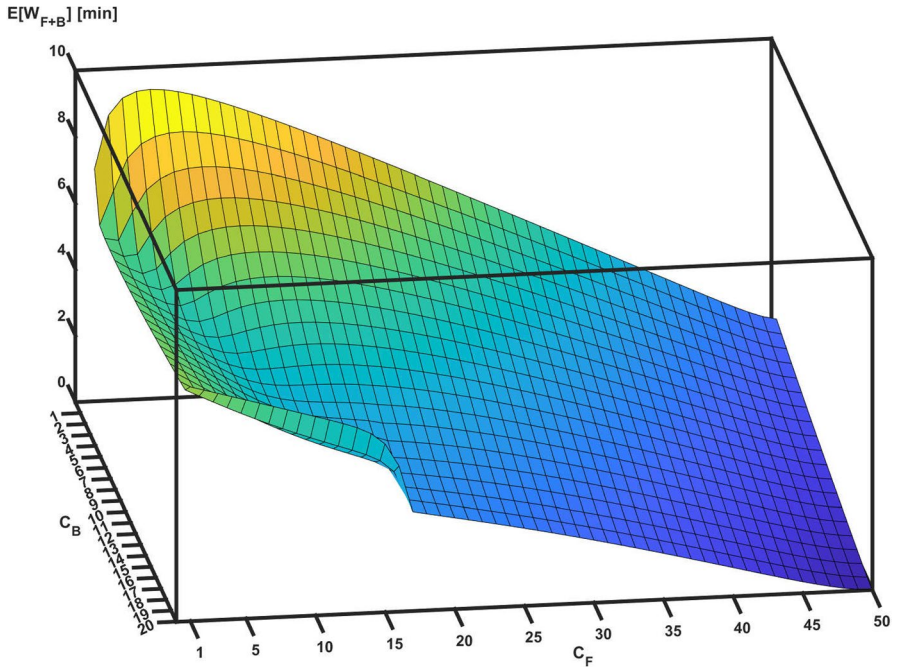

Fig. 4 $E[W_{F+B}]$ as a function of the number of agents C_F and C_B

allocation; see, for example, Papadopoulos et al. (2009). We use the idea presented in Gershwin and Schor (2000) to solve the buffer allocation problem. Contrary to Gershwin and Schor (2000), we consider two performance measures, the service level and the expected waiting time. In general, it is not possible to find a unique allocation that maximizes the service level and minimizes the expected waiting time. Thus, it is sufficient to find a feasible allocation. Instead of formulating a dual problem to maximize the service level and minimize the waiting time, we consider a constraint satisfaction problem. Therefore, we split the whole problem into a primal and a constraint satisfaction problem. We now introduce the Primal Staffing Model, which is used to describe the primal problem. This introduction is followed by the Feasible Allocation Model denoting the constraint satisfaction problem.

5.1 Primal staffing model

The objective (8) of the *Primal Staffing Model* is the determination of the number of agents on both stages (C_F, C_B) that minimize the total number of agents C^{tot} so that the *X*/*Y* service level *SL* is greater than or equal to a specified value SL^{min} (9) and that the expected waiting time $EW(C_F, C_B) = E[W_{F+B}]$ is lower than or equal to a fixed maximal value EW^{max} (10).

$$\min C^{tot} = C_F + C_B \tag{8}$$

subject to

$$SL(C_F, C_B) \geq SL^{min} \tag{9}$$

$$EW(C_F, C_B) \leq EW^{max} \tag{10}$$

$$C_F^{min} \leq C_F \leq K_F \tag{11}$$

$$C_B^{min} \leq C_B \leq K_B \tag{12}$$

$$C_F, C_B \in \mathbb{N} \tag{13}$$

Constraints (11) and (12) ensure that the number of agents is between the minimal amounts C_F^{min} and C_B^{min} and the maximal amounts, which are determined by the capacity in the front office K_F and in the back office K_B. C_F and C_B are elements of the set of natural numbers (13), so they are nonnegative integers.

5.2 Feasible allocation model

To find a feasible allocation for the agents $\mathbf{C}= (C_F, C_B)$ on both stages that satisfies the *X*/*Y* service level *SL* and the expected waiting time $EW(C_F, C_B) = E[W_{F+B}]$ so that the total number of agents is equal to a desired value C^{tot} (16), we propose the following constraint satisfaction model:

$$SL(C_F, C_B) \geq SL^{min} \tag{14}$$

$$EW(C_F, C_B) \leq EW^{max} \tag{15}$$

$$C_B = C^{tot} - C_F \tag{16}$$

$$C_F^{min} \leq C_F \leq K_F \tag{17}$$

$$C_B^{min} \leq C_B \leq K_B \tag{18}$$

$$C_B \in \mathbb{N} \tag{19}$$

Because we only have two decision variables, we can reduce the Feasible Allocation Model by one variable in (16). Constraints (17) and (18) are equivalent to constraints (11) and (12) of the *Primal Staffing Model*. C_B is also an element of the set of natural numbers (19).

6 Methods

A general solving method for the buffer allocation problem and thus also for our problem is the usage of a Markovian evaluative method (see, e.g., Papadopoulos et al. (2009)). For optimization, the use of complete enumeration is a common approach. Since the computing time for an enumeration can be very long, we developed a bisection method for a reduction. The next section presents the solution method.

6.1 Primal staffing algorithm

Contrary to Gershwin and Schor (2000), we consider two performance measures, the service level and the expected waiting time. We calculate the minimal number of agents C^{tot} needed to satisfy one or two performance measures, which is determined in the primal problem, with the Primal Staffing Algorithm. In each iteration, the determined C^{tot} is transferred to the Feasible Allocation Algorithm. Using this algorithm, an allocation $\mathbf{C} = (C_F, C_B)$ to the front and back office is determined. This allocation is then the input to the next iteration of the Primal Staffing Algorithm. This procedure is repeated until all performance measures are satisfied and a minimal number of agents or no solution is found. The solution procedure can be applied to one or two performance measures. We solve the Staffing Model - Primal problem with the Primal Staffing Algorithm **(see Appendix A for the pseudocode)**. In contrast to Gershwin and Schor (2000), the performance measures to be considered are not always monotonic, as already shown in Sect. 4. In the Primal Staffing Algorithm, we use a binary search. In our numerical experiments, it has been shown that good results are obtained if the assumption that monotonicity holds in the Primal Staffing Algorithm is made. The input to the algorithm is the capacity of the front and back office, K_F and K_B; the minimum number of agents in both offices, C_F^{min}, C_B^{min}; and the values for the desired performance measures, e.g., SL^{min} and EW^{max}. Other inputs include the threshold for overflow Y, the arrival rates λ_F, λ_B, service rates $\mu_F, \mu_{B_1}, \mu_{B_2}$, impatience rate ν and the fraction of calls b for second-level calls.

In the first iteration, the elements of the interval $[(C_F^{min} + C_B^{min}),(C_F^{min} + C_B^{min}) + 1 ,...,(K_F+K_B)]$ to be searched are sorted in ascending order of size. In each iteration, we reduce the interval by half. First, a lower bound for C^{tot} is defined by $l = C_F^{min} + C_B^{min}$ and an upper bound by $u = K_F + K_B$. Therefore, we define the middle of the interval $m = \lfloor \frac{l+u}{2} \rfloor$, which is transferred to the Feasible Allocation Algorithm described in Sect. 6.2. Now, we have two cases regarding the solution of the value for the middle m of the interval. In case one, $SL(m) \geq SL^{min}$ and $EW(m) \leq EW^{max}$, and all performance measures are satisfied. Assuming that the service level is monotonic and that the expected waiting time is given, all solutions of the interval $[(m + 1), u]$ also satisfy the performance measures. Since we minimize the number of agents, these solutions are suboptimal. Therefore, we must search for a minimum in the interval $[l, m]$. The upper bound of C^{tot} is updated to $u = m$.

In the second case, one or two of the performance measures are not sufficient. Again, because of the assumption of the monotonicity property, in this case, all solutions in the interval $[l, m]$ are infeasible. We must search for a minimum in the interval $[(m+1), u]$, and the lower bound is updated to $l = m + 1$.

We repeat the binary search until it is no longer possible to reduce the interval because the variables are integers. If a solution exists, the minimal number of agents is either the lower bound or the upper bound.

If no monotonicity is given, it cannot be excluded that during the binary search in the Primal Staffing Algorithm, the interval in which the minimum exists is cut off. In the case that a solution was found, the interval from $[(C_F^{min} + C_B^{min}), C^{tot*}]$ must be searched for a better solution. In case no solution was found, in the worst situation, all values for C^{tot} must be searched until either a solution was found or until all values of C^{tot} were tested (complete enumeration). Regarding the expected waiting time $E[W_{F+B}]$, if it is ensured that the monotonicity of the service level holds, we can neglect compliance with the waiting time. If the waiting time in the Primal Staffing Algorithm is not met, then the interval $[(C_F^{min} + C_B^{min}), m]$ can be cut off. In the remaining interval $[m+1, (K_F + K_B)]$, whether the waiting time is fulfilled is tested.

6.2 Feasible allocation algorithm

As mentioned before, an allocation for the agents is found by the Feasible Allocation Algorithm based on the total number of agents C^{tot} obtained using the Primal Staffing Algorithm. A bisection method is used in the Feasible Allocation Algorithm. Each element of the interval consists of two parts, each of which, when summed, equals the value of the total number of agents. Therefore, the number of agents in the front and back office is defined as follows:

$$\begin{aligned} \mathbf{C_F} &= (C_{F1}, \ldots, C_{Fi}, \ldots, C_{Fn}) \\ &= (C^{tot} - C_{B1}, \ldots, C^{tot} - C_{Bi}, \ldots, C^{tot} - C_{Bn}), \end{aligned} \tag{20}$$

$$\mathbf{C_B} = (C_{B1}, \ldots, C_{Bi}, \ldots, C_{Bn}), \tag{21}$$

where C_i represents the number of agents for allocation i, resulting in $n = (C_{Bn} - C_{B1}) + 1$ allocations. The values for $\mathbf{C_F}$ are sorted in descending order, and the values for $\mathbf{C_B}$ are sorted in ascending order. Depending on the combination, the smallest value for $\mathbf{C_B}$ is $C_{B1} = Max(C_F^{min}, C_B^{min}, C^{tot} - K_F)$, and the largest value is $C_{Bn} = Min(K_B, C^{tot} - C_B^{min})$. This constraint ensures that the capacity is maintained and that the number of agents in both offices is at least C_F^{min} and C_B^{min}.

In each iteration, the middle $m' = \lfloor \frac{left+right}{2} \rfloor$ of the interval to be searched is determined, with $left = 1$ and $right = n$. Thus, the index of the allocation considered is determined by m'. In our numerical experiments, we observe that the curve of SL has concave and nonconcave areas. In addition, the expected waiting time EW has a minimum. See Fig. 5 as an example. For this reason, we do not know at which point of the curve a feasible solution is found, and therefore, we do not want to cut off the

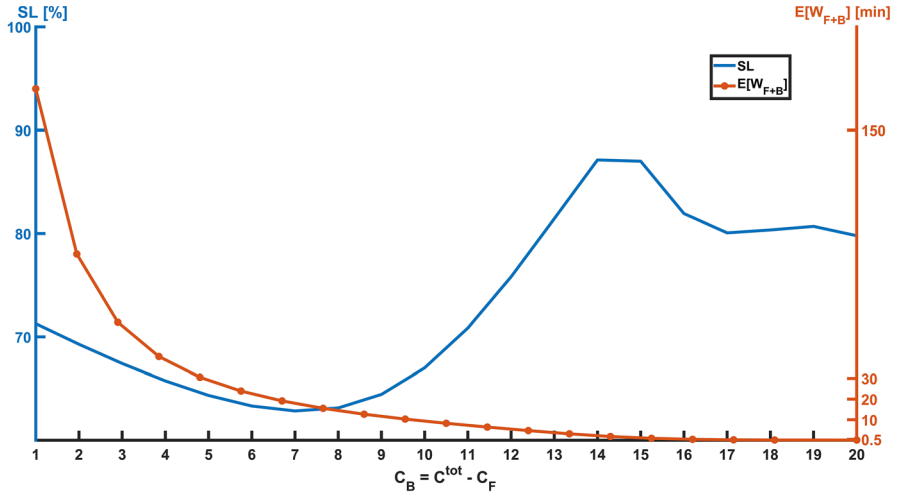

Fig. 5 An example of the service level and waiting time curves for the Feasible Allocation Algorithm

interval in which the solution is located. To avoid this issue, we calculate the first derivatives of the service level $SL'(C_{m'})$ and expected waiting time $EW'(C_{m'})$. The derivatives are numerically approximated by the forward difference:

$$SL'(C_{m'}) = SL(C_{m'+1}) - SL(C_{m'}), \tag{22}$$

$$EW'(C_{m'}) = EW(C_{m'+1}) - EW(C_{m'}), \tag{23}$$

where the number of agents in the back office increased by 1. The function values of the service level are determined by (1) or (4) and those of the expected waiting time are determined using (7). By using the first derivatives, we can distinguish the four following cases **(see Appendix B for the pseudocode)**:

1. In the first case, both desired performance measures are met. Hence, the Feasible Allocation Algorithm terminates and returns the feasible allocation to the Primal Staffing Algorithm.
2. In the second case, *EW* is sufficient, but *SL* is not. The *SL* improves if $SL'(C_{m'}) > 0$. For this reason, we need to reallocate an agent from the front office to the back office in the next iteration. If $SL'(C_{m'}) < 0$, *SL* decreases. Thus, one agent C_B must be reallocated from the back office to the front office in the next iteration of the algorithm.
3. In the third case, *SL* is sufficient, but the desired *EW* is not sufficient. Therefore, the derivative of *EW* is accounted for. If $EW'(C_{m'}) > 0$, then reallocating an agent from the front office to the back office would increase the waiting time. Therefore, C_B must be reallocated in the next iteration of the algorithm. The interval to the right of the middle m' is cut off.

4. If both measures are not met and if $EW'(C_{m'}) > 0$ and $SL'(C_{m'}) < 0$, EW will increase and SL will decrease. Since it cannot be excluded that EW and SL will improve again, we must reallocate an agent C_B from the back office to the front office during the next iteration, and the algorithm is not aborted. If $SL'(C_{m'}) > 0$, we need to reallocate an agent from the front office to the back office during the next iteration. This process improves SL, but EW continues to increase. Since it is not known whether EW will decrease again (which may well happen), the algorithm is not aborted. If $EW'(C_{m'}) < 0$, then a reallocation of a front office agent to the back office occurs if either $SL'(C_{m'}) > 0$ or $SL'(C_{m'}) < 0$. In the first option, both measures are improved. In the last option, SL will be reduced. As it cannot be excluded that SL will improve again, a reallocation of a front office agent to the back office during the next iteration is required.

7 Numerical results

To evaluate the performance of the staffing approach, we performed a number of numerical experiments. For this purpose, the performance analysis and algorithms were implemented in MATLAB (R2022b). The steady-state equations are solved with the so-called backslash operator. We compare the results and computation time of the presented solution method with complete enumeration to highlight the relevance of the algorithm. Furthermore, we compare the results of the two service levels SL_F and SL_{F+B}.

The aim of our sensitivity analysis in this section is to study the impact of the fraction of second-level service b and the impatience rate ν on the minimum number of agents C^{tot}.

7.1 Performance

We compare the results of a small and a medium-sized call center. The capacity of the small call center is $K_F = 25$ and $K_B = 10$. For a medium-sized call center, we double the capacity. The arrival rate λ_F varies in steps to increase the load, which may imply a period during the day when few customers call ($\lambda_F = 2$) or a significant peak load ($\lambda_F = 6$ or $\lambda_F = 12$). The waiting-time limit is set to $Y = 1/3$, which corresponds to 20 s. The fraction of second-level service is set to $b = 0.1$. The values $\nu = 0.1$, $\nu = 3$ and $\nu = 10$ are considered for the impatience rate. Thus, three possibilities are analyzed in terms of reneging and waiting until overflow occurs. On average, customers can wait longer until overflow occurs instead of reneging ($Y > 1/\nu$), they may renege before an overflow becomes possible ($Y < 1/\nu$), or both occur at the time threshold ($Y = 1/\nu$). The target values for the performance measures are selected realistically, such that, e.g., the 80/20 rule is used for the service level ($SL^{min} = 80\,\%$). This rule includes that 80 % of the customers receive service within 20 s. The target expected waiting time is half a minute, e.g., $EW^{max} = 0.5$. All cases are tested with identical processing rates and with unbalanced processing

Table 2 Test cases and results

(a) Test cases

Case	K_F	K_B	λ_F	λ_B	ν	μ_F	μ_{B_1}	μ_{B_2}
1	25	10	2	0.2	0.1	0.25	0.2	0.125
2	25	10	2	0.2	3	0.25	0.2	0.125
3	25	10	2	0.2	10	0.25	0.2	0.125
4	25	10	2	0.2	0.1	0.25	0.25	0.25
5	25	10	2	0.2	3	0.25	0.25	0.25
6	25	10	2	0.2	10	0.25	0.25	0.25
7	25	10	4	0.4	0.1	0.25	0.2	0.125
8	25	10	4	0.4	3	0.25	0.2	0.125
9	25	10	4	0.4	10	0.25	0.2	0.125
10	25	10	4	0.4	0.1	0.25	0.25	0.25
11	25	10	4	0.4	3	0.25	0.25	0.25
12	25	10	4	0.4	10	0.25	0.25	0.25
13	25	10	6	0.6	0.1	0.25	0.2	0.125
14	25	10	6	0.6	3	0.25	0.2	0.125
15	25	10	6	0.6	10	0.25	0.2	0.125
16	25	10	6	0.6	0.1	0.25	0.25	0.25
17	25	10	6	0.6	3	0.25	0.25	0.25
18	25	10	6	0.6	10	0.25	0.25	0.25
19	50	20	8	0.8	0.1	0.25	0.2	0.125
20	50	20	8	0.8	3	0.25	0.2	0.125
21	50	20	8	0.8	10	0.25	0.2	0.125
22	50	20	8	0.8	0.1	0.25	0.25	0.25
23	50	20	8	0.8	3	0.25	0.25	0.25
24	50	20	8	0.8	10	0.25	0.25	0.25

Table 2 (continued)

(a) Test cases

Case	K_F	K_B	λ_F	λ_B	ν	μ_F	μ_{B_1}	μ_{B_2}
25	50	20	10	1	0.1	0.25	0.2	0.125
26	50	20	10	1	3	0.25	0.2	0.125
27	50	20	10	1	10	0.25	0.2	0.125
28	50	20	10	1	0.1	0.25	0.25	0.25
29	50	20	10	1	3	0.25	0.25	0.25
30	50	20	10	1	10	0.25	0.25	0.25
31	50	20	12	1.2	0.1	0.25	0.2	0.125
32	50	20	12	1.2	3	0.25	0.2	0.125
33	50	20	12	1.2	10	0.25	0.2	0.125
34	50	20	12	1.2	0.1	0.25	0.25	0.25
35	50	20	12	1.2	3	0.25	0.25	0.25
36	50	20	12	1.2	10	0.25	0.25	0.25

(b) Results

C^{tot}	C_F	C_B	SL_F	SL_{F+B}	$E[W_{F+B}]$	Staffing algorithm		Complete enumeration		Sol.[c]
						Comp.[a]	Eval.[b]	Comp.	Eval.	
13	10	3	82.83	82.82	0.33	0.97	19	10.55	250	1
12	9	3	84.44	84.43	0.12	0.57	15	10.20	250	1
11	9	2	83.52	83.46	0.43	0.67	23	9.05	250	1
12	10	2	82.79	82.79	0.27	0.58	18	8.67	250	1
11	9	2	84.42	84.42	0.10	0.67	23	8.79	250	1
11	9	2	83.58	83.58	0.05	0.63	23	8.83	250	1
22	18	4	80.57	80.43	0.44	0.50	19	8.03	250	1

Table 2 (continued)

(b) Results

C^{tot}	C_F	C_B	SL_F	SL_{F+B}	$E[W_{F+B}]$	Staffing algorithm		Complete enumeration		Sol.[c]
						Comp.[a]	Eval.[b]	Comp.	Eval.	
19	15	4	80.52	80.45	0.18	0.78	29	9.36	250	1
19	16	3	83.15	82.74	0.48	0.73	28	8.89	250	1
21	18	3	83.24	83.30	0.24	0.50	19	8.62	250	2
17	15	2	80.18	80.14	0.27	0.76	28	8.75	250	1
18	16	2	83.41	83.36	0.22	0.76	28	8.85	250	1
27	23	4	85.63	85.73	0.45	0.64	26	8.87	250	1
26	22	4	80.70	80.13	0.40	0.64	26	8.94	250	1
26	22	4	80.65	80.01	0.34	0.65	26	8.68	250	1
26	23	3	86.26	87.34	0.19	0.64	25	8.70	250	1
25	22	3	81.78	81.86	0.17	0.70	25	8.44	250	1
25	22	3	81.17	81.14	0.12	0.61	25	8.88	250	1
43	35	8	82.81	82.78	0.30	29	30	1,457	1,000	2
35	29	6	82.03	81.90	0.44	32	40	1,518	1,000	2
35	29	6	82.03	81.92	0.36	32	39	1,589	1,000	1
39	35	4	80.60	80.59	0.41	21	25	1,445	1,000	1
32	28	4	80.36	80.36	0.14	32	43	1,524	1,000	1
32	29	3	82.05	81.97	0.46	33	41	1,592	1,000	1
51	41	10	81.40	81.48	0.24	30	34	1,447	1,000	3
43	36	7	81.77	81.46	0.47	26	29	1,485	1,000	2
42	35	7	80.79	80.59	0.36	27	28	1,559	1,000	1
47	42	5	84.44	84.58	0.29	23	28	1,436	1,000	2
39	35	4	80.27	80.23	0.31	24	27	1,493	1,000	1

Table 2 (continued)

(b) Results

C^{tot}	C_F	C_B	SL_F	SL_{F+B}	$E[W_{F+B}]$	Staffing algorithm		Complete enumeration		Sol.[c]
						Comp.[a]	Eval.[b]	Comp.	Eval.	
39	35	4	80.95	80.92	0.22	24	26	1,579	1,000	1
54	45	9	81.33	81.65	0.41	40	31	1,499	1,000	1
50	42	8	81.20	80.78	0.42	29	33	1,493	1,000	2
50	42	8	81.44	81.07	0.36	29	33	1,554	1,000	1
50	45	5	81.74	82.35	0.39	28	33	1,474	1,000	1
46	41	5	80.27	80.25	0.20	24	26	1,492	1,000	1
46	41	5	80.18	80.16	0.13	26	25	1,580	1,000	1

[a]Computation time in [sec.], [b]Evaluations, [c]Solutions, SL_F, SL_{F+B} in [%], $E[W_{F+B}]$ in [min.]

Table 3 Comparison of the results

Case	Solution	C_F	C_B	SL_{F+B}	$E[W_{F+B}]$
10	**1**	**18**	**3**	**83.30**	**0.24**
	2	17	4	80.21	0.22
19	**1**	**35**	**8**	**82.78**	**0.30**
	2	34	9	81.05	0.23
20	**1**	**29**	**6**	**81.90**	**0.44**
	2	28	7	80.35	0.18
25	1	43	8	87.16	0.49
	2	42	9	84.36	0.31
	3	**41**	**10**	**81.48**	**0.24**
26	**1**	**36**	**7**	**81.46**	**0.47**
	2	35	8	80.54	0.23
28	**1**	**42**	**5**	**84.58**	**0.29**
	2	41	6	81.97	0.20
32	**1**	**42**	**8**	**80.78**	**0.42**
	2	41	9	80.00	0.23

SL_{F+B} in [%], $E[W_{F+B}]$ in [min.]

rates $\mu_F > \mu_{B_1} > \mu_{B_2}$. C_F^{min} and C_B^{min} are set to 1 in each office. The combination of the parameters results in 36 instances, listed in Table 2a.

The cases are each calculated considering the service level with respect to the blocking probability of front office customers SL_F and weighted blocking probability SL_{F+B} in addition to the expected waiting time $E[W_{F+B}]$.

The results of the algorithm in regard to a small call center (instances 1–18) and a medium-sized call center (instances 19–36) are identical to the results of the complete enumeration with respect to the minimization of C^{tot}; see Table 2b. The staffing algorithm is devised to terminate if a feasible allocation is found that minimizes C^{tot}. This approach was taken to reduce the computation time. The number of allocations that minimize C^{tot} and are determined by the enumeration are listed in the last column of Table 2b. In most cases, a unique solution exists. The solutions to the enumeration and the algorithm are therefore identical. We note that an optimal solution is found for the test cases using the staffing algorithm.

Table 3 shows a comparison of the results, considering SL_{F+B} and $E[W_{F+B}]$ as performance measures. Up to 3 optimal allocations are found by applying enumeration. The solution determined by the algorithm is in bold. For instance, in case 10, increasing C_B degrades SL_{F+B} and improves $E[W_{F+B}]$. Both allocations result in a minimum number of agents C^{tot} of 21.

Comparing the solutions using SL_F and SL_{F+B} of Table 2b, they are identical regarding the minimum number of agents C^{tot}. As the same allocations were found, the results are also identical regarding the expected waiting time $E[W_{F+B}]$. The results are the same because the blocking probability of the back office customers $P(blocking_B)$ is low and the weighting by λ_{eff_B} is marginal. Therefore, the difference between the two service levels is at most approximately 1 percentage point; see, for example, case 16.

It can be concluded that if the impatience of the customers increases, the number of agents C^{tot} is reduced. If the impatience rate is increased from $\nu = 3$ to $\nu = 10$, C^{tot} can be reduced further (e.g., cases 26 and 27) or C^{tot} can be stable (e.g., cases 35 and 36). The reduction of C^{tot} is at most 4 (cases 10 and 11) for the small call center and 9 for the medium-sized call center (cases 25–27).

An exception is case 12. Here, C^{tot} increases by 1 again. When comparing cases 11 and 12, it is noticeable that the service level $SL_{F+B}(15, 2)$ is fulfilled in case 11 and not in case 12. This result occurs because the reneging probability is $P(reneging) \approx 18\ \%$ for $\nu = 3$. For $\nu = 10$ (case 12), $P(reneging)$ increases to 20.23 %, and the service level is therefore lower. Hence, the service level SL_{F+B} is fulfilled with $C_F = 16$ and $C_B = 2$.

Using Table 2b, the time savings of the staffing algorithm compared to complete enumeration is significant. The greater the front and back office capacities are, the greater the computation time. For smaller call centers, the algorithm is clearly faster, but the computing time between 8 and 11 seconds using complete enumeration is still acceptable. For medium-sized call centers (e.g., $K_F = 50, K_B = 20$), a computing time reduction up to 98.55 % (case 22) demonstrates the speed and importance of the algorithm because the number of Markov chains that are solved increases at complete enumeration. For example, for $K_F = 25$ and $K_B = 10$, there are $K_F \cdot K_B = 250$ evaluations, and for $K_F = 50, K_B = 20$, the number of evaluations rises to 1,000. The computing time increases due to the increased number of evaluations and because of the state space's size. For instance, this phenomenon can be observed when comparing case 18 and case 19 in the results for the staffing algorithm. The number of evaluations has increased by only 5, but the calculation time has increased by 4,654 % due to the increase in the state space. The significance of reducing the computing time becomes clear when a calculation is made for more than just one period. In a call center, for example, the working day is divided into 24 periods of 30 min each. For each period, a calculation by the algorithm is needed. In addition, different targets for the service conditions are tested.

7.2 Sensitivity analysis

7.2.1 Influence of fraction *b* requiring second-level service in the back office

In this section, we analyze the influence of the fraction b of calls that require second-level service. When b grows, more customers require second-level service in the back office. For that purpose, we use case 9 and increase the capacity to $K_F = 50$ and $K_B = 20$. Figure 6 shows that the number of agents C^{tot} more than doubles when b is increased. The number of agents in the back office C_B increases almost to the capacity limit at $b = 0.7$ and $b = 0.8$.

In addition to the increase in C_B, the number of front office agents C_F also increases to meet the performance measures. From $b = 0.9$, the required performance measures can no longer be met by additional front office agents. When the capacity limit $K_F + K_B$ is reached, the weighted expected waiting time $E[W_{F+B}]$ is

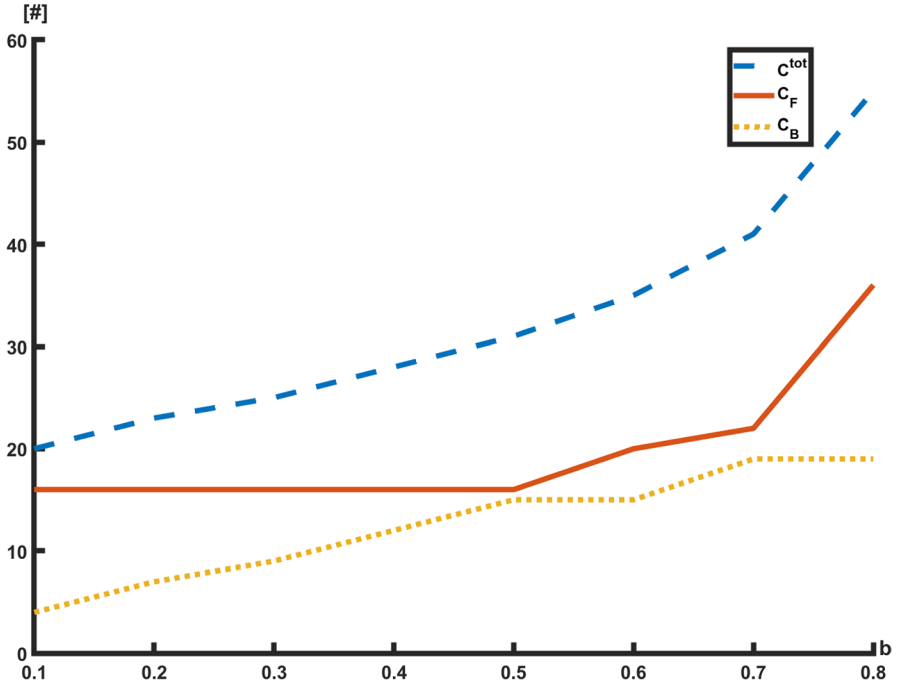

Fig. 6 Impact of b on the minimum number of agents C^{tot} and on the allocation

sufficient. Due to a high blocking probability of 74.69 % in the back office and thus a weighted blocking probability of 22.90 %, the service level is only 77.10 %.

As mentioned in Sect. 4, the monotonicity of the service level in C^{tot} does not always hold. To check the monotonicity, we vary the value b from 0.1 to 1 with a step size of 0.1. We use instance 32. For $b = 0.1$, SL_F and SL_{F+B} are monotonic in C^{tot}. As an example, Fig. 7 shows the curves of the service levels, blocking probabilities $P(blocking_F)$, $P(blocking_B)$ and reneging probability $P(reneging)$ with $b = 0.1$.

Then, the service level is no longer monotonically increasing in C^{tot} for $b = 0.2$ since at $C_F = 49$, the value first increases and then decreases again at $C_F = 50$.

In general, it can be observed that a higher value for b increases the reneging probability for $C_B = 1$ and running C_F. At $b = 0.1$ and $C_F = 49$, the reneging probability is 2.79 %, and at $b = 0.2$, it is 4.26 %. The reneging probability $P(reneging)$ decreases to 0 % at $C_F = 50$. Therefore, the service level of arriving customers, e.g., which only takes reneging into account, increases. Simultaneously, the blocking probabilities $P(blocking_F)$ and $P(blocking_B)$ in the front and back office increase. As a result, SL_F and SL_{F+B} decrease.

7.2.2 Influence of impatience rate ν on the number of agents C^{tot}

The expected waiting time $E[W_{F+B}]$ and the X/Y service level are affected by the impatience rate ν. To analyze the impact of ν, we use case 9 with $K_F = 50$ and

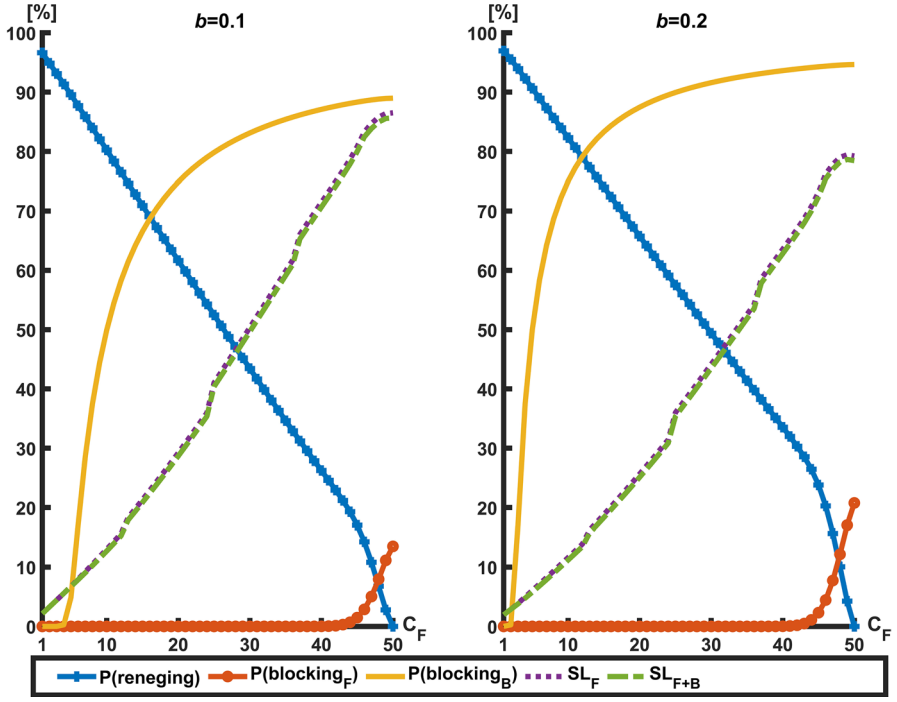

Fig. 7 Comparison of SL_F, SL_{F+B} and reneging and blocking probabilities as a function of the number of agents C_F for case 32 and $C_B = 1$

$K_B = 20$ and vary the fraction b to receive second-level service in the back office from 0.1 to 0.8 with a step size of 0.1.

Figure 8 shows the results for very patient customers ($\nu = 0.1$), very impatient customers ($\nu = 10$) and those in between ($\nu = 3$). The performance measures are fulfilled for all values shown for C^{tot}.

It should be noted that in the case of very patient customers, the minimum number of agents C^{tot} is higher by up to 21 % ($b = 0.1, \nu = 3$). For very impatient customers and those in between, C^{tot} is identical for $b = 0.3$ to $b = 0.5$ and differs by 1 agent at most. For $b = 0.6$ to $b = 0.8$, the minimum number of agents C^{tot} is lower the more impatient the callers are. The minimum staffing requirement is thus dependent on the impatience rate ν.

8 Conclusions

We have extended the performance analysis from Stolletz and Manitz (2013) to an optimization problem to determine the minimum number of agents in multi-stage call centers. We introduced a fast algorithm to determine the minimum total number of agents and their allocation. The algorithm can be applied arbitrarily if the properties of the performance indicators to be considered are known. The

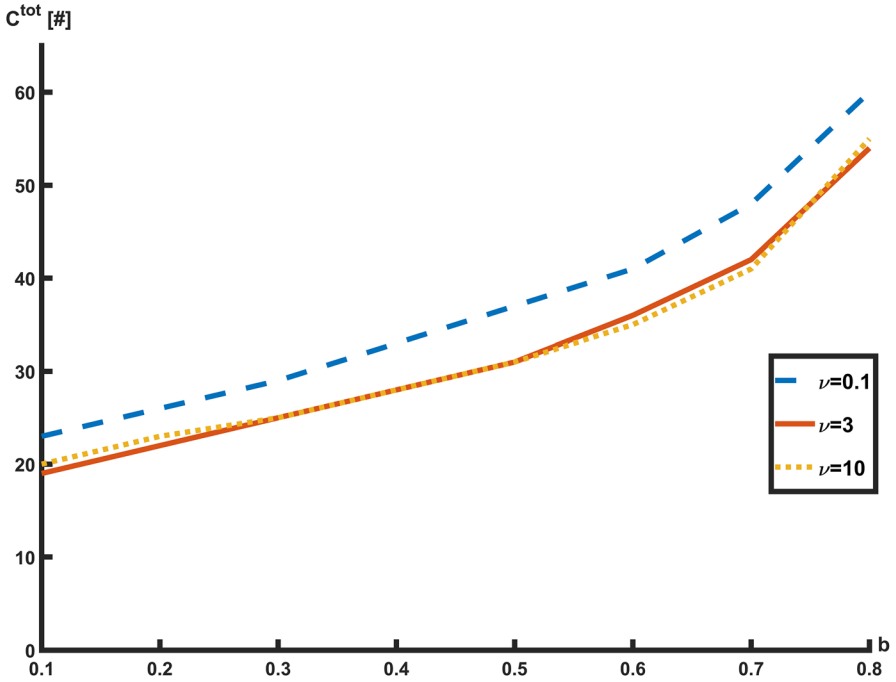

Fig. 8 Impact of ν on the minimum number of agents C^{tot} for various values of fraction b to receive second-level service in the back office

proposed staffing algorithm always found the optimal solution for all test cases. We can show it with a complete enumeration for small test cases. Nevertheless, it is a heuristic approach; thus, an optimal solution is not guaranteed. However, this limitation is negligible if, for example, too many back-office agents have been scheduled. These agents are in the office anyway and are only activated when they are needed. If an extra agent is added and is not busy, the agent can work on something else during this time, e.g., e-mails. If an unexpectedly high call load occurs, then this agent can be seen as a buffer. An advantage of this method is the short computation time compared to a complete enumeration or a simulation.

The calculated staffing requirements can be combined with a shift scheduling problem when considering multiple periods. For a multiple-period model, we can aggregate stationary service measures. In this case, the day is divided into various periods, and the presented solution approach is applied to each period. The stationary backlog-carryover (SBC) approach as proposed by Stolletz (2008) or the stationary independent period-by-period (SIPP) approach can then be used for approximation; see, e.g., Green et al. (2001). The staffing and scheduling problem can be solved in two steps. First, the required staffing levels for various periods are calculated, and based on the result, the agents are scheduled into shifts. The scheduling problem can be solved by determining the staffing levels with our method using the approach presented in Bhulai et al. (2008). In this context, global performance measures can be considered to avoid overstaffing. Other

extensions are the consideration of multi-skill agents in one or both offices and solutions to optimization problems in which the waiting time value $Y = t$ is a managerial decision variable which of course can be done with the methods we have presented.

Appendix A Pseudocode primal staffing algorithm

Algorithm 1 Primal Staffing Algorithm

procedure PRIMAL($C_F^{min}, C_B^{min}, K_F, K_B, \lambda_F, \lambda_B, b, Y, \nu, \mu_F, \mu_{B_1}, \mu_{B_2}, SL^{min}, EW^{max}$)
$l = C_F^{min} + C_B^{min}$, $u = K_F + K_B$
while $l \leq u$ **do**
    $m = \lfloor (l + u)/2 \rfloor$
    procedure FEASIBLE($C_F^{min}, C_B^{min}, K_F, K_B, m, \lambda_F, \lambda_B, b, Y, \nu, \mu_F, \mu_{B_1}, \mu_{B_2}, SL^{min}, EW^{max}$)
    return SL_{max}, EW_{min}, C
    end procedure
    if $SL_{max} \geq SL^{min}$ *and* $EW_{min} \leq EW^{max}$ **then**
        $u = m$
    else
        $l = m$
    end if
end while
$C^* = C$
$SL^* = SL_{max}$
$EW^* = EW_{min}$

Appendix B Pseudocode feasible allocation algorithm

Algorithm 2 Feasible Allocation Algorithm

procedure FEASIBLE($C_F^{min}, C_B^{min}, K_F, K_B, m, \lambda_F, \lambda_B, b, Y, \nu, \mu_F, \mu_{B_1}, \mu_{B_2}, SL^{min}, EW^{max}$)
$C^{tot} = m$
$C_{B1} = Max(C_F^{min}, C_B^{min}, C^{tot} - K_F)$
$C_{Bn} = Min(K_B, C^{tot} - C_{B1})$
$n = (C_{Bn} - C_{B1}) + 1$
$C_B = (C_{B1}, C_{B1} + 1, ..., C_{Bn})$
$C_F = (C_{F1}, C_{F2}, ..., C_{Fn}) = (C^{tot} - C_{B1}, C^{tot} - C_{B2}, ..., C^{tot} - C_{Bn})$
$C = (C_1, C_2, ..., C_n)^T = (C_F, C_B)^T$
$left = 1,\ right = n$
while $left \leq right$ **do**
  $m^{'} = \lfloor (left + right)/2 \rfloor$
  $SL(C_{m'}) =$ DETERMINE$SL(C_{m'}), EW(C_{m'}) =$ DETERMINE$EW(C_{m'})$
  if $SL(C_{m'}) \geq SL^{min}$ *and* $EW(C_{m'}) \leq EW^{max}$ **then**
    $SL_{max} = SL(C_{m'}), EW_{min} = EW(C_{m'}), C = C_{m'}$
    return SL_{max}, EW_{min}, C
  end if
  $SL(C_{m'+1}) =$ DETERMINE$SL(C_{m'+1}), EW(C_{m'+1}) =$ DETERMINE$EW(C_{m'+1})$
  $SL'(C_{m'}) = SL(C_{m'+1}) - SL(C_{m'}), EW'(C_{m'}) = EW(C_{m'+1}) - EW(C_{m'})$
  if $SL(C_{m'+1}) \geq SL^{min}$ *and* $EW(C_{m'+1}) \leq EW^{max}$ **then** ▷ Case 1
    $SL_{max} = SL(C_{m'+1}), EW_{min} = EW(C_{m'+1}), C = C_{m'+1}$
    return SL_{max}, EW_{min}, C
  end if
  if $SL(C_{m'}) < SL^{min}$ *and* $EW(C_{m'}) \leq EW^{max}$ **then** ▷ Case 2
    if $EW'(C_{m'}) > 0$ **then**
      if $SL'(C_{m'}) > 0$ **then**
        $left = m^{'}$
      else
        $right = m^{'}$
      end if
    else
      if $SL'(C_{m'}) > 0$ **then**
        $left = m^{'}$
      else
        $right = m^{'}$
      end if
    end if
  end if
  if $SL(C_{m'}) \geq SL^{min}$ *and* $EW(C_{m'}) > EW^{max}$ **then** ▷ Case 3
    if $EW'(C_{m'}) > 0$ **then**
      if $SL'(C_{m'}) > 0$ **then**
        $right = m^{'}$
      else
        $right = m^{'}$
      end if
    else
      if $SL'(C_{m'}) > 0$ **then**
        $left = m^{'}$
      else
        $left = m^{'}$
      end if
    end if
  end if
  if $SL(C_{m'}) < SL^{min}$ *and* $EW(C_{m'}) > EW^{max}$ **then** ▷ Case 4
    if $EW'(C_{m'}) > 0$ **then**
      if $SL'(C_{m'}) > 0$ **then**
        $left = m^{'}$
      else
        $right = m^{'}$
      end if
    else
      if $SL'(C_{m'}) > 0$ **then**
        $left = m^{'}$
      else
        $right = m^{'}$
      end if
    end if
  end if
end while
end procedure

Author contributions Michael Manitz: Supervision. Marc-Philip Piehl: Conceptualization, Methodology, Writing - Original Draft, Software.

Funding Open Access funding enabled and organized by Projekt DEAL. No funds, grants, or other support was received.

Declarations

Conflict of interests The authors have no relevant financial or non-financial interests to disclose.

Data The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

References

Aksin Z, Armony M, Mehrotra V (2007) The modern call center: A multi-disciplinary perspective on operations management research. Prod Oper Manag 16(6):665–688. https://doi.org/10.1111/j.1937-5956.2007.tb00288.x

Barth W, Manitz M, Stolletz R (2010) Analysis of two-level support systems with time-dependent overflow-a banking application. Prod Oper Manag 19(6):757–768. https://doi.org/10.1111/j.1937-5956.2010.01155.x

Bekker R, Koole GM, Nielsen BF et al. (2011) Queues with waiting time dependent service. Queueing Syst 68(1):61–78. https://doi.org/10.1007/s11134-011-9225-2

Bhulai S, Koole G, Pot A (2008) Simple methods for shift scheduling in multiskill call centers. Manuf Serv Oper Manag 10(3):411–420. https://doi.org/10.1287/msom.1070.0172

Chevalier P, van den Schrieck JC (2008) Optimizing the staffing and routing of small-size hierarchical call centers. Prod Oper Manag 17(3):306–319. https://doi.org/10.3401/poms.1080.0033

Gans N, Koole G, Mandelbaum A (2003) Telephone call centers: Tutorial, review, and research prospects. Manuf Serv Oper Manag 5(2):79–141. https://doi.org/10.1287/msom.5.2.79.16071

Gershwin SB, Schor JE (2000) Efficient algorithms for buffer space allocation. Ann Oper Res 93(1/4):117–144. https://doi.org/10.1023/A:1018988226612

Green LV, Kolesar PJ, Soares J (2001) Improving the sipp approach for staffing service systems that have cyclic demands. Oper Res 49(4):549–564. https://doi.org/10.1287/opre.49.4.549.11228

Grossman TA, Oh SL, Rohleder TR, et al. (2001) Call centers. In: Gass SI, Harris CM (eds) Encyclopedia of Operations Research and Management Science. Springer US, New York, NY, p 73–76, https://doi.org/10.1007/1-4020-0611-X

Henderson SG, Mason AJ (1998) Rostering by iterating integer programming and simulation. In: Medeiros DJ (ed) 1998 Winter Simulation Conference. IEEE, Piscataway, N.J and New York, N.Y and San Diego, Calif, pp 677–683, https://doi.org/10.1109/WSC.1998.745050

Kim JW, Park SC (2010) Outsourcing strategy in two-stage call centers. Comput Oper Res 37(4):790–805. https://doi.org/10.1016/j.cor.2009.06.020

Koole G, Mandelbaum A (2002) Queueing models of call centers: An introduction. Ann Oper Res 113(1/4):41–59. https://doi.org/10.1023/A:1020949626017

Koole G, Pot A (2006) An overview of routing and staffing algorithms in multi-skill customer contact centers. Technical Report, Department of Mathematics, Vrije Universiteit Amsterdam, The Netherlands

Koole G, van der Sluis E (2003) Optimal shift scheduling with a global service level constraint. IIE Trans 35(11):1049–1055. https://doi.org/10.1080/07408170304398

Koole GM, Nielsen BF, Nielsen TB (2012) First in line waiting times as a tool for analysing queueing systems. Oper Res 60(5):1258–1266. https://doi.org/10.1287/opre.1120.1089

Koole GM, Nielsen BF, Nielsen TB (2015) Optimization of overflow policies in call centers. Probab Eng Inf Sci 29(3):461–471. https://doi.org/10.1017/S0269964815000091

Liao S, Koole G, van Delft C et al. (2012) Staffing a call center with uncertain non-stationary arrival rate and flexibility. OR Spectr 34(3):691–721. https://doi.org/10.1007/s00291-011-0257-0

Papadopoulos CT, O'Kelly MEJ, Vidalis MJ, et al. (2009) Analysis and Design of Discrete Part Production Lines, Springer Optimization and Its Applications, vol 31. Springer-Verlag New York, New York, NY, https://doi.org/10.1007/978-0-387-89494-2, http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10309711

Pinedo M, Seshadri S, Shanthikumar JG (2000) Call centers in financial services: Strategies, technologies, and operations. In: Melnick EL, Nayyar PR, Pinedo ML, et al. (eds) Creating Value in Financial Services. Springer US, Boston, MA, p 357–388, https://doi.org/10.1007/978-1-4615-4605-4_18

Pot A, Bhulai S, Koole G (2008) A simple staffing method for multiskill call centers. Manuf Serv Oper Manag 10(3):421–428. https://doi.org/10.1287/msom.1070.0173

Stolletz R (2003) Performance Analysis and Optimization of Inbound Call Centers, Lecture Notes in Economics and Mathematical Systems, vol 528. Springer, Berlin and Heidelberg,. https://doi.org/10.1007/978-3-642-55506-0

Stolletz R (2008) Approximation of the non-stationary m(t)/m(t)/c(t)-queue using stationary queueing models: The stationary backlog-carryover approach. Eur J Oper Res 190(2):478–493. https://doi.org/10.1016/j.ejor.2007.06.036

Stolletz R, Manitz M (2013) The impact of a waiting-time threshold in overflow systems with impatient customers. Omega 41(2):280–286. https://doi.org/10.1016/j.omega.2012.05.001

Wallace RB, Whitt W (2005) A staffing algorithm for call centers with skill-based routing. Manuf Serv Oper Manag 7(4):276–294. https://doi.org/10.1287/msom.1050.0086