Real-Time Scheduling Policies for Multiclass Call Centers with Impatient Customers

Oualid Jouini[†] Auke Pot[‡] Yves Dallery[†] Ger Koole[‡]

[†] Ecole Centrale Paris
Laboratoire Génie Industriel
Grande Voie des Vignes
92295 Châtenay-Malabry Cedex
France

[‡] Vrije Universiteit Amsterdam
Department of Mathematics
De Boelelaan 1081a
1081 HV Amsterdam
The Netherlands

walid.jouini@ecp.fr, yves.dallery@ecp.fr sapot@few.vu.nl, koole@few.vu.nl

Corresponding author: Oualid Jouini, tel: +33 1 41 13 11 66, fax: +33 1 41 13 12 72

Submitted To *Naval Research Logistics*, January 2007

Abstract

We consider a call center model with two classes of impatient customers, VIP and less important ones. After entering the queue, a customer will wait a random length of time for service to begin. If service has not begun by this time he will renege and considered to be lost. We focus on developing queue joining and call selection policies. The policies satisfying performance constraints related to the ratio of the loss probabilities, as well as the variance of waiting times in queue. We propose and compare several real-time scheduling policies in order to reach our objective. The policies are characterized by simplicity and ease of implementation.

keywords: call centers; queueing systems; reneging; dynamic scheduling policies; non-preemptive priority

1 Introduction

A call center, or in general a contact center, is defined as a service system in which agents (servers) serve customers (callers), over telephone, fax, email, etc. In the past few years, call centers have been introduced with great success by many service-oriented companies such as banks and insurance companies. They become the main point of contact with the customer, and an integral part of the majority of corporations. Managing a call center is a diverse challenge due to many complex factors. This paper deals with a real-time problem of call centers, namely customer routing and server scheduling. Given a system structure and staffing level, our purpose is to develop routing schemes for arrival calls subject to satisfying some given quality of service constraints.

It is common in modern call centers, to have multiple types of customers as well as agents with different skills. It is referred to as multi-skill call centers. The typical example, see Gans et al.

(2003), is an international call center where incoming calls are in different languages. Related studies include those by Garnett and Mandelbaum (2001), Akşin and Karaesmen (2002), and references therein. This topic is out of the scope of this paper. Our concern here is a full-flexible call center. We assume that all agents are flexible enough (polyvalent) to answer all requirements of service. However, we divide customers into two different classes according to their importance, VIP and ordinary customers. The resulting model for our call center here has a V-design according to the canonical designs presented in Garnett and Mandelbaum (2001). In addition, we let customers be impatient. Incorporating reneging in theoretical models is of value. In reality, it is natural that a waiting customer will wait for only a limited time, and will hang up within that time. Ignoring reneging leads to overstaffing and pessimistic estimation of queueing delays. Garnett et al. (2002) show using numerical examples that models with and without abandonment tend to give very different performance measures even if the abandonment rate is small. Models including reneging are therefore closer to reality. This is necessary to obtain more accurate managerial insights.

In this paper, we discuss various real-time (online) scheduling policies subject to satisfying differentiated service levels related to the probabilities of being lost. The target consists of a balance between the achieved service levels of customer classes. This objective is motivated by a situation that often occurs in practice due to the uncertain environment of call centers. It happens when the workload prediction step is incorrectly done. Several studies as in Jongbloed and Koole (2001) and Avramidis et al. (2004) have shown that the arrival process and the workload are hard to predict in call centers. We show in this paper that using online scheduling policies is an effective way to compensate workforce for mismatches between different customer classes, and meet targets on service levels. The main advantage of our control policies is that they require no information about the arrival processes in advance. If fluctuations in workload occur, the policies are adapted such that targets on the service levels are met.

In general, the provision of differentiated service levels relies on the use of priority queues. Schrage and Miller (1966) have shown that scheduling policies similar to multiclass priority queues allow to achieve high performances, often nearly as good as those under optimal policies. Note, in addition, that the priority schemes are easy to implement, which explain their prevalence in practice. However, it is well known that fixed strict priority policies result either in satisfying target performances for lower priority customers and an over service level for higher priority ones, or in satisfying the target for high priority customers while having heavily penalized lower priority customers.

Our purpose here is to develop simple and useful dynamic routing policies that are based on priority schemes. On the one hand, optimal routing policies are very complex to obtain because of the mathematical difficulties. On the other hand, they are usually not interesting to implement in practice due to their several requirements about real-time system information, see Koole and Pot (2006). Hence, it might be more beneficial to use simple scheduling policies instead of attempting to use optimal ones. We derive various schemes for dynamic assignment of customers to queues, as well as for selection of waiting calls. The policies we propose are characterized to be workconserving (non-idling). A policy is defined to be workconserving if there can be no idling servers when there are waiting customers, which is natural for large service systems such as call centers. Through the analysis below, we also focus our interest on the achieved variance of the waiting time in order to differentiate the proposed policies. We do not focus on the achieved first moment of the waiting time. In fact, our proposed policies allow to achieve not very different values of mean waiting times. We thereafter prefer a system with low waiting time variance than a system that is faster on average but highly variable. A further advantage of minimizing the variance of the waiting time is related to the announcement of anticipated delays to customers. Computing the full distribution of the state-dependent waiting time is a very complex task. So, we aim to minimize its variance such that announcing the state-dependent mean waiting time would not be a bad prediction. Further details on call centers with delays announcement can be found in Whitt (1999).

The interesting side of the policies we suggest comes from their simplicity. They do not require information about the workload process. The analysis yields quantitative insights, as well as useful principles and guidelines for the control problem. Note that non-workconserving policies, such as thresholds or reservations of agents for important customers, are not considered in this paper. In our opinion, the restriction does not decrease the usefulness of the analysis because we conjecture that the larger the call centers, the more effective are workconserving policies, see Pot (2006). A further reason is that we want to prevent our analysis from being too complicated and therefore not interesting for practitioners.

Here is how the rest of the paper is organized. In Section 2, we review two kinds of literature close to our work. The first one deals with queueing models incorporating reneging, and the second deals with results about optimal scheduling policies. In Section 3, we give a comprehensive presentation of the problem we study in this paper: Section 3.1 is devoted to formulate the queueing model of the call center, Section 3.2 gives some preliminary results, and Section 3.3 concretely expresses our objective. In Section 4, we develop dynamic scheduling policies that allow to meet our objective. We propose various policies belonging to two different families. In Section 4.1, we focus on the family of queue joining policies. In Section 4.2, we investigate the dual family of call selection policies. In Section 5, we present and discuss simulation experiments of the proposed policies. Sections 5.1 and 5.2 are devoted to the simulation experiments of the queue joining and call selection policies, respectively. In Section 5.3, we give some elements of comparison between both families of policies. In Section 6, we focus on extending our analysis to the case of three customer classes. In Section 7, we give some concluding remarks and highlight some directions for future research.

2 Literature Review

The literature related to this paper spans into two main areas. The first deals with queueing systems with impatient customers. The second area deals with the control of queueing systems, specifically, the problem of customer routing and server scheduling.

In the following, we highlight some of the literature with regard to the first area. Queueing models incorporating impatient customers have received a lot of attention in the literature. Gross and Harris (1998) define the impatience through three different forms. The first is balking, that is, the reluctance of a customer to join a queue upon arrival. The second is reneging, which means the reluctance to remain in queue after joining and waiting. Finally, the third is jockeying between separate queues. Jockeying means that one customer has the possibility to change to one queue while he is waiting in another. In this paper, we consider the second form of impatience. The other forms are not allowed. To underline the importance of the abandonment modeling in the call center field, the authors in Gans et al. (2003) and in Mandelbaum and Zeltyn (2006) give some numerical examples that point out the effect of abandonment on performances. The literature on queueing models with reneging focus especially on performance evaluation. We refer the reader to Ancker and Gafarian (1962), Garnett et al. (2002), and references therein for simple models assuming exponential reneging times. In Garnett et al. (2002), the authors study the subject of Markovian abandonments. They suggest an asymptotic analysis of their model under the heavy-traffic regime. Their main result is to characterize the relation between the number of agents, the offered load and system performances such as the probability of delay and the probability to abandon. This can be seen as an extension of the results of Halfin and Whitt (1981) by adding abandonments. Zohar et al. (2002) investigate in their work the relation between customers reneging and the experience of waiting in queue. Other papers have allowed reneging to follow a general distribution. Related studies include those by Baccelli and Hebuterne (1981), Brandt and Brandt (2002), Ward and Glynn (2003), and references therein.

Let us now focus on the second area of literature closed to our work, that is, the control of queueing systems. Scheduling policies has been studied in great depth within the context of queueing systems. A scheduling policy, or a discipline of service, prescribes the order in which customers are served. It is tied to identifiable characteristics of customers. Arrival time is certainly one of these characteristics, it is the basis for the most familiar disciplines as the first come, first served (FCFS) discipline and the last come, first served (LCFS) discipline. Several other characteristics are the bases for queue disciplines. Customers may be processed according to service times, which may lead to the well known Shortest Remaining Processing Time discipline (SRPT.) The queue discipline may also be based on the customer type, for example, VIP customers are scheduled first. Or the queue discipline may be an hybrid strategy that accounts for more than one characteristic. The focus in control problems is on determining the form of the optimal policy to optimize system performance. Randolph (1991) classifies scheduling policies into those using dynamic schedule rules and those using static schedule rules. A dynamic schedule is a discipline that is continuously updated as customers arrive and are processed. However, a static schedule is state of system independent, it is beforehand defined and never altered. Each one of the above classes of policies can be further classified into two major types: agent scheduling and customer routing. As defined in Garnett and Mandelbaum (2001), agent scheduling is described by a control decision taken whenever an agent turns idle and there are queued customers: which customer, if any, should be routed to this agent. However, a scheduling policy based on a customer routing rule, is defined by a control decision taken whenever a customer arrives: which idle agent, if any, should serve this customer, if not, to which queue should the customer of interest be routed.

Under some given performance constraints, we develop in this paper various dynamic scheduling policies based on both types of control decision. Our basic call center model has a V-design, with two infinite queues. When studying agent scheduling policies, we assume a priority schema between both queues; a high priority queue and a lower priority one. We then focus on policies that assign a priority level to customers upon arrival. We should note that there are two further possible refinements in priority situations, namely preemption and non-preemption. In the preemptive case, a customer with high priority is allowed to enter service immediately even if another one with lower priority is already present in service at its arrival epoch. However, a priority discipline is said to be non-preemptive if there is no interruption. A customer with higher priority just goes to the head of the queue and waits for his turn. The scheduling policies analyzed in this work are characterized to follow a non-preemptive priority schemes.

In the following, we present some known results about optimal scheduling policies. Note that due to the complexity of such studies, most of the existing research considers simple queueing models. Moreover, the literature illustrates that optimal results are difficult to obtain. The structure of optimal policies are model dependent and often difficult to generalize to more complicated cases. Schrage and Miller (1966) proves that the SRPT policy, which schedules in a preemptive manner the customer with the smallest remaining processing time at every point in time, is optimal with respect to minimizing mean sojourn times in an $M/G/1$ system. Pekoz (2002) addresses the analysis of a multiserver non-preemptive priority queue with exponentially distributed interarrival and service times. He finds and evaluates the performance of an asymptotically optimal policy that minimizes the expected queueing delay for high priority customers. Guérin (1998) presents a model without waiting queues. It contains a multi-server station, which receives low and high priority arrivals. He develops an admission policy for the low priority customers such that the fraction of blocked high priority customers is bounded and he analyzes the system under that policy. Örmeci (2004) considers a markovian call center model with two classes of customers, one pool of generalists, two pools of specialists, and no waiting rooms. Then, the author derives the structure of dynamic admission policies that maximizes, in the long-run, the total expected discounted revenue. Aguir et al. (2006) present an optimization problem for an $M/M/1$ queue with two classes of customers. They prove and characterize a class of optimal static scheduling policies subject to satisfying differentiated performances for customer classes. The proposed policies are based on strict priority rules, and the performances are measured in terms of the mean waiting time and the 80/20 rule. The authors investigate extensions to the multiserver queue, and analyze a dual class of static policies based on agent scheduling. We also refer the readers to Xu et al. (1992), Huang (2001), Gans and Zhou (2003) for more results. Other papers, related to scheduling problem for multiserver systems under the asymptotic heavy-traffic regime, include those by Gans and van Ryzin (1998), Bassamboo et al. (2004), Armony (2005), and references therein.

The literature presents different numerical techniques to improve the performance of stochastic processes. Examples are perturbation analysis and likeli-hood-ratio methods, see for example Meketon (1987) for an survey. Both methods have in common that they allow us to update the policy in an online fashion. They also can handle arrival processes of which not much is known in advance, as is the case in our study.

3 Framework

In this section, we first describe the basic model of our call center. Second, we present notations related to the performances we consider in this paper, and develop some preliminary results. Finally based on the preliminary study, we concretely specify our motivation and objective with regard to the scheduling policies we aim to develop.

3.1 Model Formulation

We model our call center as a queueing system with two classes of customers; important customers type A, and less important ones type B. The model consists of two infinite queues type 1 and 2, and a set of s identical servers representing the set of agents. All agents are able to answer all types of customers. The call center is operated in such a way that at any time, any call can be addressed by any agent. So upon arrival, a call is addressed by one of the available agents, if any. If not, the call must join one of the queues. As mentioned before, we consider two types of

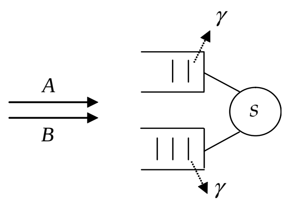

Figure 1: The basic model

scheduling policies, queue joining and call selection policies. The first type determines the rule of assigning customers upon arrival to one of the queues. However, the second type determines at each service completion which waiting call, if any, should start service. Within each queue, customers are served in order of their arrivals, i.e., under the FCFS discipline. Interarrival times and service times are assumed to be i.i.d., and follow general distributions. In certain cases, we shall in particular consider the exponential distribution for successive service times. Then, a customer is served with rate, say μ, independent of the customer type.

In addition, we assume that the customers are impatient. After entering the queue, a customer will wait a random length of time for service to begin. If service has not begun by this time he will renege (leaves the queue.) Times before reneging, for type A and B customers, are assumed to be i.i.d. and exponentially distributed with rate γ. Assuming identical distribution of patience within each class, independently from their position in the queue, seems to be a plausible assumption for call centers, see Gans et al. (2003). Indeed, the tele-queueing experience in call centers is fundamentally different from that of a physical queue, in the sense that customers do not see others waiting and need not be aware of their "progress" (position in the queue) if the call center does not provide information about queueing delays. The system is workconserving, i.e., a server is never forced to be idle with customers waiting. Finally, retrials are ignored, and reneging is not allowed once one customer starts his service. We do not allow also jockeying between separate queues. Following similar arguments, the behavior of this call center can be viewed as a variety of a $GI/GI/s+M$ queueing system. The symbol M after the $+$ is to indicate the markovian assumption for times before reneging. Note that owing to abandonments, the system is unconditionally ergodic. The resulting model is shown on Figure 1.

In this paper, the main results concerns the case when service times, as well as times before reneging, are identically distributed for both customer types. There are two reasons for that. The first reason is related to the nature of the call center we consider here, and which is the case in many other call center applications. In fact, the difference making a type A customer more important for the company than a type B customer is that the first pays more money than the second one. In concrete terms, if our call center owns every month from one customer an amount of money crossing a given threshold, then that customer is of type A, else he is of type B. We consider that the queries asked by customers, as well as the patience experiences do not differ too much from one type of customer to another. Therefore, it would be credible to assume a common distribution. The second reason is due to the complexity of the analysis when assuming different service and reneging time distributions. Our objective here is to investigate simple models that allow us to better understand the system behavior and to gain general useful guidelines.

Finally, note that assuming exponential distribution for service times and times before reneging is not a bad approximation. It is true that in real call centers cases, impatience times need not be exponential, and they can vary significantly with the type of service, the information provided during waiting, etc. We refer the reader to Zohar et al. (2002) and Mandelbaum et al. (2000) who show how such assumptions may be violated. However, our model is still of interest in practice as mentioned by Pierson and Whitt (2006). The authors have shown, using various simulation experiments, that a variety of an $M/M/s+M$ (Erlang-A) model provides a good approximation of the $M/GI/s+GI$ model.

3.2 Preliminaries

In this section, we first present notations and definitions about the performances we are interested on. The performances are defined in terms of the fraction of customers who abandon, the mean and the variance of the waiting time in queue. Second, we develop some structural results about the relation of these performances with the scheduling policies.

We denote by m the type of a customer, $m \in \{A, B\}$. We assume that at time $t = 0$, the system starts empty. Under some given scheduling policy π, let $n^m(t)$ be the number of type m arrivals during the interval of time $[0, t]$, $t > 0$. Let $a^m_\pi(t)$ be the number of type m customers who abandon the queue, and $b^m_\pi(t)$ the number of those who get service. We define a first service level in terms of the fraction of customers who abandon within each type, as well as for all types of customers. The fraction of type m customers who abandon, say $Q^m_\pi(t)$, during $[0, t]$ is defined by $Q^m_\pi(t) = \frac{a^m_\pi(t)}{n^m(t)}$. As for the overall service level, it is defined by $Q_\pi(t) = \frac{a^A_\pi(t)+a^B_\pi(t)}{n^A_\pi(t)+n^B_\pi(t)}$.

During the stationary regime, the service level for type m customers, say Q^m_π, and the overall service level for all types, say Q_π, are given by $Q^m_\pi = \lim_{t\to\infty} Q^m_\pi(t)$, and $Q_\pi = \lim_{t\to\infty} Q_\pi(t)$.

Recall that due to abandonments, the system is ergodic so that the latter limits exist. Let us now define the ratio, c_π, of the stationary service level of customers A over that of customers B. It is given by $c_\pi = \frac{Q^A_\pi}{Q^B_\pi}$.

We define the mean waiting time in queue for class m and the overall mean waiting time in queue for all customer types similarly. Note that we only define these quantities for the customers who enter service. Under a given scheduling policy π, let $w^m_{q,\pi}(i, t)$ be the waiting time in queue of the i^{th} type m customer who enters service, $0 \le i \le b_\pi^m(t)$. As in the usual way, the mean waiting time in queue $W_{q,\pi}^m(t)$ for type m customers during $[0,t]$ is defined by

$$W_{q,\pi}^m(t) = \frac{1}{b_\pi^m(t)} \sum_{i=1}^{b_\pi^m(t)} w_{q,\pi}^m(i,t). \tag{1}$$

Also, we define the overall mean waiting time in queue, during the interval $[0,t]$ by

$$W_{q,\pi}(t) = \frac{1}{b_\pi^A(t) + b_\pi^B(t)} \left(\sum_{i=1}^{b_\pi^A(t)} w_{q,\pi}^A(i,t) + \sum_{i=1}^{b_\pi^B(t)} w_{q,\pi}^B(i,t) \right). \tag{2}$$

During the stationary regime, the mean waiting time in queue for type m customers, say $W_{q,\pi}^m$, and the overall mean waiting time in queue for all customers, say $W_{q,\pi}$, are given by $W_{q,\pi}^m = \lim_{t\to\infty} W_{q,\pi}^m(t)$, and $W_{q,\pi} = \lim_{t\to\infty} W_{q,\pi}(t)$.

Now, we are going to underline one important property of performance measures. It is the variance of the waiting time. Minimizing the number of customers lost, or minimizing the mean waiting time in queue are only some important properties among many others. It has often been argued that a system with reasonable and predictable waiting times may be more desirable than a system with a lower mean waiting time but a higher variance. We refer the reader to Lu and Squillante (2004) for more details.

As above, we define the variance of the waiting time for type m customers and an overall variance for all types, in both transient and stationary regimes. The variance $Var_\pi^m(t)$ for type m customers during $[0,t]$ is defined by

$$Var_\pi^m(t) = \frac{\sum_{i=1}^{b_\pi^m(t)} \left(w_{q,\pi}^m(i,t) - W_{q,\pi}^m(t) \right)^2}{b_\pi^m(t)}. \tag{3}$$

We define the overall variance of the waiting time in queue for the customers who enter service during the interval of time $[0,t]$ by

$$Var_\pi(t) = \frac{\sum_{m\in\{A,B\}} \sum_{i=1}^{b_\pi^m(t)} \left(w_{q,\pi}^m(i,t) - W_{q,\pi}(t) \right)^2}{\sum_{m\in\{A,B\}} b_\pi^m(t)}. \tag{4}$$

During the stationary regime, the variance for type m customers, say Var_π^m, and the overall variance for all types, say Var_π, are given by $Var_\pi^m = \lim_{t\to\infty} Var_\pi^m(t)$, and $Var_\pi = \lim_{t\to\infty} Var_\pi(t)$.

Finally, we define the standard deviation, in transient and stationary regimes, for each type and for all types by taking the square root of the variances defined above. We denote these quantities by $\sigma_\pi^m(t)$, $\sigma_\pi(t)$, σ_π^m and σ_π, respectively.

In what follows, we present some results about the relation between the performance measures of interest and the discipline of service. Let us recall a known result for queueing system with infinitely patient customers, that is, a customer never leaves the queue before beginning service. It is well known that the expected time in system and expected time in queue are independent of the queue discipline. We only need to assume that the remaining total service or work required at any point during an arbitrary busy period is order-of-service independent. In other words, no service needs are created or destroyed within the system: no renege in the midst of service, no preemption when service times are not exponentially distributed, no forced idleness of servers, and so on. The proof can be easily done by comparing the diagrams of the cumulative work for two different queue disciplines during the busy period, elsewhere both systems behaves identically owing to the workconserving property.
We should note that one may find counterexamples for the above result if service times are not assigned when service begins, see Whitt (1992). However, it is common that service times are associated upon customer arrivals. In such cases, one may still assume that service times are i.i.d. and independent of the arrival process to get the result. We refer the reader to Berger and Whitt (1992) for more discussion.

We investigate below some conservation results in a more general queueing system including reneging. By means of Theorem (1), we motivate our consideration for only workconserving policies. Theorem (1) concerns a $GI/M/s+M$ system, which has an i.i.d. and generally distributed interarrivals, exponential service times, s servers, and exponentially distributed patience times.

Theorem 1 *Consider a $GI/M/s+M$ system with non-preemptive service discipline. the fraction of abandoning customers is equal or higher under non-workconserving policies in comparison to workconserving policies.*

The proof is given in Appendix B.

In what follows, we emphasize a number of theorems concerning workconserving policies. In Theorems (2) and (3), we investigate the conservation of the fraction of abandoning customers and the average waiting time in queue with respect to the scheduling policies, respectively. Some consequences are next derived in Corollaries (1) and (2).

Theorem 2 *Consider a $GI/GI/s+M$ queue. Times before reneging are assumed to be i.i.d. and exponentially distributed. Then, the fraction of abandoning customers, is constant for any workconserving non-preemptive scheduling policy.*

The proof is given in Appendix B.

Note that the result in Theorem (2) does not hold if service times are order of service dependent, or if we allow preemption when service times are not exponentially distributed, or if times before reneging are not identically and exponentially distributed. The proof of the conservation of Q for any workconserving policy (with or without preemption) can be easily obtained from Lemma (2) in Jouini and Dallery (2006). In the latter, the authors prove the result for a $GI/M/s/K+M$ queue with limited waiting space.

In Theorem (3), we focus on the conservation of the waiting time in queue with respect to workconserving non-preemptive scheduling policies. We consider again a $GI/GI/s+M$ queue, and we focus on three different definitions of the average waiting time. Let W_q be the average waiting time in queue for served customers. Let W_q^{ab} be the one for abandoning customers, i.e., the average spending time in queue before leaving the system without being served. Finally, we define W_q^{tot} as the overall waiting time in queue for all customers, i.e., served as well as abandoning customers. In Theorem (3), we prove an intuitive result for the conservation of W_q^{tot}. In addition, we show a counter-intuitive result for W_q and W_q^{ab}. Although the number of abandonments as shown in Theorem (2) does not vary for any workconserving non-preemptive scheduling policy, W_q and W_q^{ab} do vary.

Theorem 3 *Consider a $GI/GI/s+M$ queue. Times before reneging are assumed to be i.i.d. and exponentially distributed. When considering the class of workconserving non-preemptive scheduling policies, the following holds*

1. *W_q^{tot} does not depend on the scheduling policy.*
2. *W_q and W_q^{ab} depend on the scheduling policy.*
3. *The upper (lower) bound of W_q is achieved under the FCFS (LCLS) discipline of service.*
4. *The upper (lower) bound of W_q^{ab} is achieved under the LCLS (FCFS) discipline of service.*

The proof is given in Appendix B.

Note also that although the first moment W_q^{tot} does not depend on the discipline of service, the second moment of the overall waiting time and thus the full distribution does depend on the discipline of service. As shown in Theorem (3), the maximum of the average waiting time for served customers, W_q, is achieved under FCFS discipline. However, we conjecture, based on a well known property in queueing literature, that the minimum of its variance is also achieved under the FCFS policy. In practice, if the value of W_q under the FCFS policy is not too far from that under another policy, thereafter, a call center manager will usually prefer the FCFS policy owing to its fairness.

We refer the reader to Avi-Itzhak and Levy (2004) for more details on the fairness property in queueing systems.

We finally comment that the result in Theorem (3) is still valid when considering also preemptive scheduling policies, however, service times have to be exponentially distributed.

Corollary 1 *Consider a $GI/GI/s{+}M$ queue with two classes of customers A and B. Service times and times before reneging are identically distributed for both types of customers. Then, the overall service level Q and the overall expected waiting time W_q^{tot} are constant for any workconserving non-preemptive scheduling policy.*

The proof is given in Appendix B.

In the case of a two-class queue, we denote by π_A and π_B the policies that give strict non-preemptive priority to type A and B customers, respectively.

Corollary 2 *Consider a $GI/GI/s + M$ queue with two classes of customers A and B. Service times and times before reneging are identically distributed for both types of customers. Then, for any workconserving non-preemptive policy, π, the achieved service level ratio in the stationary regime, c_π, satisfies the following relation*

$$c_{\pi_A} \leq c_\pi \leq c_{\pi_B}, \tag{5}$$

The proof is given in Appendix B.

Note that values of c_π ranging out of the interval $[c_{\pi_A}, c_{\pi_B}]$ may be achieved through non-workconserving policies, such as thresholds or reservations policies. These policies are indeed useful to discriminate between customer classes. Under such policies, the lower bound for Q^A (Q^B) is equal to that achieved under the policy that gives strict preemptive priority to type A (B) customers. Obviously, the upper bound for Q^A or Q^B is 1. It is reached for a given type by simply refusing service for all customers of that type.

With the waiting time being defined as the waiting time of all customers (including served and abandoned customers), the following theorem holds even for non-workconserving policies.

Theorem 4 *Consider a $GI/GI/s{+}M$ queue, with two classes of customers A and B, and working under a given scheduling policy π. Service times and times before reneging are identically distributed for both types of customers. Let c_π be the stationary achieved ratio between the fractions of abandoning customers. Then, the stationary ratio between the average waiting times is also c_π.*

The proof is given in Appendix B.

After this preliminary study, we are now ready to tackle the main part of this paper, i.e., developing the scheduling policies.

3.3 Goal and Motivation

In this section, we motivate our objective with regard to the scheduling policies we aim to find. Before that, we highlight some known limits of the fixed strict priority policy. Consider the policy that assigns customers A to the high priority queue 1 and customers B to the low priority queue 2. It is well known that this policy makes the system highly unbalanced, and the QoS of customers B tends to be very low. Such behavior is undesirable for a call center manager. Unfortunately as mentioned before, in practice the workload is either underestimated, or overestimated due to the highly uncertain environment of call centers. In both cases, the staffing step is incorrectly done, and as a consequence, service levels will be much more affected. Let us give an explanation. When the workload is underestimated, the QoS of customers B deteriorates even more. The system becomes less and less stable for that class, since most of the time, the service capacity will be assigned to handle type A customers. The unfairness is still valid when the workload is overestimated. Customers of type A get a service level that approaches the maximum while there is really no need for that. In such cases, a policy providing a slightly lower service level for customers A and a higher level for customers B would be more valuable.

Having in mind above arguments, we formulate in the following our objective. We assume that scheduling of agents has already taken place, such that the number of available agents is known in advance. We aim to develop scheduling policies allowing, even in the case of unfavorable situations, to reach a fixed balance, during the stationary regime, for both service levels independently of the available service capacity. We should note that such policies need no information about the arrival process. The service level we consider is the fraction of abandoning calls, i.e, the probability to abandon. We also keep in mind the advantages of having lower variances of waiting times. In addition, we focus on studying simple policies that are easy to implement in practice, namely, priority scheduling policies. For practical reasons, we apply some simplifications to the general model. We assume that for each class of customers, interarrivals are i.i.d. and exponentially distributed. We assume also that service times are i.i.d. and exponentially distributed. We will not focus on the overall achieved fraction of abandoning customers, because anyway, this quantity is unchanged for any workconserving policy. In concrete terms, we specify our objective as follows. We aim to develop simple scheduling policies that minimizes the variance of the waiting time in queue of VIP customers, subject to satisfying a target ratio, c^*, of customer classes service levels.

4 Real-Time Scheduling Policies

The call center we consider here allows for flexible scheduling through dynamic alternate routing and sequencing, hereafter referred to as dynamic scheduling. Actually, this is easily possible for most call centers due to the technology development of their equipments. However, an interesting and challenging problem is to design scheduling policies that are simple to implement and their performances are acceptable for a call center manager. In this section, we restrict ourselves to develop simple online scheduling policies allowing to achieve an objective ratio, c^*. We consider several techniques that do take transient service levels into account and, hence, can be classified as online updating methods and real-time routing. The principle of our policies is that we adjust them during the evolution of the process. The adjustments depend on the history of the process.

Without loss of generality, we only consider an objective ratio, such that, $c^* < 1$. In fact, as we shall explain in the proof of Theorem (5), the case $c^* = 1$ reduces to the case of the FCFS discipline of service. As for the case $c^* > 1$, it is on the one hand not relevant for our analysis because we must keep in mind that type A customers are VIP. On the other hand, even if we would like to investigate that case, it suffices to apply the analysis for the case $c^* < 1$, by exchanging customers A by customers B and vice versa.

We propose several dynamic scheduling policies belonging to two different families. The first is the class of queue joining policies, which consists of assigning arriving customers to one of the queues. When considering such policies, we assume in addition that customers in queue 1 have priority over customers in queue 2 in the sense that agents are providing assistance to customers belonging to queue 1 first. The priority rule is non-preemptive, which simply means that an agent currently serving a customer pulled from queue 2, while a new arrival customer joins queue 1, will complete this service before turning to queue 1 customer. Note that the priority rule in call centers is non-preemptive. It is not common to interrupt the service of a customer to let another one with higher priority start service. The second family of policies is a dual to the first class. It consists in server allocation policies. There is no fixed priority between both queues. Whenever a server becomes idle, it has to decide which customer from the queues, if any, should be selected for service. The proposed queue joining and call selection policies are given next in Sections 4.1 and 4.2.

4.1 Queue Joining Policies

We propose three scheduling policies, say π_1, π_2 and π_3. The policies are belonging to the class of queue joining policies. We consider dynamic (state-dependent) policies, in the sense that upon arrival, a policy determines a rule for the queue assignment. Our policies do not anticipate on future events. They just react to the realization of the ratio that is determined by the history of the process. Moreover, the proposed policies are easy to understand and implement in practice. Based on simulation experiments, a comparison analysis with regard to the variance of the waiting time is thereafter addressed in Section 5.

Scheduling Policy π_1. The scheduling policy π_1 starts work identically as a strict priority policy giving the higher priority to customers A. After the epoch when the first type B customer finishes

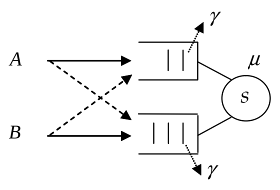

Figure 2: Scheduling policy π_1

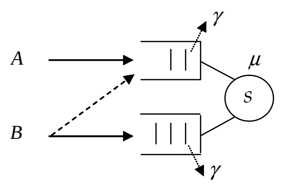

Figure 3: Scheduling policy π_2

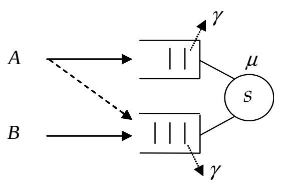

Figure 4: Scheduling policy π_3

his service, we apply the following assignment rule for any new arrival, which we denote by the k^{th} arrival. Let D_k be the epoch of that arrival. Let Q_k^A (Q_k^B) be the achieved service level from $t = 0$ until D_k for customers A (B.) Let c_k be the achieved ratio from $t = 0$ until D_k, $c_k = Q_k^A/Q_k^B$. If $c_k < c^*$, then we give high priority to customers B, that is, if the new arrival is type A, it is routed to queue 2, otherwise, it is routed to queue 1. However if $c_k \geq c^*$, we give high priority to type A customers, that is, if the new arrival is type A, it is routed to queue 1, and if it is type B, it is routed to queue 2. The scheduling policy π_1 is shown on Figure 2.

Scheduling Policy π_2. The scheduling policy π_2 starts work identically as π_1 until the time when the first customer B finishes service. Following the same notations as in the last paragraph, let a new arrival enter the system. Under π_2, a customer A is always routed to queue 1. However, the assignment rule of customers B is as follows. If $c_k < c^*$, a new type B arrival is routed to queue 1, otherwise if $c_k \geq c^*$, it is routed to queue 2. The scheduling policy π_2 is shown on Figure 3.

Scheduling Policy π_3. The scheduling policy π_3 starts work identically as policies π_1 and π_2 until the first customer B finishes service. Again following the same notations, let a new arrival enter the system. Under π_3, a customer B is always routed to queue 2. However, the assignment rule of customers A is as follows. If $c_k \geq c^*$, then a new type A arrival is routed to queue 1, otherwise if $c_k < c^*$, it is routed to queue 2. The scheduling policy π_3 is shown on Figure 4.

The scheduling policy π_1 can be immediately obtained intuitively. It allows the achieved ratio to be updated upon each arrival such that it converges in the long-run to the objective. The idea behind policy π_2 is that we keep always customers A in the high priority queue, however when it is necessary, we assign customers B to this queue to improve their service level (which deteriorates the service level of customers A.) Such rule allows to increase the transient ratio and to keep it close to the objective. As a consequence, the ratio would converge in a distant future to the desired value. Policy π_3 can be viewed as another variant. It allows some times to penalize customers A by assigning them to the low priority queue, which again allows to increase the transient ratio. Our choice for policies π_2 and π_3 comes from the assumption, $c^* < 1$.

In Theorem 5, we give some properties about the achieved ratios under the policies we propose.

Theorem 5 *Using the above notations, the following holds.*

1. *π_1 reaches c^* if and only if $c_{\pi_A} \leq c^* \leq c_{\pi_B}$.*
2. *π_2 reaches c^* if and only if $c_{\pi_A} \leq c^* \leq 1$.*
3. *π_3 reaches c^* if and only if $c_{\pi_A} \leq c^* \leq 1$.*

The proof is given in Appendix B.

One may construct several auxiliary policies similar to the above ones. For example instead of changing the priority rule at each new arrival epoch, we only change it at the arrival epoch of the customer who finds all servers busy and both queues empty. Then, we continue that rule until the end of the current busy period. With regard to reaching the stationary target ratio, the latter class of policies has the same properties as those for the class of policies π_1, π_2 and π_3. One drawback could be that they are less reactive to correct the transient ratio. A further possibility is to construct similar policies by changing the priority rule cyclically; at a given arrival and based on the transient ratio, we determine the priority rule and we continue it for a given fixed number of the next following arrivals. Once the cycle finishes, we determine the priority rule at the epoch of the arrival that follows the cycle. Again, we keep that rule for the same given fixed number of new arrivals, and so on.

4.2 Call Selection Policies

In the following, we focus on a dual class of policies, namely the class of call selection policies. We assign all type A (type B) customers to queue 1 (queue 2). On the contrary to the previous class of policies, we do not implement a priority discipline between both queues. Jobs from the waiting queues are selected by using so-called waiting time factors, as described for example in Lu and Squillante (2004). These selections occur after each service completion of an agent, when there is at least one non-empty queue. The choice about the next call that is served depends on waiting time factors, of which exactly one is associated with each queue. When selecting a job, the agent considers the first job in each queue. From this set of jobs, he chooses the job of which the product of the waiting time and the waiting time factor is the highest. The factor is queue-dependent only.

An advantage of waiting time factors is their flexibility with regard to the many different routing policies that are possible. Setting both waiting times equal results in a policy that serves jobs from both classes in a FCFS order, and by taking one waiting time factor equal to 0, one of the types has full priority above the other, independent of the waiting times.

The waiting time factors are adjusted dynamically, for which we consider different methods. Let us denote these factors by α_A and α_B for queues 1 (waiting type A customers) and 2 (waiting type B customers.)

The first method that we consider has one parameter, denoted by β, with $\beta \in [0,1]$. At each point in time, the waiting time factor of one queue is set to β, while the other waiting time factor is set to 1. Thus, there are two possibilities at each point in time, namely $(\alpha_A(t), \alpha_B(t)) \in \{(1,\beta), (\beta, 1)\}$.

Updates of the waiting time factors occur as follows. After each event the service level of each type is computed by considering the historical events. These quantities are calculated by dividing the number of abandoned customers by the number of arrivals. The service levels determine whether or not the waiting time factors are interchanged. If the ratio between the two service levels at time t exceeds c^*, the waiting time factors are interchanged, i.e., (α_A, α_B) is set to (α_B, α_A). If we define $Q^A(t)$ and $Q^B(t)$ as the transient service levels of each job types at time t, then we can write

$$(\alpha_A(t), \alpha_B(t)) \leftarrow \begin{cases} (1,\beta) \text{ if } Q^A(t) \geq c^* Q^B(t) \\ (\beta,1) \text{ if } Q^A(t) < c^* Q^B(t) \end{cases} . \tag{6}$$

Reasoning as in the proof of Theorem (5), we easily state that waiting time factors between 0 and 1 reach any objective between the two most extreme cases in which one of both types has full priority, namely c_{π_A} and c_{π_B}. In addition, we conjecture that for each feasible objective c^*, it exists $\beta_{c^*} \in [0,1]$ so that choosing any value in $[0, \beta_{c^*}]$ and assigning it to the parameter β allows to reach c^*. Any value beyond β_{c^*} could not allow to reach c^*. Just think of what happens near the boundary $\beta = 0$. In this case, the jobs of one type gets all priority if the waiting time factor $\alpha_A(t)$ and $\alpha_B(t)$ are not swapped. When β increases, jobs of the other type also get some priority, such that the difference in priority and the expected number of abandonments between both types decreases. At a certain point, while increasing β, the difference is so small that the objective c^* can no longer be reached.

This type of policy has several appealing properties. In the first place, it takes the empirical waiting times of the customers into account. Thus, if a customer waits very long it is seved, depending on the waiting time factor that indicates the importance of the customer type. In the second place, customers of each call type are served according to a FCFS policy because there are separate queues for the call types. For that reason, the policy is fair with respect to the order of service. In the third place, the reactivity of the policy is high, compared to the policies from

Section 4.1, because if customers of a certain type are available in a queue they can immediately be served, without serving customers of the other type first.

Alternatively, another method for updating the waiting time factors is to keep α_A fixed over time and to update α_B dynamically. Under the assumption that α_A is fixed, it is straightforward to decrease α_B when Q^A is high, and increase α_B otherwise. A difficulty is to develop a method that updates α_B in a simple but effective way, without many control parameters. We investigated several possibilities by means of numerical experiments. As a result, we decided to decrease the parameter α_B by multiplication by η and decrease it by division by η, with $0 < \eta \leq 1$. The factor α_B was updated after each event. The conclusion of these experiments is that we could not find a policy that performed better than the first method described in this section.

5 Simulation Results

Even though the workconserving scheduling policies we present here, lead to the target ratio and does not affect the overall fraction of abandoning calls, it will affect the waiting time variance. Because certainty is usually preferred to uncertainty, minimizing the variance is a logical basis for selecting the appropriate policy. A low variance is very useful when we want to estimate and inform customers about their queueing delays, see Armony et al. (2005). In fact, deriving the state-dependent estimation is too complicate, even more for our context here. The best we can do could be computing the mean value of the state-dependent waiting time. Then, a lower variance will give more credibility to the anticipated mean waiting times. A further advantage for a system with a low waiting time variance is related to its inherent fairness with regard to customers delays. Serving customers within comparable delays represents indeed an important issue for both actors, the manager and the customer.

Both analytic and numeric methods are too complex for a direct analysis of the policies comparison. Hence, we resort to simulation experiments to prove their efficiency and gain useful guidelines. We consider six systems, denoted by System 1, ..., System 6. Systems parameters are chosen so as we get realistic scenarios. The number of servers is $s = 50$. The common service rate is $\mu = 0.2$, i.e., the mean service time for one customer is 5 minutes. The common reneging rate is $\gamma = 0.33$. From one system to another, we vary the total arrival rate so as we get different "service utilizations", $\frac{\lambda_A+\lambda_B}{s\mu}$. We choose, $\lambda_A = \lambda_B = 5$, 6, 7, 9, 11, and 13, respectively. The "service utilization" is increasing (starting from 100% in system 1 until 260% in system 6.) We consider balanced cases for both arrival processes so that we facilitate the comment of the simulation results. Recall that abandonments make our systems unconditionally stable. The simulations are done for the target ratios $c^* = 0.5$, 0.7 and 0.9. We determined for each system the interval $[c_{\pi_A}, c_{\pi_B}]$, and we checked that the values $c^* = 0.5$, 0.7 and 0.9 are ranging in all of these intervals. Note that for the symmetric cases we consider ($\lambda_A = \lambda_B$), we easily see that $c_{\pi_A} = \frac{1}{c_{\pi_B}}$. For queue joining policies, we give the performance measures under policies π_1, π_2 and π_3, as well as those under policy π_A (high priority for type A customers.) As for call selection policies, we give the performance measures by choosing the parameter β equal to 0, 1/4, 1/2, 3/4, and 1. To be coherent we only look at values of β that allow us to reach the objective.

5.1 Experiments for the Queue Joining Policies

The simulation results are presented in Table 5.1 below for $c^* = 0.7$, and in Tables A and A of Appendix A for $c^* = 0.5$ and 0.9, respectively. The rows, corresponding to the quantity c, are to indicate the achieved stationary ratio under each scheduling policy.

		π_A	π_1	π_2	π_3		π_A	π_1	π_2	π_3
	c	0.254	0.700	0.700	0.700		0.189	0.700	0.700	0.700
	Q^A	2.611%	5.216%	5.225%	5.222%		5.598%	14.597%	14.516%	14.516%
	Q^B	10.076%	7.452%	7.464%	7.460%		29.649%	20.853%	20.737%	20.737%
Sys 1:	Q	6.343%	6.334%	6.344%	6.341%	Sys 2:	17.623%	17.724%	17.626%	17.626%
$\lambda_A = 5$	W^A	0.076	0.139	0.148	0.147	$\lambda_A = 6$	0.167	0.389	0.438	0.432
$\lambda_B = 5$	W^B	0.266	0.196	0.199	0.205	$\lambda_B = 6$	0.889	0.574	0.577	0.624
	W	0.167	0.167	0.174	0.176		0.475	0.478	0.505	0.524
	σ^A	0.139	0.331	0.266	0.274		0.187	0.634	0.428	0.463
	σ^B	0.516	0.428	0.397	0.380		0.928	0.793	0.679	0.622
	σ	0.383	0.383	0.338	0.332		0.718	0.721	0.567	0.554
	c	0.153	0.700	0.700	0.700		0.146	0.700	0.700	0.700
	Q^A	7.601%	23.662%	23.607%	23.605%		11.300%	36.610%	37.284%	36.691%
	Q^B	49.733%	33.802%	33.724%	33.721%		77.587%	52.300%	53.263%	52.416%
Sys 3:	Q	28.666%	28.731%	28.665%	28.663%	Sys 4:	44.441%	44.446%	45.273%	44.553%
$\lambda_A = 7$	W^A	0.229	0.614	0.761	0.740	$\lambda_A = 9$	0.345	0.812	1.319	1.221
$\lambda_B = 7$	W^B	1.767	0.955	0.966	1.116	$\lambda_B = 9$	3.804	1.349	1.603	1.974
	W	0.771	0.772	0.856	0.915		1.043	1.043	1.440	1.544
	σ^A	0.217	0.933	0.525	0.612		0.292	1.379	0.693	0.892
	σ^B	1.243	1.159	0.851	0.774		1.792	1.861	0.969	1.019
	σ	1.056	1.058	0.703	0.717		1.626	1.626	0.834	1.019
	c	0.185	0.700	0.700	0.700		0.260	0.700	0.700	0.700
	Q^A	17.034%	44.920%	44.920%	44.919%		25.586%	50.685%	51.043%	50.673%
	Q^B	92.083%	64.172%	64.172%	64.170%		98.242%	72.407%	72.919%	72.390%
Sys 5:	Q	54.557%	54.547%	54.546%	54.545%	Sys 6:	61.916%	61.545%	61.981%	61.531%
$\lambda_A = 11$	W^A	0.534	0.825	1.669	1.553	$\lambda_A = 13$	0.846	0.923	1.969	1.844
$\lambda_B = 11$	W^B	5.973	1.289	2.124	2.565	$\lambda_B = 13$	8.089	1.146	2.712	2.968
	W	1.008	1.008	1.848	1.952		1.014	1.003	2.234	2.248
	σ^A	0.388	1.402	0.852	1.078		0.484	1.027	1.025	1.150
	σ^B	2.466	2.124	1.010	1.173		3.148	1.695	0.958	1.082
	σ	1.738	1.738	0.944	1.221		1.281	1.311	1.063	1.249

Table 1: Simulation experiments for $c^* = 0.7$ - queue joining policies

In what follows, we discuss the simulation results. As expected, the total fraction of abandoning calls, Q, is independent of the scheduling policy. The negligible deviations in the values presented in Tables 5.1, A and A are due to the simulation duration. These quantities will necessarily coincide when running the simulations for a very long duration. We check from the experiments that the target ratio is always met by policies π_1, π_2 and π_3, which agrees with Theorem (5). For each system, the value of the ratio under policy π_A represents a lower bound for the achievable ratio under any workconserving non-preemptive scheduling policy. We can not do better when considering that class of policies.

In what follows, we address a comparison analysis with regard to the standard deviation of the waiting time in queue. We do not conduct a rigorous analysis in the sense that we do not prove our statements. Such a work is of great value. We leave it for a future research. However, we give here some general ideas and intuitive explanations to support the claims we derive.

For type A customers, starting from the lower value, most experiments show that the standard deviation values are structured in turn for policies π_A, π_2, π_3 and π_1. Let us give an intuitive explanation. The reason is basically related to the well known property in queueing theory which claims that the FCFS discipline minimizes waiting time variance (time in queue and in system) when the queue discipline is service time independent. We refer the reader to Randolph (1991) for more discussion. The best we can do for customers A under a workconserving non-preemptive policy is to not give at any time the higher priority to customers B. Such situation allows customers-A waiting times to be as low as possible. This is the case for policy π_A. Next, since the discipline of service within queue 1 is FCFS, then π_A should lead to the lower variance. With regard to the order of service of customers A, policy π_1 deviates more than π_2 and π_3 from the FCFS discipline. This tells us that π_1 has the higher variance. When comparing policies π_2 and π_3, one may see that on the contrary of π_3, policy π_2 respects the FCFS order for customers A, which indicates a lower variance than under π_3.

For type B customers, starting from the lower value, we conclude from the majority of the experiments that the standard deviation values are structured in turn for policies π_3, π_2, π_1 and π_A. When comparing policies π_1, π_2 and π_3, the explanation is identical to that conducted for the first comment. As for policy π_A, the only explanation we have is related to the waiting time values. The larger waiting times of type B customers are achieved under π_A, because of their lower priority. However due to uncertainty in arrivals, reneging and service times, there is a non-zero probability that some customers B enter service without waiting or within a short delay. This allows their waiting times variance to be as a consequence the higher.

Recall that a further investigation is required. We only gave some directions to compare the achieved values of variances. For instance, the comparison should lie in the values of λ_A and λ_B, also in the objective c^*. An objective close to 1 would make our policies work in a similar manner than that of the FCFS discipline, whereas an objective far from 1 (being under or beyond) would make the policies similar to the strict priority policy. Based on the analysis here, we can not distinguish a best policy. However, one may recommend policy π_2. First, it reaches the objective ratio. Second, it gives (in most cases) the lower variance of VIP customers waiting times. Third, it allows to have a “good” variance for customers B, as well as for all customer types.

5.2 Experiments for the Call Selection Policies

The simulation results are presented in Table 5.2 below for $c^* = 0.7$, and in Tables A and A of Appendix A for $c^* = 0.5$ and 0.9, respectively. The rows (corresponding to the quantity c) are to indicate the achieved stationary ratio under different values of β.

		β=0	β=1/4	β=1/2	β=3/4	β=1		β=0	β=1/4	β=1/2	β=3/4	β=1
	c	0.700	0.700	0.700	0.826	1.001		0.700	0.700	0.700	0.815	1.000
	Q^A	5.211%	5.209%	5.203%	5.698%	7.101%		14.523%	14.478%	14.501%	15.802%	17.768%
	Q^B	7.444%	7.441%	7.433%	6.898%	7.094%		20.747%	20.683%	20.716%	19.389%	17.764%
Sys 1:	Q	6.328%	6.325%	6.318%	6.298%	7.098%	Sys 2:	17.635%	17.581%	17.609%	17.596%	17.766%
$\lambda_A = 5$	W^A	0.142	0.148	0.151	0.166	0.204	$\lambda_A = 6$	0.404	0.435	0.449	0.492	0.552
$\lambda_B = 5$	W^B	0.200	0.207	0.210	0.197	0.204	$\lambda_B = 6$	0.592	0.628	0.643	0.605	0.552
	W	0.171	0.177	0.180	0.181	0.204		0.495	0.528	0.542	0.547	0.552
	σ^A	0.321	0.275	0.252	0.275	0.325		0.596	0.462	0.394	0.417	0.475
	σ^B	0.421	0.387	0.370	0.339	0.325		0.755	0.644	0.591	0.533	0.475
	σ	0.375	0.336	0.317	0.309	0.325		0.684	0.565	0.508	0.481	0.475
	c	0.700	0.700	0.700	0.814	1.000		0.700	0.700	0.700	0.825	1.000
	Q^A	23.644%	24.077%	23.602%	25.811%	28.690%		36.769%	36.627%	36.671%	40.258%	44.432%
	Q^B	33.777%	34.396%	33.717%	31.709%	28.695%		52.527%	52.324%	52.387%	48.796%	44.365%
	Q	28.711%	29.237%	28.660%	28.760%	28.693%		44.648%	44.476%	44.529%	44.527%	44.399%
Sys 3:	W^A	0.659	0.766	0.780	0.868	0.976	Sys 4:	0.943	1.249	1.343	1.519	1.732
$\lambda_A = 7$	W^B	1.012	1.153	1.163	1.093	0.976	$\lambda_A = 9$	1.528	2.000	2.148	1.957	1.732
$\lambda_B = 7$	W	0.823	0.945	0.958	0.976	0.976	$\lambda_B = 9$	1.195	1.571	1.688	1.721	1.732
	σ^A	0.860	0.608	0.456	0.466	0.533		1.291	0.818	0.505	0.477	0.545
	σ^B	1.075	0.835	0.695	0.607	0.533		1.723	1.164	0.796	0.626	0.545
	σ	0.982	0.747	0.610	0.550	0.533		1.520	1.049	0.759	0.592	0.545
	c	0.700	0.700	0.700	0.839	1.000		0.700	0.700	0.700	0.850	1.000
	Q^A	44.889%	45.423%	44.901%	50.211%	54.646%		51.523%	50.678%	50.715%	56.615%	61.474%
	Q^B	64.127%	64.890%	64.144%	59.846%	54.600%		73.604%	72.397%	72.450%	66.605%	61.512%
Sys 5:	Q	54.508%	55.157%	54.523%	55.029%	54.623%	Sys 6:	62.564%	61.538%	61.583%	61.610%	61.493%
$\lambda_A = 11$	W^A	1.010	1.631	1.760	2.073	2.342	$\lambda_A = 13$	1.097	1.889	2.101	2.488	2.846
$\lambda_B = 11$	W^B	1.573	2.740	2.986	2.695	2.342	$\lambda_B = 13$	1.393	3.270	3.773	3.248	2.846
	W	1.232	2.065	2.243	2.351	2.342		1.201	2.384	2.701	2.819	2.846
	σ^A	1.394	0.975	0.511	0.482	0.547		1.063	1.050	0.483	0.481	0.546
	σ^B	2.054	1.555	0.866	0.633	0.547		1.676	1.877	0.886	0.631	0.546
	σ	1.708	1.348	0.902	0.635	0.547		1.320	1.552	1.037	0.668	0.546

Table 2: Simulation experiments for $c^* = 0.7$ - call selection policies

The end of this section is devoted to a discussion of the simulation results. To this end, we discuss a number of observations.

Firstly, one can see that all quantities increase while the workload increases. In our opinion, this is easy to understand and needs no further explanation. Secondly, if β is high, for example 3/4 or 1, then the ratio c^* is not reached. The reason is that the weights of the waiting times are almost equal, such that the service priority of the customers among the two queues is close to a FCFS policy. Such high values of β does not sufficiently allow to discriminate one type of customers to the detriment of another. Both types have almost equal priority of service. In addition, the parameter β_{c^*} (introduced in Section 4.2) is increasing in c^*. The explanation is as follows. As c^* increases, giving the priority to one type over the other one is less and less needed. This allows β_{c^*} to be more and more high. In the limit case (for $c^* = 1$), β_{c^*} reaches its upper bound ($\beta_{c^*} = 1$.) Thirdly, we see from the experiments that the overall average waiting time of the served customers increases as β increases. The reason pertains to the fact that as β increases, the scheduling policy approaches the FCFS policy. As shown in Theorem (3), the FCFS policy maximizes the mean waiting time in queue. Meanwhile, the average waiting time of the abandoned customers decreases (because the average waiting time of both served and abandoned customers does not vary.) Fourthly, the variance of the waiting times (for all types) is decreasing in β. One intuitive explanation is due to the known property of the FCFS policy, which minimizes the waiting times variance. As β increases, the behavior of our policy approaches that of the FCFS policy, and as a consequence, the overall variance decreases. In the limit case ($\beta = 0$), we have an alternation for service selection which is the far one from the FCFS policy. As expected, its corresponding variances are the highest.

5.3 Comparison

In this section we compare both types of policies, the queue joining and call selection policies. We consider the call section policies with the highest parameter β while allowing to reach the target ratio, so that the comparison would be coherent. From the experiments, we point out two important conclusions. The first conclusion deals with the average waiting times of the served customers. The second one focuses on the achieved variances values.

As expected, the average waiting times of the served customers are shorter for the queue joining policies. This conclusion is valid for each types of customers, as well as for all types. The reason is dealing with the order under which customers are scheduled for service. Under the call selection policies, customers of each type are served in the order of their arrival. This allow the waiting times of each type to be the highest (see Theorem (3)). As a consequence, the mean waiting time for all served customers is also the highest.

The second conclusion is that the variances (equivalently the standard deviations) of the waiting times are lower for the call selection policies. The explanation is identical to the previous one and is again pertaining to the order of service of customers. This would make the call selection policies likely, nevertheless, computing the parameter β_{c^*} is far from being an easy task.

One may also compare both types of policies from the reactivity perspective. We mean by reactivity, the speed of a given policy to make the transient ratio $c(t)$ close to the objective. Although both types of policies allow to reach the objective ratio in the distant future, we comment that they have different reactivities. The call selection policies are more reactive. One intuitive explanation is as follows. When for example the transient ratio is lower than the objective, both policies give the higher priority to type B customers in order to increase the transient ratio. A queue joining policy as π_2 will assign a new customer B to the queue with higher priority, however it may happens that there are type A waiting customers in that queue ahead of the customer B of interest. In such case, the transient ratio would thereafter still decrease. He would increase once we finish to serve the type A customers waiting in the highest priority queue. This case does not occur under the call selection policies. Depending on the circumstances, the latter policies immediately allow to increase or decrease the transient ratio, by scheduling one given type for service.

The reactivity of the call selection policy strongly depends on the value of β. It is maximized for $\beta = 0$, such that customers of the call type with the highest expected abandonment rate are served first, compared to the ratio c^*. For higher values of β the waiting times of the customers are also taken into account, such that customers that have waited longer than others have a higher priority. This can yield actions that are sometimes not optimal with respect to the ratio. By means of the waiting time factors, a trade-off is made between serving the customer with the longest waiting time and serving the customer that improves the abandonment ratio. Sometimes the optimal actions of both objectives will be the same, but this will not always be the case.

6 Extensions

In this section, we discuss some extensions of the analysis of this work. In Section 6.1, we investigate the extension of the proposed queue joining policies to the case of three customer classes. In Section 6.2, we tackle that extension for the call selection policies.

6.1 Extension for the Queue Joining Policies

In this section, we focus on the extension of the queue joining policies to the case of three customer classes. We consider a generalization of our call center queueing model with three customer types A, B and C. In addition, the queueing model has three infinite queues denoted by queue 1, 2, and 3. Our basic goal here is to discuss the usefulness of our online policies subject to reaching a proportionality on the service levels. We denote the stationary fractions of abandoning calls of type A, B and C by Q^A, Q^B and Q^C, respectively. In the case of three classes, we have to define two objective ratios. Without loss of generality, the first objective ratio, say c_1^*, is defined for types A and B by $c_1^* = Q^A/Q^B$. The second, say c_2^*, is defined for types B and C by $c_2^* = Q^B/Q^C$. As a consequence, the ratio Q^A/Q^C will be equal to $c_1^* \times c_2^*$, which we denote by c_3^*.

For workconserving non-preemptive policies, let S be the set of all feasible couples of ratios $(\frac{Q^A}{Q^B}, \frac{Q^B}{Q^C})$. Thus the following policy, say π_1', allows to reach any objective (c_1^*, c_2^*), such that $(c_1^*, c_2^*) \in S$.

Scheduling Policy π_1'. The scheduling policy π_1' is an extension of policy π_1. Initially, it starts work identically as a strict non-preemptive policy giving the higher priority to customers A, then B, and customers C have the lower priority. After the epoch when at least one type B customer and one type C customer have finished their service, we apply the following assignment rule for any new arrival, denoted by the k^{th} arrival. Let D_k be the epoch of that arrival. Let Q_k^A, Q_k^B and Q_k^C be the achieved service levels from $t = 0$ until D_k for customers A, B and C, respectively.
Let $c_{1,k}$, $c_{2,k}$ and $c_{,k}$ be the achieved ratios from $t = 0$ until D_k, i.e., Q_k^A/Q_k^B, Q_k^B/Q_k^C and Q_k^A/Q_k^C, respectively.

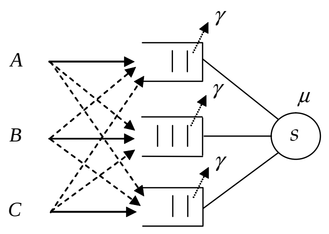

Figure 5: Scheduling policy π'_1

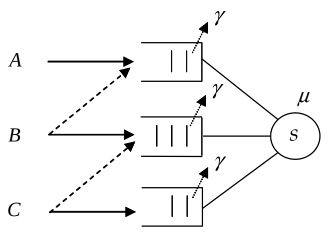

Figure 6: Scheduling policy π'_2

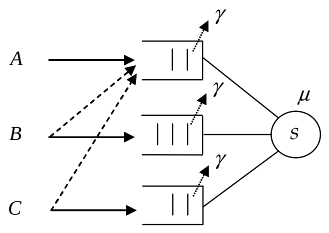

Figure 7: Scheduling policy π'_3

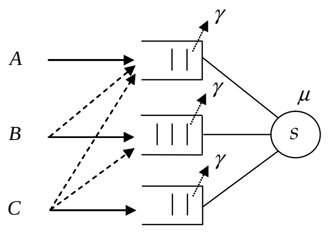

Figure 8: Scheduling policy π'_4

- If $c_{1,k} \geq c_1^*$, then we give the priority to type A over type B. Otherwise, we give the priority to type B over type A.
- If $c_{2,k} \geq c_2^*$, then we give the priority to type B over type C. Otherwise, we give the priority to type C over type B.
- If $c_{,k} \geq c_3^*$, then we give the priority to type A over type C. Otherwise, we give the priority to type C over type A.

From the previous tests, we therefore determine a strict priority level for each customer type. So, customers with the highest priority are assigned to queue 1. Those with the second higher priority are assigned to queue 2. Finally, those with the lower priority are assigned to queue 3. Thereafter based on its type, the new arrival is treated under this queue joining rule. The queue joining rule is updated for every new arrival. The scheduling policy π'_1 is shown on Figure 5.
For example, let a new arrival occur. Based on the history of the process, assume that we have $c_{1,k} < c_1^*$, $c_{2,k} \geq c_2^*$ and $c_{,k} < c_3^*$. Thus, starting from the highest priority, the priority levels are structured in turn for type B, C and A. Hence for the next arrival, a type B, C or A have to be assigned to queue 1, 2 or 3, respectively.

Let us remember that type A customers are more valuable for the company than type B ones, who are more valuable than type C customers. Then, both objective ratios c_1^* and c_2^* should be in practice strictly lower than 1. In such a case, the scheduling policy π'_2, shown on Figure 6, should perform better with regard to the variance of the waiting time in queue. However, it could not achieve any objective in S ($c_1^* > 1$ or $c_2^* > 1$.) One may also propose additional alternative policies, namely π'_3 (Figure 7) and π'_4 (Figure 8.) The latter policies are different of π'_2 in the sense that they allow to achieve values beyond 1 for c_2^*. Note that it would be interesting to investigate similar policies for models with only two queues instead of three. It would also be interesting to compare the proposed policies in this section with regard to the variance of the waiting time and the region of feasible objectives. We leave this work for future research.

6.2 Extension for the Call Selection Policies

Recall that Section 5.2 discussed call selection policies with waiting time factors for 2 customer types. There, an updating method was introduced to achieve ratios between the abandonment percentages. In this section we discuss a generalization of the policy of Section 5.2 to 3 customer types. The objective is the obtain a ratio between the abandonment percentages of the 3 job types. Conform the notation of Section 6.1, the abandonment percentages Q^A, Q^B, and Q^C are subject to converge to $Q^A/Q^B = c_1^*$ and $Q^B/Q^C = c_2^*$. In this section, we use other notation. We define the ratios $\tilde{c}_1^*$, $\tilde{c}_2^*$, and $\tilde{c}_3^*$, and the objective is that the ratio $Q^A : Q^B : Q^C$ converges to $\tilde{c}_1^* : \tilde{c}_2^* : \tilde{c}_3^*$. Note that the relation between this objective and the objective of Section 6.1 is that $\tilde{c}_1^* = 1$, $\tilde{c}_2^* = 1/c_1^*$, and $\tilde{c}_3^* = 1/(c_1^* c_2^*)$.

The order of service is determined by waiting time factors again, which are indicated at time t by the vector $(\alpha_A(t), \alpha_B(t), \alpha_C(t)) =: \alpha(t)$. The customer with the longest weighted (according to the factors) waiting time is served first.

There are several ways to update the waiting time factors. Our suggestion is to restrict the waiting time factors $\alpha_m(t)$ to be a permutation of $\{0, \beta, 1\} = H$ for all t, with $m \in \{A, B, C\}$.

There are different methods to update the waiting time factors. For example, the permutation of H among the queues could be controlled (dynamically) as follows. First, the total number of abandonments is counted over all types. Then, this number is divided among the job types in such a way that the abandonment percentages are proportional to the parameters c_m^*. Next, for each type the difference between on the one hand the percentages resulting from this calculation and on the other hand the real abandonment percentage observed until time t is calculated. Then, these differences are ordered increasingly and the elements in H are ordered increasingly. Finally, according to these orderings, the job type with the highest difference gets the highest waiting time factor, the job type with the second-highest difference gets the second-highest waiting time factor, and the type with the smallest difference gets the smallest waiting time factor.

We conjecture that all possible objectives can be met by using a permutation of $H = \{0, \beta, 1\}$, because each type can get full priority or no priority. However, minimization of the variances may require other waiting time factors, which is a subject for future research.

7 Conclusions and Further Research

We focused on a fundamental short-term problem for the management of call centers. We considered a two-class call center and developed real-time scheduling policies. The scheduling policies are belonging to the family of queue joining and call selection policies. We focused on service levels criteria related to the fraction of abandoning customers and the variance of the queueing delay. These policies are characterized to be relevant in practice. They are easy to understand for managers, predictable and easy to implement. Addressing exact analyzes for such problems is often too complex, and a challenging issue is to design dynamic scheduling policies that are simple, predictable and whose performance is good in an appropriate sense.

First, we gave some structural results in order to better understand the impact of scheduling policies on the performance measures of interest. Second, we proposed several dynamic scheduling policies allowing to meet a target ratio between the fractions of abandoning calls. Thereafter, we conducted a simulation study to compare the proposed policies with regard to the achieved variances of waiting times. Finally, we presented their extension to the case of three customer classes.

An interesting subject for future research would be to investigate static scheduling policies analogous to the queue joining policies. Let us give further details. The idea behind static policies comes from Aguir et al. (2006). In their work, the authors consider an identical model to the one described in Section 3.1, whereas they do not allow customers to renege while waiting in queue. They characterize a class of optimal static policies subject to satisfying differentiated performances for customer classes. The proposed policies are based on two parameters p_A and p_B. The quantity p_A (p_B) represents the static probability to assign new type A (B) arrivals to the queue with the highest priority, i.e., queue 1. Hence a new type A arrival is routed to queue 2 with probability $1-p_A$, and a new type B arrival is routed to queue 2 with probability $1-p_B$. Unfortunately, such analysis is untractable when considering reneging. Closed-form expressions for the quantities Q^A and Q^B are indeed not available. Even an exact numerical computation is not possible. Here we only give an alternative idea to tackle the problem and leave a rigorous analysis for future research.

Given an objective c^*, a possible method to get the probabilities p_A and p_B would be as follows. We simulate our system under a dynamic policy that achieves the objective ratio, as those proposed in Section 4.1. At the end of the simulation run, we take the proportions of types A and B customers assigned to queue 1. These quantities are thereafter assigned to p_A and p_B, respectively. It is easy to see for example that for policy π_2, $p_A = 1$, and for policy π_3, $p_B = 0$.

One would expect that the static policies yield to higher variances of the waiting times than those achieved under their corresponding online policies. This may be due to a known property in queueing theory. The general idea is that dynamic policies achieve the lower variance of the waiting time, then cyclic policies, and finally static policies (based on random assignment) achieve the higher values. However, we should note that static policies are easier to implement in practice.

In this paper, the analysis focus on a given period of the day. We want in a future study to focus on a whole call center day. it would be interesting to translate the performance objectives of the whole day into a set of objectives corresponding to the day periods.

References

Aguir, M., Karaesmen, F., and Dallery, Y. (2006). Head-Of-The-Line Priority with Random Assignment. Working paper, Ecole Centrale Paris, France.

Akşin, O. Z. and Karaesmen, F. (2002). Designing Flexibility: Characterizing the value of Cross-Training Practices. INSEAD, France, under revision.

Ancker, C. J. and Gafarian, A. (1962). Queueing with Impatient Customers Who Leave at Random. *Journal of Industrial Engineering*, 13:84–90.

Armony, M. (2005). Dynamic Routing in Large-Scale Service systems with Heterogeneous Servers. *Queueing Systems: Theory and Applications (QUESTA)*, 51:287–329.

Armony, M., Shimkin, N., and Whitt, W. (2005). The Impact of Delay Announcements in Many-Server Queues with Abandonment. Working paper, New York University.

Avi-Itzhak, B. and Levy, H. (2004). On Measuring Fairness in Queues. *Advances In Applied Probability*, 36:919–936.

Avramidis, A., Deslauriers, A., and l'Ecuyer, P. (2004). Modeling Daily Arrivals to a Telephone Call Center. *Management Science*, 50:896–908.

Baccelli, F. and Hebuterne, G. (1981). On Queues With Impatient Customers. *Performance'81 North-Holland Publishing Company*, pages 159–179.

Bassamboo, A., Harrison, J., and Zeevi, A. (2004). Dynamic Routing and Admission Control in High-Volume Service Systems: Asymptotic via Multi-Scale Fluid Models. Submitted for publication.

Berger, A. and Whitt, W. (1992). Comparisons of Multi-Server Queues with Finite Waiting Rooms. *Stochastic Models*, 8:719–732.

Brandt, A. and Brandt, M. (2002). Assymptotic Results and a Markovian Approximation for the M(n)/M(n)/C + GI System. *Queueing Systems: Theory and Applications (QUESTA)*, 41:73–94.

Gans, N., Koole, G., and Mandelbaum, A. (2003). Telephone Call Centers: Tutorial, Review, and Research Prospects. *Manufacturing & Service Operations Management*, 5:73–141.

Gans, N. and van Ryzin, G. (1998). Optimal Dynamic Scheduling of a General Class of Parallel-Processing Queueing Systems. *Advances in Applied Probability*, 30:1130–1156.

Gans, N. and Zhou, Y. (2003). A Call-Routing Problem with Service-Level Constraints. *Operations Research*, 51:255–271.

Garnett, O. and Mandelbaum, A. (2001). An Introduction to Skills-Based Routing and its Operational Complexities. Teaching notes, Technion.

Garnett, O., Mandelbaum, A., and Reiman, M. (2002). Designing a Call Center with Impatient Customers. *Manufacturing & Service Operations Management*, 4:208–227.

Gross, D. and Harris, C. (1998). *Fundamentals of Queueing Theory.* Wiley series in probability and mathematical statistics. 3rd edition.

Guérin, R. (1998). Queueing-Blocking System with two Arrival Streams and Guard Channels. *IEEE Transactions on Communications*, 36:153–163.

Halfin, S. and Whitt, W. (1981). Heavy-Traffic Limits for Queues with Many Exponential Servers. *Operations Research*, 29:567–588.

Huang, T. (2001). Analysis and Modeling of a Threshold Based Priority Queueing System. *Computer Communications*, 24:284–291.

Jongbloed, G. and Koole, G. (2001). Managing Uncertainty in Call Centers using Poisson Mixtures. *Applied Stochastic Models in Business and Industry*, 17:307–318.

Jouini, O. and Dallery, Y. (2006). Monotonicity Properties for Multiserver Queues with Reneging and Finite Waiting Lines. To appear in Probability in the Engineering and Informational Sciences.

Koole, G. and Pot, A. (2006). An Overview of Routing and Staffing Algorithms in Multi-Skill Customer Contact Center. Submitted for publication.

Lu, Y. and Squillante, M. (2004). Scheduling to Minimize General Functions of the Mean and Variance of Sojourn Times in Queueing Systems. Working Paper, IBM Research Division.

Mandelbaum, A., A., S., and Zeltyn, S. (2000). Empirical Analysis of a Call Center. Technical Report, Technion.

Mandelbaum, A. and Zeltyn, S. (2006). Staffing Many-Server Queues with Impatient Customers: Constraint Satisfaction in Call Centers. Working paper, Thechnion, Haifa, Israel.

Meketon, M. (1987). Newblock Optimization in Simulation: A Survey of Recent Results. *Proceedings of the 1987 Simulation Conference*, pages 58–67.

Örmeci, E. (2004). Dynamic Admission Control in a Call Center with One Shared and Two Dedicated Service Facilities. *IEEE Transactions on Automatic Control*, 49:1157–1161.

Pekoz, E. (2002). Optimal Policies for Multi-Server Non-Preemptive Priority queues. *Queueing Systems: Theory and Applications (QUESTA)*, 42:91–101.

Pierson, M. P. and Whitt, W. (2006). A Statistically-Fit Markovian Approximation of a Basic Call-Center Model. Working paper, Columbia University.

Pot, A. (2006). *Routing and Planning Algorithms for Multi-Skill Contact Centers.* Ph.D. Thesis, Vrije Universiteit Amsterdam, The Netherlands.

Randolph, W. H. (1991). *Queueing Methods for Services and Manufacturing.* Prentice Hall.

Schrage, L. and Miller, L. (1966). The Queue M/G/1 with the Shortest Remaining Processing Time Discipline. *Operations Research*, 14:670–684.

Ward, A. and Glynn, P. (2003). A Diffusion Approximation for a Markovian Queue with Reneging. *Queueing Systems: Theory and Applications (QUESTA)*, 43:103–128.

Whitt, W. (1992). Counterexamples for Comparisons of Queues with Finite Waiting Rooms. *Queueing Systems: Theory and Applications (QUESTA)*, 10:271–278.

Whitt, W. (1999). Improving Service by Informing Customers about Anticipated Delays. *Management Science*, 45:192–207.

Xu, S., Righter, R., and J.G., S. (1992). Optimal Dynamic Assignment of Customers to Heterogeneous Servers in Parallel. *Operations Research*, 40:1126–1138.

Zohar, E., Mandelbaum, A., and N., S. (2002). Adaptive Behavior of Impatient Customers in Tele-Queues: Theory and Empirical Support. *Management Science*, 48:566–583.

Appendixes

A Other Experiments

In this appendix, we present supporting simulation experiments for the analysis of Section 5. We consider the systems already chosen in Section 5. In Tables A and A, we show of the queue joining policies, the simulation results for $c^* = 0.5$ and $c^* = 0.9$, respectively. In Tables A and A, we show the results of the call selection policy for $c^* = 0.5$ and $c^* = 0.9$, respectively.

		π_A	π_1	π_2	π_3		π_A	π_1	π_2	π_3
	c	0.253	0.500	0.500	0.500		0.180	0.500	0.500	0.500
	Q^A	2.889%	4.349%	5.010%	4.270%		6.655%	12.281%	11.750%	11.973%
	Q^B	11.414%	8.699%	10.020%	8.541%		36.871%	24.562%	23.501%	23.946%
Sys 1:	Q	7.152%	6.524%	7.515%	6.406%	Sys 2:	21.763%	18.421%	17.626%	17.960%
$\lambda_A = 5$	W^A	0.084	0.118	0.143	0.121	$\lambda_A = 6$	0.199	0.327	0.350	0.348
$\lambda_B = 5$	W^B	0.303	0.229	0.263	0.231	$\lambda_B = 6$	1.149	0.693	0.639	0.716
	W	0.188	0.172	0.201	0.174		0.582	0.496	0.484	0.518
	σ^A	0.147	0.280	0.248	0.238		0.210	0.557	0.376	0.431
	σ^B	0.556	0.472	0.504	0.439		1.085	0.872	0.816	0.739
	σ	0.413	0.390	0.398	0.355		0.848	0.742	0.637	0.621
	c	0.153	0.500	0.500	0.500		0.146	0.500	0.500	0.500
	Q^A	7.601%	19.107%	19.110%	19.098%		11.300%	29.627%	29.628%	29.786%
	Q^B	49.733%	38.214%	38.219%	38.197%		77.587%	59.254%	59.255%	59.573%
Sys 3:	Q	28.666%	28.661%	28.665%	28.646%	Sys 4:	44.441%	44.440%	44.443%	44.681%
$\lambda_A = 7$	W^A	0.229	0.489	0.595	0.567	$\lambda_A = 9$	0.345	0.651	0.982	0.891
$\lambda_B = 7$	W^B	1.767	1.140	1.044	1.268	$\lambda_B = 9$	3.804	1.720	1.561	2.272
	W	0.771	0.771	0.790	0.870		1.043	1.043	1.194	1.396
	σ^A	0.217	0.798	0.481	0.597		0.292	1.148	0.646	0.894
	σ^B	1.243	1.225	1.118	0.938		1.792	2.050	1.485	1.343
	σ	1.056	1.056	0.850	0.839		1.626	1.625	1.073	1.269
	c	0.185	0.500	0.500	0.500		0.255	0.500	0.500	0.500
	Q^A	17.026%	36.363%	36.367%	36.358%		24.993%	41.028%	41.027%	41.030%
	Q^B	92.070%	72.725%	72.734%	72.715%		98.082%	82.055%	82.055%	82.060%
Sys 5:	Q	54.545%	54.545%	54.551%	54.536%	Sys 6:	61.543%	61.542%	61.542%	61.545%
$\lambda_A = 11$	W^A	0.534	0.709	1.248	1.095	$\lambda_A = 12$	0.824	0.876	1.442	1.303
$\lambda_B = 11$	W^B	5.967	1.705	1.997	2.895	$\lambda_B = 12$	8.005	1.422	2.483	3.217
	W	1.008	1.008	1.472	1.635		1.003	1.003	1.685	1.750
	σ^A	0.388	1.122	0.789	1.030		0.477	0.813	0.895	1.021
	σ^B	2.467	2.537	1.589	1.658		3.122	2.229	1.482	1.621
	σ	1.737	1.738	1.145	1.499		1.311	1.311	1.149	1.438

Table 3: Simulation experiments for $c^* = 0.5$ - queue joining policies

B Proofs

In this appendix, we present the proofs of Section 4.

Proof of Theorem 1. We prove the theorem by considering two systems with different service policies. These are two identical $GI/M/s + M$ systems (except for the service policy), denoted by 1 and 2. System 1 has a workconserving policy and system 2 has a non-workconserving policy. We allow servers in system 2 to be idle. All servers in system 1 are busy as long as jobs are present.

We describe a method to construct a sample path in both systems during a period of length T, and the systems start in the same state. The method can generate all possible sample paths in each system, according to the right distributions. We have to show that for a coupled sample path in both systems the number of abandonments is higher, or equal, in system 2. This implies a higher or equal expected fraction of abandoning calls in system 2 than in system 1 when T goes to infinity.

The construction of the sample paths is as follows. The interarrival times, remaining patience times, and remaining service times are coupled between both systems. All arrival epochs coincide

		π_A	π_1	π_2	π_3		π_A	π_1	π_2	π_3
	c	0.257	0.900	0.900	0.900		0.188	0.900	0.900	0.900
	Q^A	2.702%	7.363%	7.291%	7.533%	5.686%	16.697%	18.299%	19.256%	
	Q^B	10.529%	8.181%	8.101%	8.370%		30.241%	18.552%	20.332%	21.396%
Sys 1:	Q	6.616%	7.772%	7.696%	7.952%	Sys 2:	17.964%	17.624%	19.315%	20.326%
$\lambda_A = 5$	W^A	0.079	0.194	0.208	0.215	$\lambda_A = 6$	0.170	0.448	0.566	0.597
$\lambda_B = 5$	W^B	0.279	0.216	0.226	0.237	$\lambda_B = 6$	0.910	0.503	0.609	0.665
	W	0.175	0.205	0.217	0.226		0.484	0.475	0.587	0.630
	σ^A	0.141	0.417	0.325	0.337		0.189	0.694	0.480	0.512
	σ^B	0.529	0.450	0.374	0.374		0.941	0.741	0.563	0.563
	σ	0.393	0.434	0.350	0.356		0.729	0.718	0.523	0.539
	c	0.153	0.900	0.900	0.900		0.146	0.900	0.900	0.900
	Q^A	7.601%	27.154%	27.166%	27.154%		11.300%	42.109%	42.105%	42.111%
	Q^B	49.733%	30.171%	30.185%	30.171%		77.587%	46.788%	46.783%	46.790%
Sys 3:	Q	28.666%	28.663%	28.676%	28.663%	Sys 4:	44.441%	44.449%	44.444%	44.450%
$\lambda_A = 7$	W^A	0.229	0.721	0.902	0.894	$\lambda_A = 9$	0.345	0.967	1.576	1.550
$\lambda_B = 7$	W^B	1.767	0.822	0.960	1.009	$\lambda_B = 9$	3.804	1.124	1.665	1.785
	W	0.771	0.771	0.931	0.950		1.043	1.042	1.619	1.662
	σ^A	0.217	1.021	0.535	0.570		0.292	1.553	0.619	0.723
	σ^B	1.243	1.088	0.628	0.613		1.792	1.697	0.657	0.709
	σ	1.056	1.055	0.583	0.594		1.626	1.625	0.639	0.726
	c	0.185	0.900	0.900	0.900		0.217	0.900	0.900	0.900
	Q^A	17.022%	51.676%	51.830%	51.673%		20.791%	55.262%	55.265%	55.264%
	Q^B	92.072%	57.418%	57.589%	57.414%		95.873%	61.403%	61.406%	61.404%
Sys 5:	Q	54.547%	54.547%	54.709%	54.543%	Sys 6:	58.331%	58.332%	58.336%	58.334%
$\lambda_A = 11$	W^A	0.534	0.946	2.105	2.052	$\lambda_A = 13$	0.666	0.935	2.315	2.269
$\lambda_B = 11$	W^B	5.970	1.078	2.273	2.382	$\lambda_B = 13$	7.028	1.035	2.530	2.631
	W	1.008	1.008	2.184	2.207		0.981	0.981	2.414	2.437
	σ^A	0.388	1.633	0.730	0.859		0.436	1.468	0.789	0.906
	σ^B	2.466	1.846	0.687	0.762		2.812	1.686	0.686	0.751
	σ	1.738	1.737	0.715	0.831		1.574	1.574	0.750	0.857

Table 4: Simulation experiments for $c^* = 0.9$ - queue joining policies

		β=0	β=1/4	β=1/2	β=3/4	β=1		β=0	β=1/4	β=1/2	β=3/4	β=1
	c	0.500	0.500	0.647	0.827	0.999		0.500	0.500	0.623	0.815	1.000
	Q^A	4.202%	4.186%	5.024%	5.706%	6.310%		12.112%	11.802%	13.512%	15.795%	17.635%
	Q^B	8.404%	8.371%	7.765%	6.899%	6.313%		24.115%	23.600%	21.734%	19.385%	17.610%
Sys 1:	Q	6.303%	6.278%	6.395%	6.302%	6.311%	Sys 2:	18.113%	17.701%	17.623%	17.590%	17.623%
$\lambda_A = 5$	W^A	0.117	0.123	0.145	0.166	0.181	$\lambda_A = 6$	0.333	0.358	0.419	0.492	0.548
$\lambda_B = 5$	W^B	0.226	0.233	0.216	0.197	0.181	$\lambda_B = 6$	0.697	0.722	0.676	0.605	0.548
	W	0.170	0.177	0.180	0.181	0.181		0.502	0.527	0.541	0.547	0.548
	σ^A	0.267	0.207	0.235	0.274	0.306		0.515	0.344	0.348	0.418	0.474
	σ^B	0.460	0.428	0.380	0.339	0.306		0.826	0.706	0.613	0.533	0.474
	σ	0.378	0.338	0.317	0.309	0.306		0.701	0.573	0.509	0.481	0.474
	c	0.500	0.500	0.617	0.814	1.000		0.500	0.500	0.634	0.825	1.000
	Q^A	19.057%	19.135%	21.880%	25.714%	28.689%		29.592%	29.621%	34.533%	40.186%	44.396%
	Q^B	38.076%	38.260%	35.473%	31.629%	28.695%		59.170%	59.301%	54.496%	48.703%	44.441%
Sys 3:	Q	28.567%	28.698%	28.677%	28.672%	28.692%	Sys 4:	44.318%	44.461%	44.515%	44.444%	44.419%
$\lambda_A = 7$	W^A	0.525	0.608	0.724	0.865	0.976	$\lambda_A = 9$	0.750	1.009	1.254	1.519	1.731
$\lambda_B = 7$	W^B	1.197	1.323	1.241	1.088	0.976	$\lambda_B = 9$	1.903	2.497	2.284	1.958	1.731
	W	0.816	0.918	0.957	0.972	0.976		1.173	1.555	1.677	1.722	1.731
	σ^A	0.738	0.433	0.385	0.466	0.533		1.081	0.534	0.401	0.477	0.546
	σ^B	1.147	0.877	0.714	0.607	0.533		1.897	1.146	0.756	0.626	0.546
	σ	0.994	0.752	0.615	0.549	0.533		1.539	1.085	0.765	0.592	0.546
	c	0.500	0.500	0.659	0.839	1.000		0.500	0.500	0.681	0.850	1.000
	Q^A	36.391%	36.423%	43.711%	50.188%	54.526%		41.089%	41.107%	49.901%	56.676%	61.492%
	Q^B	72.817%	72.841%	66.339%	59.804%	54.556%		82.218%	82.181%	73.261%	66.675%	61.499%
Sys 5:	Q	54.604%	54.632%	55.025%	54.996%	54.541%	Sys 6:	61.654%	61.644%	61.581%	61.678%	61.496%
$\lambda_A = 11$	W^A	0.842	1.309	1.710	2.067	2.340	$\lambda_A = 13$	0.973	1.561	2.063	2.488	2.847
$\lambda_B = 11$	W^B	2.011	3.618	3.185	2.687	2.340	$\lambda_B = 13$	1.706	4.888	3.888	3.249	2.847
	W	1.193	2.002	2.263	2.344	2.340		1.144	2.333	2.698	2.819	2.847
	σ^A	1.131	0.541	0.412	0.481	0.546		0.891	0.464	0.417	0.481	0.546
	σ^B	2.426	1.399	0.778	0.631	0.546		2.240	1.472	0.788	0.631	0.546
	σ	1.717	1.382	0.918	0.633	0.546		1.368	1.625	1.042	0.668	0.546

Table 5: Simulation experiments for $c^* = 0.5$ - call selection policies

		β=0	β=1/4	β=1/2	β=3/4	β=1		β=0	β=1/4	β=1/2	β=3/4	β=1
	c	0.900	0.900	0.900	0.900	1.000		0.900	0.900	0.900	0.900	1.001
	Q^A	6.017%	5.988%	6.005%	6.010%	6.292%		16.713%	16.704%	16.755%	16.689%	17.627%
	Q^B	6.684%	6.656%	6.674%	6.678%	6.301%		18.571%	18.550%	18.613%	18.562%	17.607%
Sys 1:	Q	6.348%	6.327%	6.340%	6.344%	6.297%	Sys 2:	17.642%	17.627%	17.684%	17.626%	17.617%
$\lambda_A = 5$	W^A	0.162	0.168	0.172	0.173	0.181	$\lambda_A = 6$	0.468	0.501	0.514	0.519	0.548
$\lambda_B = 5$	W^B	0.179	0.186	0.189	0.190	0.181	$\lambda_B = 6$	0.523	0.558	0.572	0.578	0.548
	W	0.171	0.177	0.180	0.181	0.181		0.495	0.529	0.543	0.548	0.548
	σ^A	0.359	0.318	0.299	0.290	0.306		0.657	0.535	0.476	0.450	0.474
	σ^B	0.388	0.351	0.334	0.326	0.306		0.703	0.588	0.534	0.509	0.474
	σ	0.374	0.335	0.317	0.309	0.306		0.681	0.563	0.506	0.481	0.474
	c	0.900	0.900	0.900	0.900	1.000		0.900	0.900	0.900	0.900	1.000
	Q^A	27.202%	27.189%	27.243%	27.115%	28.767%		42.202%	42.098%	42.113%	42.102%	44.500%
	Q^B	30.224%	30.207%	30.265%	30.127%	28.705%		46.890%	46.774%	46.798%	46.780%	44.501%
Sys 3:	Q	28.713%	28.698%	28.754%	28.621%	28.736%	Sys 4:	44.546%	44.436%	44.456%	44.441%	44.500%
$\lambda_A = 7$	W^A	0.774	0.870	0.904	0.917	0.976	$\lambda_A = 9$	1.121	1.471	1.572	1.607	1.732
$\lambda_B = 7$	W^B	0.879	0.982	1.017	1.031	0.976	$\lambda_B = 9$	1.292	1.692	1.807	1.847	1.732
	W	0.826	0.925	0.959	0.973	0.976		1.203	1.577	1.684	1.722	1.732
	σ^A	0.944	0.696	0.568	0.509	0.533		1.449	0.979	0.689	0.537	0.546
	σ^B	1.008	0.761	0.638	0.581	0.533		1.578	1.083	0.779	0.619	0.546
	σ	0.977	0.731	0.606	0.548	0.533		1.515	1.036	0.743	0.590	0.546
	c	0.900	0.900	0.900	0.900	1.000		0.900	0.900	0.900	0.900	1.000
	Q^A	52.089%	51.710%	51.699%	51.701%	54.505%		58.302%	58.297%	58.298%	58.307%	62.094%
	Q^B	57.875%	57.457%	57.438%	57.443%	54.497%		64.778%	64.777%	64.771%	64.780%	62.106%
Sys 5:	Q	54.982%	54.584%	54.5%	54.572%	54.501%	Sys 6:	61.540%	61.537%	61.535%	61.543%	62.100%
$\lambda_A = 11$	W^A	1.172	1.902	2.088	2.150	2.339	$\lambda_A = 13$	1.159	2.227	2.502	2.597	2.892
$\lambda_B = 11$	W^B	1.332	2.214	2.438	2.516	2.339	$\lambda_B = 13$	1.253	2.613	2.964	3.086	2.892
	W	1.247	2.049	2.252	2.322	2.339		1.202	2.404	2.714	2.821	2.892
	σ^A	1.602	1.217	0.800	0.553	0.546		1.309	1.392	0.894	0.565	0.55
	σ^B	1.798	1.390	0.926	0.647	0.546		1.492	1.649	1.069	0.673	0.55
	σ	1.699	1.310	0.879	0.626	0.546		1.397	1.527	1.005	0.663	0.55

Table 6: Simulation experiments for $c^* = 0.9$ - call selection policies

in both systems. Because patience and service times are exponentially distributed, they are memoryless. This allows us to assign new times to the jobs after each event. We describe a procedure that assigns exactly one time to each job after each event, either a service time to a job that is being served or a patience time to a job that is waiting in the queue. A time is generated randomly from the corresponding distribution and each time is assigned twice: to a job in system 1 and to a job in system 2. The assignment repeats, and it continues until all customers have received a new time, unless the number of customers in service or in queue is higher in one of both systems. In that case, times are assigned to the remaining jobs afterwards. Thus, not each time is assigned to a job in both systems. Consequently, if a time is assigned to a job in only one of both systems, this time is not coupled to a job in the other system. As a result, the processes can differ between both systems if not all service and patience times are coupled.

For an arbitrary choice of coupled sample paths, we define $a_i(t)$ as the total number of abandonments until time t in system $i \in \{1, 2\}$, and we define $x_i(t)$ as the number of jobs in system i at time t, with $t \in [0, T]$. We show that for every t it holds that

$$a_1(t) \leq a_2(t) \text{ and } x_1(t) + a_1(t) \leq x_2(t) + a_2(t). \tag{7}$$

Assume that Relation (7) holds until time epoch t at which an event occurs. We show that the inequalities also hold after the event. The symbol t^+ denotes a time epoch immediately after the event and t^- denotes a time epoch just before the event. To prove that Relation (7) also holds

after the event, we consider all possible events:
- There is a departure in both systems of jobs with coupled service times. Then, Relation (7) holds because we have $x_1(t^+) = x_1(t^-) - 1$ and $x_2(t^+) = x_2(t^-) - 1$.
- There is an abandonment in both systems of jobs with coupled patience times $x_1(t^+) = x_1(t^-) - 1$ and $x_2(t^+) = x_2(t^-) - 1$, $a_1(t^+) = a_1(t^-) + 1$ and $a_2(t^+) = a_2(t^-) + 1$. Then, Relation (7) also holds.
- A departure occurs in system 1, such that $x_1(t^+) = x_1(t^-) - 1$ which also satisfies Relation (7).
- There is a departure in system 2, $x_2(t^+) = x_2(t^-) - 1$. This event is only possible if $x_1(t^-) < x_2(t^-)$, such that Relation (7) holds at time epoch t^+.
- There is an abandonment in system 1, $x_1(t^+) = x_1(t^-) - 1$, $a_1(t^+) = a_1(t^-) + 1$. This is only possible if $x_1(t^-) > x_2(t^-)$, which implies that $a_1(t^-) < a_2(t^-)$. We see that Relation (7) is also satisfied after the event.
- Finally, if a job in system 2 abandons, we get $x_2(t^+) = x_2(t^-) - 1$, and $a_2(t^+) = a_2(t^-) + 1$, which also satisfies Relation (7).

If we let T go to infinity, we obtain an abandonment rate that is equal or higher in system 2, just by dividing each $a_i(t)$ by the total expected number of arrivals, which is equal in both systems. □

Proof of Theorem 2. We prove the result by coupling arguments. Consider two identical $GI/GI/s + M$ models, say Model 1 and Model 2. The discipline of service in Model 1 (Model 2) is defined by the workconserving non-preemptive policy π_1 (π_2), $\pi_1 \neq \pi_2$. Our approach is based on a single sample path. In both models, we create identical successive arrival epochs, as well as identical successive service times. Service times are assigned to servers and not to arrivals. Since times before reneging are exponentially distributed, then the decision for one customer to abandon the queue is not affected by his elapsed waiting time. This enables us to create randomly, for each customer in queue, a new maximum time of patience at each selection for service (or equivalently successful departure epoch.) Assume that at time $t = 0$ both systems are empty, and let work begins. We denote by D_k the epoch of the k^{th} departure, $k = 1, 2, ..., \infty$.
Both models behave identically until a busy period starts and the following situation occurs: a server becomes idle (service completion) and more than one customer are waiting in queue. Let D_i be the epoch of that service completion (which occur simultaneously in Models 1 and 2.) For both models, let n be the number of waiting customers in queue just before D_i, $n \geq 2$. At D_i, the idle server in Model 1 selects one customer from the queue that can be different from the one selected by the idle server in Model 2. However, the number of customers in queue goes down by 1 for both models, it becomes $n-1$. Note that the number of customers who abandon the queue is up to now

identical for both models.

In Model 1, we create for each customer waiting in queue a new maximum patience time. Without altering distributions, since times before reneging are identically distributed, we create the same set of $n-1$ maximum patience times, and we assign them arbitrary to the customers waiting in Model 2. After D_i, three events are possible: one customer reneges, or a new customer enters the system, or a server becomes idle. Recall that by construction, these events occur simultaneously in both models. Assume that the first event occurs, then the number of customers who abandon the queue goes up by 1 in both models and as a consequence is still identical for them. It is still identical also if another customer abandons the queue. It is the case as long as the number of customers in queue is larger or equal to 1. Assume now that one customer enters the system. Hence, the number of customers in queue goes up by 1 in both models. Note that if another arrival occurs or that one customer abandons the queue, then, the number of customers in queue will increase by 1 or decrease by one, respectively. Thus, the number of customers who abandon the queue vary identically from one model to another. Assume now that one server becomes idle. If the number of customers in queue is less or equal to 1, it is obvious to see owing to the workconserving property, that policies π_1 and π_2 will select the unique available customer, if any. Otherwise, the busy period ends in both models, hence, both policies will select identically new arrivals for service until the beginning of the next busy period. However, if the number of customers in queue is greater or equal to 2, the selected customer for service may be different in both models. As above, we create for the remaining waiting customers in both models, the same set of maximum patience times. Recall that until now, the number of customers who abandon the queue is still identical for both models. Continuing with the same arguments, we state that during the steady state, the number of customers who abandon the queue in Model 1 coincides with the one in Model 2. Since by construction of the sample path the number of arrivals are also equal for both models, hence, we conclude that the fraction of abandoning customers is unchanged, $Q_{\pi 1} = Q_{\pi 2}$. This completes the proof of the theorem. □

Proof of Theorem 3. We prove the first statement by coupling arguments. Using the same notations as in the proof of Theorem (2), we couple Models 1 and 2 using a single sample path. We showed that the number of waiting customers in queue is identically distributed for both models. Then, the mean number of customers in queue, say L_q^{tot}, does not depend on the scheduling policy. Let λ be the average rate of arrivals. So, we state from the Little's Law that $\lambda\, W_q^{tot} = L_q^{tot}$. Then, we easily deduce that W_q^{tot} is independent of the scheduling policy, which completes the proof of the first statement.

We use a simple counterexample to prove the second statement. Let us couple Models 1 and

2. Model 1 is working under the FCFS discipline, and Model 2 is working under a workconserving non-preemptive policy different of the FCFS discipline, say π. Policy π works identically as the FCFS discipline except when 2 customers are waiting in queue and a service completion occurs. At that moment, π selects the younger customer (the second one), whereas the FCFS discipline chooses of course the one in the head of the queue (the older customer.) Using a single sample path, both models behave identically until the first time when two customers are waiting in queue and a service completion occurs. Let us stop our clock temporarily. Let D be the epoch of that event. We denote by A_1 and A_2 the first and the second waiting customers in the queue of Model 1, respectively. The same customers are also waiting in the queue of Model 2. Let w_1 and w_2 be the ages in queue of customers A_1 and A_2, respectively. Since A_1 entered in system before A_2, then $w_1 > w_2$. Also, let w_q^{FCFS} and w_q^{π} be the cumulative waiting times in queue for served customers in Model 1 and Model 2, respectively. Up to now, we have $w_q^{FCFS} = w_q^{\pi}$. Let our clock resumes ticking. The idle server in Model 1 selects the customer waiting in the head of the queue, namely A_1. However, the same server in Model 2 selects A_2. Updating the cumulative waiting times for the served customers leads to $w_q^{FCFS} = w_q^{\pi} + (w_1 - w_2)$, hence, $w_q^{FCFS} > w_q^{\pi}$. From the memoryless property of the distribution of times before reneging, we generate a new time before reneging and affect it twice: to A_2 (the unique customer waiting in Model 1) and to A_1 (the unique customer waiting in Model 2.) Only three non-zero probability events are possible: either a service completion occurs, or a customer abandons the queue, or a new customer joins the queue. Note that these events occur simultaneously for both models. If a service completion occurs first, then the idle server will select the unique available customer in queue, which allows w_q^{FCFS} to coincide again with w_q^{π}. If A_2 abandons in Model 1 and A_1 abandons in Model 2, then w_q^{FCFS} is still greater than w_q^{π}. Both quantities will never coincide thereafter. Assume now that a new arrival occurs. From the structure of both policies, we see that future events lead to: either A_1 in Model 2 and A_2 in Model 1 enter service, or A_1 and A_2 abandon before being served, or A_1 abandons and A_2 gets service. The event A_1 gets service and A_2 abandons is not possible. Thereafter, if A_1 abandons, then w_q^{FCFS} and w_q^{π} will never coincides again. Otherwise, we see that these quantities coincide again (in the best case) further to a given combination of events. A central statement is that there is no possible combination, in any point of the sample path, which may make w_q^{FCFS} strictly lower than w_q^{π}. Following the same explanation until the stationary regime and knowing from Theorem (2) that the number of served customers is the same for both models, we finally state that the stationary average waiting time W_q^{FCFS} will be strictly greater than the stationary average waiting time W_q^{π}. This completes the proof of the second statement.

To prove the third statement, we again couple Model 1 and Model 2. The scheduling policy for Model 1 is the FCFS policy. The one for Model 2 is different of FCFS and is denoted by π'. Then, at least for some situations, the oldest customer (waiting in the head of the queue of Model 2) looses the higher priority for service. Taking a single sample path, let us now compare the cumulative waiting times for served customers in Model 1, say w_q^{FCFS} with that in Model 2, say $w_q^{\pi'}$. Initially and as long as both policies (FCFS and π') select identically the waiting customers, w_q^{FCFS} equals $w_q^{\pi'}$. The first time when π' selects a customer in a different manner as that in the FCFS discipline, w_q^{FCFS} is no longer equal to $w_q^{\pi'}$. Since the FCFS discipline selects the oldest customer, hence, w_q^{FCFS} becomes strictly larger than $w_q^{\pi'}$. Each time FCFS and π' select customers for service differently, w_q^{FCFS} becomes more and more larger $w_q^{\pi'}$. In a distant future, knowing that the number of served customers is unchanged under both policies, we state that the largest expected waiting time of served customers, W_q, is achieved under the FCFS policy. Applying above arguments by considering the LCLS discipline and a workconserving non-preemptive policy different of LCLS, we state that the LCLS policy is optimal subject to minimizing the average waiting time of served customers, which finishes the proof of the third statement.

The fourth statement is a direct consequence of the third one. It suffices to recall that the overall cumulative waiting time, defined as $w_q^{tot} = w_q + w_q^{ab}$, is unchanged under both policies. Hence, the policy that maximizes W_q will minimize W_q^{ab}, and vice versa. This completes the proof of the fourth statement and the theorem. □

Proof of Corollary 1. This is an immediate consequence of Theorem (2) and the first statement of Theorem (3). It suffices to divide arrivals into two streams of customers to get the result. □

Proof of Corollary 2. Consider a workconserving non-preemptive scheduling policy, say π, different from π_A. Then, in some situations under π, we first select a customer B from the queue whereas there is at least one waiting customer A. A sample path comparison of the system working under π with an identical one working under π_A may easily show that the distribution of the number of customers A waiting in the queue of the first system is greater than that for the second system. Subsequently, the number of customers A who abandon the queue is strictly greater in the first system, and equivalently, $Q_\pi^A > Q_{\pi_A}^A$. From that, one can state that

$$\min_{\pi\in\Pi}\{Q_\pi^A\} = Q_{\pi_A}^A, \tag{8}$$

where Π denotes the class of workconserving non-preemptive policies. As shown in Theorem (2), the overall fraction of customers who abandon, in the stationary regime, held constant under any workconserving non-preemptive policy. Thus, Equation (8) leads to

$$\max_{\pi\in\Pi}\{Q_\pi^B\} = Q_{\pi_A}^B. \tag{9}$$

From Equations (8) and (9), we thereafter deduce that $c_\pi \geq c_{\pi_A}$, which completes the proof of the first part of the corollary. For the second part, it suffices to follow the same arguments as above to state that $\max_{\pi\in\Pi}\{Q_\pi^A\} = Q_{\pi_B}^A$, and $\min_{\pi\in\Pi}\{Q_\pi^B\} = Q_{\pi_B}^B$. Thus, $c_\pi \leq c_{\pi_B}$. $\square$

Proof of Theorem 4. Assume there is a separate queue for each customer type. Using the memoryless property of times before reneging, we state that the fractions of abandoning customers are proportionally (with the reneging rate as a factor) to the average queue lengths of both types. Having a common reneging rate for both types, the result follows from the Little's law. $\square$

Proof of Theorem 5. We start by proving the first statement. Let us take our basic model working under the scheduling policy π_1. First, it is easy to see that there are two bounds for achievable ratios, namely c_{π_A} and c_{π_B}. The reason comes from the fact that we are considering workconserving non-preemptive policies. The lower (upper) bound, c_{π_A} (c_{π_B}), is achieved when we give strict priority to customers A (B.) In addition, assigning dynamically customers to the high or the low priority queue, as under policy π_1, will affect the quantities $Q_{\pi_1}^A$ and $Q_{\pi_1}^B$ in the stationary regime, and equivalently the ratio $c_{\pi_1} = Q_{\pi_1}^A / Q_{\pi_1}^B$. Consider a given c^* ranging between c_{π_A} and c_{π_B}. During the transient regime, if it happens that the achieved ratio is strictly lower than c^*, then giving the priority to customers B (which is possible under π_1) allows necessarily to go beyond c^* after a given duration of time. Continuing in doing this, the ratio converges to c_{π_B}. Otherwise during the stationary regime, if it happens that the achieved ratio is strictly greater than c^*, then giving the priority to customers A (which is possible under π_1) allows necessarily to go below c^* after a given duration of time. Continuing in doing this, the ratio converges to c_{π_A}. As the number of arrivals grows, these manipulations would make the difference between the achieved ratios and the objective less and less low. This would allow the stationary ratio to coincide with c^*.

Let us now focus on proving the second statement. Let c_{FCFS} be the achieved ratio under the FCFS policy. With the same explanation as above, we state that any ratio ranging from c_{π_A} to c_{FCFS} can be reached by π_2. On the one hand, The lower bound for Q^A is achieved when we give strict priority to A customers. In addition at the same time, the upper bound for Q^B is reached. Thus, the minimum possible achievable target ratio corresponds to policy c_{π_A}. On the other hand, the lower bound for Q^B is achieved by assigning all type B arrivals to queue 1, which also allows to achieve the upper bound for Q^A. This corresponds to the FCFS policy for all arrival types. Hence, the achieved ratio could not be worse than that under the c_{FCFS} policy. One may easily see that $c_{FCFS} = 1$. In fact, our model working under the FCFS manner, for each customer type and for both customer types, is simply equivalent to a single class model working under the FCFS manner.

Finally, we note that the proof of the third statement is similar to that of the second statement. This completes the proof of the theorem. $\square$