Front-office multitasking between service encounters and back-office tasks

Benjamin Legros[1] • Oualid Jouini[2] • O. Zeynep Akşin[3] • Ger Koole[4]

[1]*EM Normandie, Laboratoire Métis, 64 Rue du Ranelagh, 75016 Paris, France*

[2]*Université Paris-Saclay, CentraleSupélec, Laboratoire Genie Industriel, 3 rue Joliot-Curie 91190 Gif-sur-Yvette, France*

[3]*College of Administrative Sciences and Economics, Koç University, Rumeli Feneri Yolu, 34450 Sariyer-Istanbul, Turkey*

[4]*VU University Amsterdam, Department of Mathematics, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands*

benjamin.legros@centraliens.net • oualid.jouini@centralesupelec.fr • zaksin@ku.edu.tr • ger.koole@vu.nl

Accepted in ***European Journal of Operational Research, 2020.***

Abstract

We model the work of a front-line service worker as a queueing system. The server interacts with customers in a multi-stage process with random durations. Some stages require an interaction between server and customer, while other stages are performed by the customer as a self-service task or with the help of another resource. Random arrivals by customers at the beginning and during an encounter create random lengths of idle time in the work of the server (breaks and interludes respectively). The server considers treatment of an infinite amount of back-office tasks, or tasks that do not require interaction with the customer, during these idle times. We consider an optimal control problem for the server's work. The main question we explore is whether to use the interludes in service encounters for treating back-office, when the latter incur switching times. Under certain operating environments, working on back-office during interludes is shown to be valuable. Switching times play a critical role in the optimal control of the server's work, at times leading the server to prefer remaining idle during breaks and interludes, instead of working on back-office, and at others to continue back-office in the presence of waiting customers. The optimal policy for use of the interludes is one with multiple thresholds depending on both the customers queueing for service, and the ones who are in-service. We illustrate that in settings with multiple interludes in an encounter, if at all, the back-office work should be concentrated on fewer, longer and later interludes.

Keywords: Multitasking; front-office service work; case-manager system; queueing system control.

1 Introduction

In front-office service processes the customer is present during production and delivery and may take an active role as a co-producer. Such services require multiple phases of service employee-customer interactions and are labeled as service encounters. The focus of this paper is the work of the front-office service employee (the focal service worker) in such a service encounter. In some settings, these service encounters are called cases, and the front-office service employee is known as a case manager. Phases of the service encounter that are not performed by the focal worker are not explicitly modeled and are characterized as a delay with random duration in the focal employee's work. These delays, that occur during an ongoing service encounter while the customer is away for an offline task, are labelled as interludes for the front-office server. Each new customer arrival generates a new service encounter. The random duration during which there are no customers present in the system is labelled as a break in the focal employee's work. During times when the focal worker is not interacting with customers, i.e. during interludes and breaks, the worker can choose to remain idle or work on other tasks that can be performed without the presence of a customer. The latter tasks are known as back-office tasks and are expected to have lower urgency compared to the front-office. Alternating between front-office and back-office tasks requires a switching time for the focal server. In such a setting, the goal of this paper is to formulate and analyze the optimal control problem for the focal server' work, exploring the use of interludes and breaks to perform back-office tasks when these incur switching times.

The simplest such system is one where the front-office service worker is an investigator. In an after-sales technical service center, a customer with a repair need may report a problem to an investigator, perform several tests on the item as suggested by this server in self-service mode, report back on the results of these to the investigator to obtain a suggestion regarding next steps. At an insurance firm or bank branch, the customer who wants to buy a product makes a request with the focal server, then completes requested documentation, subsequently coming back to the investigator to conclude the application process. Both of these examples can be characterized by a three-stage process where the first and last stages are with the focal server, and the second stage consists of an external delay (an interlude) that involves either a self-service step, or another task performed by other resources. Dobson et al. (2013) provide an example from an emergency department (ED) for such a three-stage process. An investigator who is an ED physician examines the patient, then tests necessary for diagnosis are performed by a back-office, and the patient returns to the physician in stage three to conclude the investigation. In other settings, processes may be more complex involving more than three stages. For example, an audit process consists of several interactions between a client and the focal server, while other servers will provide required validations between steps. Similarly, a general practitioner (the focal server) may have several interactions with a patient, interrupted by tests or imaging performed between visits. In all of these examples, the focal server has a choice between remaining idle, starting a new service encounter, or performing some back-office tasks during interludes and breaks between customers. For instance, in the after-sales service center the server may perform some routine paper work while the customer is away, and in the insurance firm the server may work on investigations during such interludes. The general practitioner may be writing reports concerning other patients.

The existence of switching times when multitasking between two types of distinct tasks is well documented in the behavioral literature. First, multitasking or working on several tasks at a time actually means working on different tasks by alternation instead of working on them strictly at the same time. Simultaneity is not efficient for humans, as it can create interferences between jobs (Gladstones et al., 1989), and can lead to mistakes (Rosen, 2008; Lohr, 2007). Yet alternation induces inefficiency and time loss related to switching between tasks Rosen (2008). Minimizing such losses requires the limitation to alternation between at most two distinct tasks (Charron and Koechlin, 2010; Dux et al., 2009). In this paper, our main motivation is to study the role of switching times when alternating between front-office and back-office tasks in the work of a front-office service worker. Front-office tasks are considered to be of higher urgency relative to back-office tasks due to the presence of the customer. The server's multitasking control problem can be formulated as one that tries to maximize the proportion of time spent on back-office tasks while respecting some service level on the waiting times of front-office tasks. More precisely, the question we wish to analyze is how to use the interludes and breaks in service: by remaining idle and thus ready for the customer, by initiating the service of new customers, or by making use of these times to work on back-office tasks at the expense of incurring switching times to start and then to return to the customer task?

One way of avoiding switching times, while keeping the server busy, is to have the server start on a new task of similar nature. In the service front-office context, this corresponds to starting the service of a new customer during an interlude. As the front-office server initiates new service encounters during interlude times, several customers will be in the system at different stages of their service encounter. The number of customers in the system may be unrestricted in settings where the customer performs self-service tasks during interlude times. In other systems, managers may impose a limit on the number of customers that are allowed to be simultaneously in service to avoid in-service waits or other disruptions. This limit is called a caseload. Examples of such systems can be a contact center where servers manage multiple online chats (Legros and Jouini, 2019), or an emergency department where physicians treat multiple patients simultaneously (Campello et al., 2017). While our main research question is how to make use of interludes in the presence of switching times between front and back-office tasks, we also explore the effect of an exogenously determined caseload on this main problem.

More generally, white-collar service work consists of multi-stage customer-server interactions, where back-office tasks can be combined with customer service. Workflow systems enable the management of such multitasking and render automated control of such work viable. While the idea of optimally controlling the way such a server blends back-office tasks between customer encounters has been studied in the context of call center blending problems, the combined problem of blending back-office work between as well as within service encounters has not been analyzed before. The latter feature requires not just considering pre-process waiting times but also in-process waiting times for customers as the server chooses between staying idle or working on back-office tasks. Switching times have not been considered in any front office-back office blending models before.

Our analysis shows that using the interludes may make sense in some operating environments, and further explores how this choice interacts with features such as the workload, task durations, switching time durations, and caseload of the server. For three-stage systems, we characterize the optimal policy via a Markov decision process approach. Switching times play a critical role in the optimal control of the server's work, at times leading the server to prefer remaining idle during breaks and interludes, instead of working on back-office tasks, and at others to continue back-office tasks in the presence of waiting customers. For more general service encounters, we focus on a simpler non-optimal policy where the performance measures can be computed explicitly. Sensitivity analyses in the simpler policy demonstrate that blending should first be undertaken during breaks and only then attempted during interludes. Our analysis also reveals that the interlude duration should not be artificially extended by continuing the work on back-office tasks while a customer wants to finish a last stage of service. In addition, the direction of the swicthes from front-office to back-office or back-office to front-office task has an influence. In the presence of multiple interludes in an encounter, if at all, the back-office tasks should be concentrated on fewer, longer and later interludes.

2 Literature review

Models of multitasking in queueing systems can be classified into three categories based on the way the server treats the tasks: simultaneous treatment, blending treatment (treat one task at a time), and imbricated treatment (treat one task within the treatment of another one). Simultaneous treatment is related to computer multitasking in switching networks or processor sharing queues (Stolyar, 2004; Gromoll et al., 2008). Human simultaneous multitasking is rare and is typically done in the form of alternating between tasks.

The main idea in queue blending models is to determine efficient scheduling policies for the treatment of urgent and non-urgent jobs. As in this paper, the optimization problem in these studies consists of maximizing the time spent on non-urgent jobs while achieving a service level constraint on urgent ones. One important difference in all of the queue blending models as compared to our model assumptions is that the urgent tasks consist of single stages and switching times between tasks are not considered. Blending models have been widely studied in the context of call centers with urgent inbound and non-urgent outbound calls. Models by Brandt and Brandt (1999), Deslauriers et al. (2007) evaluate performance in such systems. Bhulai and Koole (2003); Gans and Zhou (2003); Legros et al. (2015) consider queue blending models where the inbound jobs have a non-preemptive priority over the outbound ones. They show that the optimal policy is a reservation threshold policy on the number of busy servers. Further references on queue blending operations include Keblis and Chen (2006); Pichitlamken et al. (2003); Pang and Perry (2014). Server's reservation is also shown to be effective in our article when switching times between tasks are considered. Armony and Maglaras (2004); Legros et al. (2016) analyze optimal server scheduling policies in models with a call-back option, which allows to transform an inbound call into an outbound one. As in this paper, they show that optimal policies are a function of the number of waiting inbound calls and have a threshold form.

Polling systems consider servers that alternate between stations. The objective in these studies is to determine the optimal priority rules between stations to minimize the expected time spent in the system or a more complex holding cost function. These queueing models are also called reentrant lines and are very complex to analyze (Hasenbein, 1997; Koole and Righter, 1998; Dai et al., 2004; Yom-Tov and Mandelbaum, 2014). None of these studies investigate the possibility to initiate non-urgent tasks and most of them do not consider switches when moving from one station to another. In Srinivasan and Gupta (1996), a system where different customer classes are served at different parallel stations, with a roving server who incurs a switching time to switch from one station to another is considered. Their analysis shows that it may be better from a work in-process minimization perspective for the server to wait at a station even when there are no waiting jobs at this station, rather than roving. In our setting, the server moves between tandem service stations and a parallel back-office station, and customers may be waiting both to get into service or to continue service at an in-process station, while back-office tasks are infinite and always available. We show that the server may choose to continue treating back-office tasks during interludes when customers are waiting, or to remain idle during these times despite the desire to treat back-office tasks.

Without switches, in a multi-phase service process, Johri and Kateiiakis (1988) show that it is optimal to keep on serving a customer until service completion instead of initiating the service of new customers, when the objective is to minimize the expected time spent in the system. Our optimization problem differs from theirs as we do not have the objective to minimize the time spent in the system by high priority jobs. Our conclusion is therefore also different; a strict priority for the oldest customer in the system may not be optimal. Iravani et al. (1997) consider a model with two phases of service, two queues, and no switch. Using a Markov decision process approach, they derive the optimal policy and show that it can be approximated by a triple-threshold policy. The thresholds determine the states at which, it is preferred to switch from the first to the second queue, it is preferred to wait in the first queue, or it is preferred to switch back from the second to the first queue. In a similar setting, with switches, we also show that the optimal policy has a threshold nature and that the switches from urgent to non-urgent jobs are done in different states than the switches back from non-urgent to urgent jobs. In the multi-server case, Andradóttir et al. (2001) determine the contexts where it is optimal to spread the severs among the different stations. With switches, Duenyas et al. (1998) provide a partial characterization of the optimal policy. In particular, as in our article, they show that the server should stay at a given station until the number of customers at another station exceeds a given threshold.

Case-manager systems are an example of imbricated service. Campello et al. (2017) study such systems where a case-manager deals with the case of a customer that consists of a random number of tasks, interspersed with so called external delays (similar to interludes herein) during which the customer is away completing tasks elsewhere. The maximum number of customers that a case-manager takes on at a time is called the caseload. The paper investigates the tradeoff between the wait by customers upon arrival which is reduced with an increase in the caseload, and the in-process wait of customers who come back from external delay and find their case-manager working on other customers' cases which is increasing in the caseload. In KC (2013), multitasking in an emergency department is measured by the number of patients simultaneously under care, thus resembling the caseload in Campello et al. (2017). Increased caseload has a negative effect on quality in KC (2013), while in Campello et al. (2017) increasing the caseload has an effect on wait times. Dobson et al. (2013) consider a model for an investigator, where again the investigator can take on new customers while an existing one is away. There is no switching time for the investigator between customers, however customers who remain in the system generate so called interruptions, which affect the efficiency of the investigator. Chat service systems in contact centers are also a form of multitasking with imbricated service, where it is assumed that a server can simultaneously treat different chats (Shae et al., 2007; Cui and Tezcan, 2016; Tezcan and Zhang, 2014), alternating between chats whenever the focal one enters into an interlude. None of these papers consider the possibility of a different type of task, like the back-office tasks we consider.

Another instance of multitasking with imbricated service is found in Gurvich and Van Mieghem (2017) between collaborative and individual work for professional service workers. Workers' individual work is interrupted by collaborative work, which requires the simultaneous presence of multiple workers. The individual tasks of Gurvich and Van Mieghem (2017) are like the back-office tasks in our setting. Collaborative tasks resemble the front-office tasks. While those that are labeled as cases of reaching out by the server are different from front-office tasks in that their demand is driven by the server, those that are labeled as responding to collaborative requests are similar to front-office tasks where the server needs to respond to customer requests arriving randomly. For this latter type, their arrivals are exogenous and require the presence of more than one party to be present. In our model the two parties are the server and the customer, while in theirs it is two servers. Similar to our paper, Gurvich and Van Mieghem (2017) emphasize the importance of efficiency losses created by switching times between different types of tasks. Collaboration is also modeled in Gurvich and Van Mieghem (2014), who study networks where some activities require the simultaneous processing by multiple types of multitasking human resources. Dobson et al. (2012) consider a system with three tandem stations and two servers. Each server serves at one station, however a third station in the middle represents a collaborative task where both servers have to be simultaneously present. This model combines the tandem nature of our service encounter, with a collaborative task between servers. While we assume back-office tasks are infinite, here patient requests are infinite so that arrivals are not random. There is no switching time between tasks for the servers, however the need to have both servers simultaneously present combined with random service times, induces the optimality of batching for the collaborative station. This resembles the queue state-dependent policies we find. In our setting, random customer returns from interludes and the need for simultaneous presence of the customer and the server, combined with switching times creates a similar effect.

3 Problem description and modeling

We consider a single-server multitasking problem with high urgency (HP) and low urgency (LP) jobs. HPs require the presence of the customer (front-office), arrive over time and should be treated quickly, whereas LPs do not require the presence of a waiting customer (back-office) and hence do not necessitate immediate treatment. The HP treatment consists of a succession of working phases interrupted by interlude phases for the server. During interlude phases, the server is not needed and can possibly work on another task. During working phases there is a direct interaction between the HP in service and the server. No other task can be assigned to the server at these moments.

Model assumptions. We model the server multitasking problem as a single-server queueing system. The server we consider is referred to as the *focal server*, and any other server is referred to as a *non-focal server*. The arrival process of HP is Poisson with arrival rate λ. If the server cannot serve an arriving HP, then the HP waits (pre-process) in a first-come-first-served (FCFS) infinite capacity queue called Queue 1. In its full generality, we model the service time of an HP by a succession of $N \geq 1$ independent working and interlude phases, where working phases are generally distributed and interlude phases are exponentially distributed. The first and the last phases are working phases (thus N is an odd number). With $N = 1$, the service is in one single phase without interlude. The investigator system, with a single interlude, corresponds to the case $N = 3$. After its last phase of service, the HP leaves the system. All HPs are assumed to require the same number of service phases. This restricts the model to settings where the service encounters can be considered as relatively standardized in terms of service phases needed per customer. Processes with standard operating procedures would fall in this group. While going through an interlude phase, the HP is routed to a non-focal resource which is modeled as a random length of the interlude duration. Thus, our model does not explicitly characterize the non-focal resource. The latter is modeled as though there is an infinite capacity with no waiting or blocking due to the unavailability of this resource. After a random interlude time spent with this non-focal resource, the HP directly continues its service if the focal server is available. Otherwise, the HP waits (in-process) in another FCFS queue, called Queue 2, until the server becomes available. We consider a dynamic state-dependent priority between Queue 1 and Queue 2 which is optimized within the optimization Problem (1) defined below.

In addition, we assume to have an infinite number of LPs that are waiting to be treated. These tasks are independent from the HPs and their service time distribution does not need to be specified. At any point in time, if the server is working on LP, she can decide to interrupt her work in order to serve an HP waiting in Queue 1 or in Queue 2. Since HP and LP represent different types of tasks, the server incurs a switching time in switching from LP to HP. The switching time can be considered as a mental adjustment time for the server, the time for the server to switch systems, or the time it takes to wrap-up the ongoing LP work. The switching time duration is random and assumed to be exponentially distributed with rate μ_{S_1}.

When the server is working on HP during a working phase of the service, the service cannot be preempted by any other jobs. At the end of a working phase, the server can decide either to remain idle, to serve another HP if there is one either in Queue 1 or in Queue 2, or to serve an LP. If the server decides to work on LP, then there also exists a switching time from HP to LP. The switching time duration is random and assumed to be exponentially distributed with rate μ_{S_2}. For simplicity, we use the notation μ_S when $\mu_{S_1} = \mu_{S_2} = \mu_S$. Note that during a switching time from HP to LP, an HP may arrive at Queue 1 or at Queue 2. In this case, we assume that the HP is directly served by the server without waiting for the end of the switch. This is a simplifying assumption made for tractability, and may not hold in some applications.

For the focal server, we distinguish the periods of time where the system is empty of HP; the *server's break*, and the ones where the server is not working on HP but at least one HP is with the non-focal server; *the interlude*. In addition, some systems limit the number of HPs which have initiated their service. The maximal number of HPs either in service with the focal server, with the non-focal server, or in Queue 2 is called the *caseload*. For instance, with caseload= 1, the server is not allowed to initiate the service of any HP until the HP currently in service has completely finished its service. In other words, with caseload= 1, a strict reservation-priority is given to Queue 2. We assume that the system parameters are such that the system is stable. The stability condition can be obtained when the focal server does not initiate LPs. For caseload= 1, the stability condition corresponds to the one of an M/G/1 queue, where the expected service time includes the working phases and the interludes (see

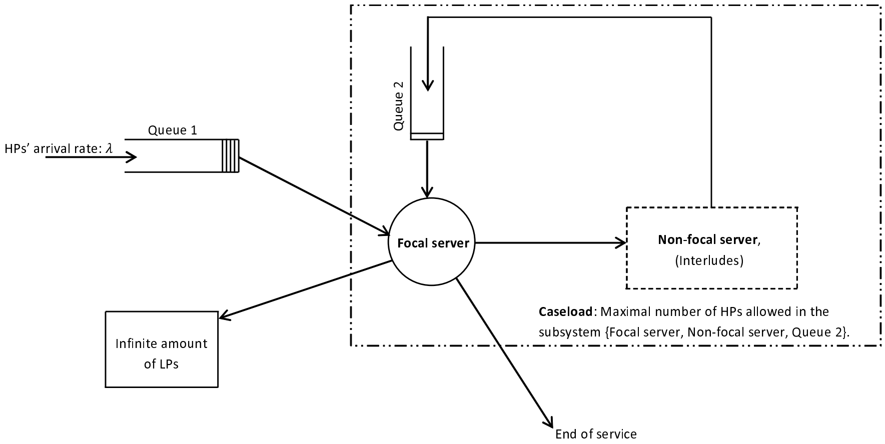

Queue 1
HPs' arrival rate: λ
Queue 2
Focal server
Non-focal server,
(Interludes)
Caseload: Maximal number of HPs allowed in the subsystem {Focal server, Non-focal server, Queue 2}.
Infinite amount of LPs
End of service

Figure 1: Model description

Section 5). The stability condition for caseload$= 1$ is a sufficient condition for having stability for caseload> 1. With infinite caseload, the system is work-conserving and some results of the queueing literature may apply. For instance if working phases are exponentially distributed with rates μ_1, μ_2, ..., and, μ_n, a necessary condition for stability is $\frac{\lambda}{\mu_1} + \frac{\lambda}{\mu_2} + \cdots + \frac{\lambda}{\mu_n} < 1$ (Dai and Weiss, 1996). Other references for the stability of reentrant lines include Hasenbein (1997); Koole and Righter (1998); Dai et al. (2004); Yom-Tov and Mandelbaum (2014). For finite caseload, the stability condition is complex to determine as the system is not work-conserving. To the best of our knowledge, the literature on reentrant lines does not provide stability conditions for this case. With exponential working phases, we can determine the stability condition by the matrix geometric technique developed in Neuts (1981). The idea is to compute the *traffic* coefficient for the associated Quasi-birth-and-death process, and, to determine the condition under which the infinitesimal generator matrix is positive recurrent. This condition will ensure the existence of the stationary regime. This method is however only limited to small caseload values. For larger values of the caseload, the system dimensionality may be too high to implement this approach. The model is depicted in Figure 1.

The optimization problem. To evaluate the performance measures related to LPs and HPs, we consider the random variables T and W which represent the proportion of time spent by the server on LP and the total waiting time in the two queues by an HP customer, respectively. We are interested in the long-run expected values of these random variables; $E(T)$ and $E(W)$. More specifically, consider an interval of time $[0, t]$ and denote by $L(t)$ the total time spent on LP by the server excluding the switching times. We define $E(T)$ as $E(T) = \lim\limits_{t \longrightarrow \infty} \frac{L(t)}{t}$. Let us denote by $A(t)$ the number of arrivals during $[0, t]$ and by W_k the total waiting time of the k^{th} customer in the system (i.e., the wait in Queue 1 and in Queue 2). We then define $E(W)$ as $E(W) = \lim\limits_{t \longrightarrow \infty} \sum\limits_{k=1}^{A(t)} \frac{W_k}{A(t)}$. The quantity $E(W)$ can be decomposed into the expected wait in Queue 1 (denoted by $E(W_1)$) plus the expected wait in Queue 2 (denoted by $E(W_2)$), $E(W) = E(W_1) + E(W_2)$.

At any point in time, the server has to decide whether to remain idle, to serve an HP, or to serve an LP in order to have an efficient allocation of the time to front and back-office tasks. As customers are actively waiting to be served, a waiting time constraint has to be met for HPs. LPs are non-urgent but are nevertheless valuable for the system, so the time spent to treat these tasks should be maximized. Having a service level constraint on urgent tasks and maximizing the proportion of time spent on non-urgent ones is consistent with the optimization problems encountered in the queue blending literature (Bhulai and Koole, 2003; Gans and Zhou, 2003; Legros et al., 2015).

We thus formulate the optimization problem as:

$$\begin{cases} \text{Maximize } E(T) \\ \text{subject to } E(W) \leq \overline{w}, \end{cases} \tag{1}$$

where the expected waiting time service level threshold is $\overline{w}$. A solution to Problem (1) exists if we have $E(W) \leq \overline{w}$ when no LPs are treated. For simplicity, we will refer to the expected proportion of time spent on LP as the expected time spent on LP instead. We choose to restrict the class of admissible policies to the class of stationary policies which are non-idling in HPs, i.e. the server is not allowed to idle if an HP is available for service. This assumption is in line with typical service management practice.

Because of the exponential assumptions for inter-arrival times at Queue 1 and for interlude times with the non-focal server (which corresponds to arrivals at Queue 2), if a decision is optimal upon a service phase completion or an arrival instant, then the same decision is also optimal later on and as long as the system state does not change. This result is due to the memoryless property of the exponential distribution. Although the server working phases can be generally distributed, job preemption is not allowed when the server is working on HP. Therefore, the *decision instants* for the server are only upon the service phase completion times and the arrival instants of HPs at Queue 1 or at Queue 2.

Consider a decision instant where the server has just completed the service phase of an HP and at least one HP is waiting in Queue 1 or in Queue 2. The focal server has to choose between scheduling an HP or an LP (or idling, but this is evidently suboptimal). Giving priority to LPs and delaying HPs obviously leads to higher waiting times. Delaying the processing of an LP job does not change the performance for this class, as we are interested in the long-term time spent by the server on these jobs. This intuitive argument implies that, when a server becomes idle and an HP is waiting, it is optimal to serve this HP. Based on this intuitive argument, we assume that the server gives priority to HPs after a service phase completion of an HP and another HP is waiting. This priority rule is consistent with similar optimization problems encountered in the queue blending literature (Bhulai and Koole, 2003; Legros et al., 2015). Moreover, during a switch from HP to LP if an HP arrives at Queue 1 or at Queue 2, then this HP is directly served.

Hence, after the service phase completion of an HP, the decision to treat LP can only be taken if there is no HP waiting in the system or if Queue 2 is empty and the caseload is such that the server is not allowed to start the service of an HP from Queue 1. In such cases, the server can decide either to remain idle or to treat LP. The value of remaining idle is to avoid any waste of time due to switching times when an HP should be served. The value of serving LPs is to increase the time spent on these jobs. This decision is complex and depends on the system state; the number of HPs in Queue 1, the number of HPs with the non-focal server, and the service progress of each HP while going through the interlude.

Consider now a situation where the server is working on LP. The next possible decision instants are those of HP arrivals at Queue 1 or at Queue 2. Due to the switching time from LP to HP, the server may not automatically interrupt the service of LP upon an HP arrival at one of the queues. It may instead be beneficial to wait until the congestion in one of the queues is sufficiently important to consider the switch. Finally, if the server is in switch from LP to HP, an arrival at Queue 1 or at Queue 2 will not modify the initial decision to treat an HP.

In Figure 2, we summarize the decision actions discussed above. The server's optimal actions are state-dependent when working on LP at an arrival instant, or when an HP working phase has finished and no HP is waiting for service. The optimal policy is a function of the number of HPs in Queue 1, in Queue 2, with the non-focal server, and the remaining service time distribution of each HP. This makes it difficult to obtain. We therefore propose the following analyses to solve the optimization problem.

1. **Optimal Policy (Section 4).** We first focus on investigator systems with $N = 1$ and $N = 3$, assuming

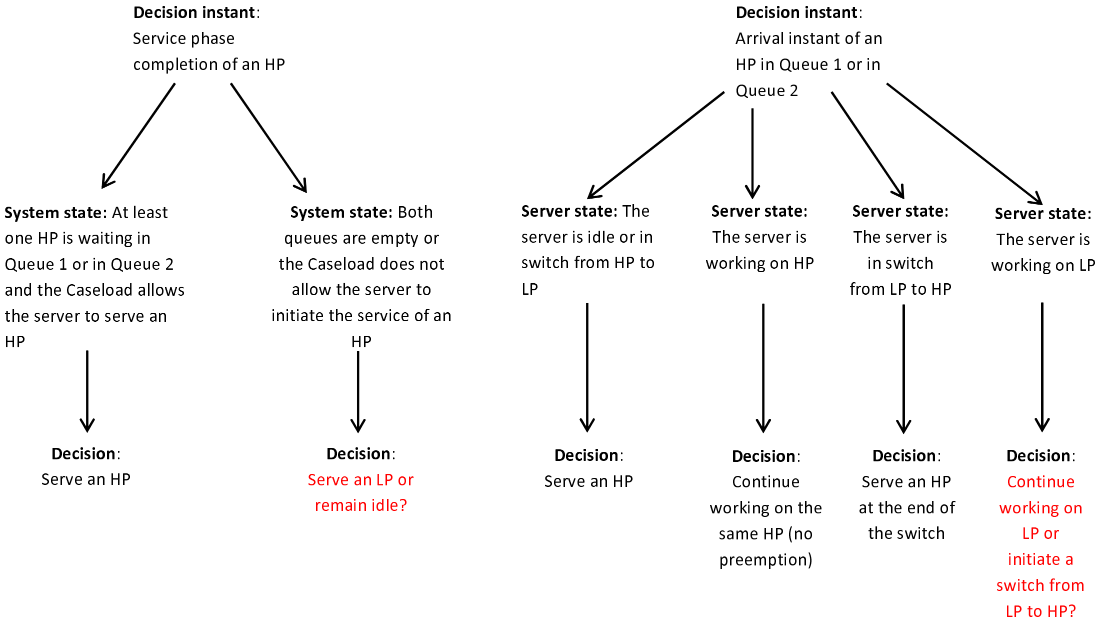

Decision instant: Service phase completion of an HP
System state: At least one HP is waiting in Queue 1 or in Queue 2 and the Caseload allows the server to serve an HP
Decision: Serve an HP
System state: Both queues are empty or the Caseload does not allow the server to initiate the service of an HP
Decision: Serve an LP or remain idle?
Decision instant: Arrival instant of an HP in Queue 1 or in Queue 2
Server state: The server is idle or in switch from HP to LP
Decision: Serve an HP
Server state: The server is working on HP
Decision: Continue working on the same HP (no preemption)
Server state: The server is in switch from LP to HP
Decision: Serve an HP at the end of the switch
Server state: The server is working on LP
Decision: Continue working on LP or initiate a switch from LP to HP?

Figure 2: Decision actions

exponential working phases for the server, and allowing for a general caseload. Using a Markov decision process (MDP) approach, we compute the optimal policy and evaluate the impact of the caseload.

2. **Policy $\mathcal{P}$ (Section 5).** We then address the problem for service encounters with multi-interludes ($N \geq 3$) and generally distributed working phases. We propose and analyze a heuristic policy, labeled as Policy $\mathcal{P}$, that is not state-dependent for the use of the interludes. Such a policy also provides managerial simplicity. Under Policy $\mathcal{P}$, we derive performance measures in closed-form, allowing us to further characterize the preference for when to use the interludes for LP work.

We end this section by a summary of the Notations used throughout the paper in Table 1.

4 Analysis of the optimal policy for investigator systems

In this section, we focus on solving Problem (1). To this end, we develop a Markov decision process approach in Section 4.1 to determine the optimal policy. Next, in Section 4.2, we consider special cases for the system parameters where structural properties of the optimal policy can be proven. Finally, in Section 4.3, we conduct a numerical study to understand the impact of the system parameters on the optimal policy. In order to define a Markov decision process for this system and to avoid dimensionality explosion, we assume here that working phases are exponentially distributed, and that the number of interludes is at most one (i.e., $N = 1$ or $N = 3$).

4.1 Markov decision process approach

We consider the case $N = 3$ and formulate the problem via the definition of states, the transition structure and the possible actions. The simpler case $N = 1$, can be easily deduced from the case $N = 3$. We denote by μ_1, μ_2, and μ_I, the exponential rates of the first working phase, the last one, and the interlude duration, respectively and by c the caseload. Recall that μ_{S_1} and μ_{S_2} are defined as the exponential rates for the switch from LP to HP and for the switch from HP to LP, respectively. We define a state of the system by $(\Omega, \vec{v})$, where Ω is the state of the server; and $\vec{v} = (x, y, z)$, for $x, y, z \geq 0$, where x, y, and z are the number of HPs in Queue 1 and in the first service phase, in

Table 1: Notations

Exogenous parameters	
λ	Arrival rate
N	Number of working and interlude phases
$\overline{x}$	Expected service time which includes the interlude time
cv	Coefficient of variation of the service time distribution; it is the ratio of the standard deviation divided by the expected value
$g(.)$	Probability density function of the service time which includes the interlude time
$\widetilde{G}(.)$	Laplace-Stieltjes Transform (LST) of the service time; $\widetilde{G}(s) = \int_{x=0}^{\infty} g(x)e^{-sx}\,\mathrm{d}x$
μ_{S_1}, μ_{S_2}	Exponential switching rate from LP to HP or from HP to LP, respectively ($a_{S_i} = \frac{\lambda}{\mu_{S_i}}$, for $i = 1, 2$)
μ_S	Exponential switching rate when $\mu_S = \mu_{S_1} = \mu_{S_2}$ ($a_S = \frac{\lambda}{\mu_S}$)
μ_I	Exponential interlude time rate when $N = 3$ ($t_I = 1/\mu_I$ and $a_I = \lambda/\mu_I$)
μ_{I_i}	Exponential interlude time rate of the i^{th} interlude when $N > 3$
$\overline{\omega}$	Waiting time objective
Control parameters and state definition for the optimal policy with Caseload= 1, and $N = 3$	
x, y, z	Number of HPs in Queue 1 and in the first service phase, in Queue 2 and in the second service phase, or with the non-focal server, respectively
H	Server working on HPs, idle and waiting for an HP, or in switch from HP to LP
L	Server working on LPs or in switch from LP to HP
u_{I_1}, u_{I_2}, u_B	Thresholds on the number of customers in Queue 1 for the switch from LP to HP respectively depending on whether the server is in an interlude and an HP is with the non-focal server, the server is in an interlude and an HP is in Queue 2, or no HP is with the non-focal server nor in Queue 2, respectively
$\tilde{u_I}$	Threshold on the number of customers in Queue 1 for the switch from HP to LP during the interlude
Control parameters of Policy $\mathcal{P}$	
n^*	Threshold on the number of customers in Queue 1 above which the server chooses to switch from LP to HP in order to serve high priority jobs ($n^* \geq 0$)
p^*	Probability to start working on LP during the interlude (with probability $1 - p^*$ the server remains available for directly serving HP)
q^*	Probability to start working on LP after a switch when the system is empty of HP (with probability $1 - q^*$ the server remains available for directly serving HP)
t^*	Extra time spent on LP before a switch during the interlude when the customer is ready for service completion
Random Variables	
X	Time actively spent by the server on serving the customer (it excludes the interlude times)
I	Time during which the customer is busy without needing the server. I is exponentially distributed with parameter μ_I
X_{S_1}, X_{S_2}	Switching times respectively from HP to LP and from LP to HP, respectively
T	Proportion of time spent on LP
W	Overall waiting time for a given customer
S	Overall time during which the server is without the customer (either waiting for the customer, in switch or busy with LP)
Probabilities for the performance evaluation under Policy $\mathcal{P}$	
$p_t(n, r)$	Probability-density of having n customers in the system, $n \geq 1$ and a remaining service time of r, $r \geq 0$, at time t (given some arbitrary initial distribution)
$p(n, r)$	$p(n, r) = \lim_{t\to\infty} p_t(n, r)$, for $n \geq 1$
$\widetilde{P}(n, s)$	$\widetilde{P}(n, s) = \int_{r=0}^{\infty} e^{-sr}p(n, r)\,\mathrm{d}r$ is the LST associated with $p(n, r)$
π_n	Stationary probability to have n customers in the system when the server is busy with high priority tasks, for $n \geq 1$ ($\pi_n = \int_{r=0}^{\infty} p(n, r)\,\mathrm{d}r$)
$\pi_{0,s}$	Stationary probability to be in a system with no high priority tasks and a server in switch from HP to LP
$\pi_{0,0}$	Stationary probability to be in a system with no high priority tasks and a server reserved for LP (not working on LP)
π_0	$\pi_0 = \pi_{0,s} + \pi_{0,0}$
ϕ_n	Stationary probability to have n customers in the system when the server is busy with low priority tasks, for $0 \leq n \leq n^*$ or in switch from LP to HP for $n > n^*$

Queue 2 and in the second service phase, and with the non-focal server, respectively. The server can be working on HP, idle and waiting for an HP, or be in switch from HP to LP. In these cases, the state of the server is denoted by H and the server treats HP as long as an available HP is present in Queue 1 or in Queue 2. Otherwise, the server can be dedicated to LP or in switch from LP to HP. In these cases, the state of the server is denoted by L and the server is not allowed to treat HP even if there are HPs in Queue 1 or in Queue 2.

We next describe the possible transitions and actions from a given state $(\Omega, \vec{v})$, for $\Omega \in \{H, L\}$ and $\vec{v} \in \mathbb{N}^3$. We denote by $\vec{e_i}$ the vector $(\delta_{i,1}, \delta_{i,2}, \delta_{i,3})$ for $i = 1, 2, 3$, where $\delta_{i,j}$ is the Kronecker delta; $\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ otherwise.

1. An arrival at Queue 1 with rate λ. The number of HPs in Queue 1 and in the first service phase is increased by 1, which changes the state to $(\Omega, \vec{v} + \vec{e_1})$.

2. An arrival at Queue 2 with rate $z\mu_I$. The number of HPs with the non-focal server is reduced by 1 and the number of HPs in Queue 2 or in the second phase of service is increased by 1. This changes the state to $(\Omega, \vec{v} - \vec{e_3} + \vec{e_2})$.

3. A service phase completion can be decided either from Queue 1 or from Queue 2 with rate μ_1 or μ_2, respectively. If $\Omega = H$, $x > 0$, $y = 0$, and $z < c$, then the server serves an HP from Queue 1. If $y > 0$ and either $x = 0$ or $y + z = c$, then the server serves an HP from Queue 2. Finally, if $x > 0$, $y > 0$, and $y + z < c$, then the server chooses the minimizing action between serving an HP from Queue 1 or from Queue 2. When a service phase completion from Queue 1 occurs, the number of HPs with the non-focal server is increased by 1 and the number of HPs in Queue 1 or in the first phase of service is reduced by 1. This changes the state to $(H, \vec{v} - \vec{e_1} + \vec{e_3})$. When a service phase completion from Queue 2 occurs, the number of HPs in Queue 2 or in the second phase of service is reduced by 1. This changes the state to $(H, \vec{v} - \vec{e_2})$.

4. A switch from HP to LP can be decided if $\Omega = H$, $x = y = 0$ (i.e., the two queues are empty) or if $y = 0$ and $z = c$ (i.e., Queue 2 is empty and the number of HPs with the non-focal server attains the caseload). If this choice is made, the state changes to $(L, \vec{v})$ upon the switch completion with rate μ_{S_2}.

5. A switch completion from LP to HP can be decided if $\Omega = L$. If this choice is made, the state changes to $(H, \vec{v})$ upon the switch completion with rate μ_{S_1}.

With finite caseload, the maximal event rate, $\lambda + c\mu_I + \mu_{S_1} + \mu_{S_2} + \mu_1 + \mu_2$, is bounded. This continuous-time model is therefore uniformizable. We then choose to discretize it (Section 11.5.2. in Puterman (1994)). We assume that $\lambda + c\mu_I + \mu_{S_1} + \mu_{S_2} + \mu_1 + \mu_2 = 1$, such that the rate out of each state is equal to 1 by adding fictitious transitions from a state to itself; then we can consider the rates to be transition probabilities. Note that the system with infinite caseload can only be approximated with a sufficiently high finite caseload. We define the dynamic programming value functions $V_k(\Omega, \vec{v})$ over $k \geq 0$ steps, depending on the state of the system $(\Omega, \vec{v})$, for $\Omega \in \{L, H\}$ and $\vec{v} \in \mathbb{N}^3$. We choose $V_0(\Omega, \vec{v}) = 0$, for $\Omega \in \{L, H\}$ and $\vec{v} \in \mathbb{N}^3$. Next, we express $V_{k+1}(\Omega, \vec{v})$ in terms of $V_k(\Omega, \vec{v})$ in the following way. The objective to maximize the time spent on LP and the constraint for HP are merged into a single cost function using a Lagrange parameter γ which accounts for the time spent by an HP in Queue 1 or in Queue 2 and with the focal server. The value of γ is chosen such that for the optimal policy $E(W) = \overline{w}$ (Altman, 1999). The procedure to set the value of γ is presented in Section 1 of the Online Appendix. Moreover, we count a reward of one when the server is treating some LPs in order to obtain the proportion of time spent on LPs. The objective for the server is to determine when to initiate the switches from HP to LP and those from LP to HP. The optimal decisions are captured by the minimization operator, $V_k^*(\Omega, \vec{v})$, in the value function formulation.

We write for $k \geq 0$, $\Omega \in \{L, H\}$ and $\vec{v} \in \mathbb{N}^3$:

$$\begin{aligned}
V_{k+1}(H,\vec{v}) = {} & \frac{\gamma(x+y)}{\lambda} + \lambda V_k(H,\vec{v}+\vec{e_1}) + z\mu_I V_k(H,\vec{v}-\vec{e_3}+\vec{e_2}) \\
& + \mathbb{1}_{x>0,y=0,z<c}\mu_1 V_k(H,\vec{v}-\vec{e_1}+\vec{e_3}) \\
& + \mathbb{1}_{y>0,(x=0\cup y+z=c)}\mu_2 V_k(H,\vec{v}-\vec{e_2}) + \mathbb{1}_{y=0,(x=0\cup z=c)}\mu_{S_2} V_k^*(H,\vec{v}) \\
& + \mathbb{1}_{x>0,y>0,y+z<c} \min\left(\mu_1(V_k(H,\vec{v}-\vec{e_1}+\vec{e_3}) - V_k(H,\vec{v})), \mu_2(V_k(H,\vec{v}-\vec{e_2}) - V_k(H,\vec{v}))\right) \\
& + (1 - \lambda - z\mu_I - \mu_1 \mathbb{1}_{x>0,y=0,z<c} - \mu_2 \mathbb{1}_{y>0,(x=0\cup y+z=c)} - \mu_{S_2} \mathbb{1}_{y=0,(x=0\cup z=c)}) V_k(H,\vec{v}), \text{ and,} \\
V_{k+1}(L,\vec{v}) = {} & \frac{\gamma(x+y)}{\lambda} - 1 + \lambda V_k(L,\vec{v}+\vec{e_1}) + z\mu_I V_k(L,\vec{v}-\vec{e_3}+\vec{e_2}) \\
& + \mu_{S_1} V_k^*(L,\vec{v}) + (1 - \lambda - z\mu_I - \mu_{S_1}) V_k(L,\vec{v}),
\end{aligned} \tag{2}$$

where $\mathbb{1}_{x\in A}$ is the indicator function of a given subset A, and $V_k^*(L,\vec{v}) = \min(V_k(H,\vec{v})+1, V_k(L,\vec{v}))$ (i.e., switching decision from LP to HP), and $V_k^*(H,\vec{v}) = \min(V_k(H,\vec{v}), V_k(L,\vec{v}))$ (i.e., switching decision from HP to LP), if $\vec{v} = z\vec{e_3}$ for $z \geq 0$ or $\vec{v} = x\vec{e_1} + c\vec{e_3}$, for $x \geq 0$ and, $V_k^*(H,\vec{v}) = V_k(H,\vec{v})$, otherwise. In Equation (2), the terms $\frac{\gamma(x+y)}{\lambda}$ measure the expected wait in the system. The terms proportional with λ represent a customer's arrival in Queue 1. The terms proportional with $z\mu_I$ correspond to an arrival in Queue 2 from a customer who completes a service with the non-focal server. The terms proportional with μ_i corresponds to the end of working phase i for $i = 1, 2$. The terms proportional with μ_{S_1} and μ_{S_2} capture the possibility to start an HP or an LP after a switch time. The minimizing operator in the first part of the equation optimizes the decision for serving a customer from Queue 1 or from Queue 2. The last term gives the fictitious transition from a state to itself. Note that a cost of one is counted if a switch from LP to HP is executed in order not to count the switching time into the time spent on LP.

To obtain the infinite horizon average optimal actions we rely on the value iteration technique by recursively evaluating V_k using Equation (2), for $k \geq 0$. As k tends to infinity, the optimal policy converges to the unique average optimal policy. This convergence result is ensured by Theorem 8.10.1 in Puterman (1994) (a countable state set, finite set of actions and a uniformizable system). As k tends to infinity, the difference $V_{k+1} - V_k$ converges to $\gamma E(W) - E(T)$ (Puterman, 1994). Since $E(W) = \overline{w}$ for the optimal policy, one can then easily compute the value of $E(T)$. Note that by redefining the cost associated with the number of HPs in Queue 1 or in Queue 2, one can also obtain $E(W_1)$ and $E(W_2)$ using value iteration.

The numerical investigations show that the optimal policy for the switches from LP to HP or from HP to LP either during the interlude or during the break is a *state-dependent threshold policy.* This policy is characterized by the following properties:

- If it is optimal to switch from LP to HP in state (L, x, y, z), then it is also optimal to operate this switch in states $(L, x+1, y, z)$, $(L, x, y+1, z)$, and $(L, x, y, z+1)$, for $x \geq 0$ and $0 \leq y+z < c$.
- If it is optimal to switch from HP to LP in state $(H, 0, 0, z+1)$ (respectively, in state $(H, x+1, 0, c)$), then it is also optimal to operate this switch in state $(H, 0, 0, z)$ (respectively, in state $(H, x, 0, c)$), for $0 \leq z < c$ and $x \geq 0$.

Even though, we can compute the optimal policy numerically, it is not possible to prove its properties using an induction step for the general case. In the following section, considering some special cases of the system parameters, we prove some of the structural properties of the optimal policy. These properties will be used in Section 5 to propose a simpler policy to solve Problem (1).

4.2 Structural properties of the optimal policy in special cases

First, we consider the case $N = 1$, where the service of an HP is executed in one single stage without interlude. The analysis of this case enables us to better characterize the optimal decisions that the server should make during the *break*. Next, we investigate the case $N = 3$ with a caseload$= 1$, where the optimal decisions during the *interlude* can be proven. Finally, we investigate the case $N = 3$ with infinite caseload and a strict priority for Queue 1 in order to prove how the state variables x, y, and z impact the optimal decisions.

4.2.1 Analysis of the case $N = 1$

Without interludes, the value function can be reformulated in a simpler form as the system becomes one-dimensional. Equation (2) is then reformulated as

$$\begin{aligned} V_{k+1}(H,x) &= \frac{\gamma x}{\lambda} + \lambda V_k(H,x+1) + \mathbb{1}_{x>0}\mu_1 V_k(H,x-1) + \mu_{S_2}\mathbb{1}_{x=0}V_k^*(H,0) \\ &\quad + (1-\lambda-\mu_1\mathbb{1}_{x>0}-\mu_{S_2}\mathbb{1}_{x=0})V_k(H,x), \text{ and,} \\ V_{k+1}(L,x) &= \frac{\gamma x}{\lambda} - 1 + \lambda V_k(L,x+1) + \mu_{S_1}V_k^*(L,x) + (1-\lambda-\mu_{S_1})V_k(L,x), \end{aligned} \tag{3}$$

where x is the number of customers in the system, with $V_k^*(L,x) = \min(V_k(H,x)+1, V_k(L,x))$ (i.e., switching decision from LP to HP), and $V_k^*(H,x) = \min(V_k(H,x), V_k(L,x))$. In Proposition 1, we prove, under some conditions reflecting a high care for customer's wait and a sufficiently long switching time, that the optimal policy for the use of the break is a threshold policy. This result extends those for the queue blending literature without switches, and will be used in Section 5 to construct Policy $\mathcal{P}$.

Proposition 1 *With $\mu_{S_2} \leq \mu_1 \leq 2\mu_{S_2}$ and $\gamma \geq \lambda$, the optimal policy for the use of the break is defined by the parameters n^* and q^*, such that a switch from LP to HP is initiated whenever the number of HPs in Queue 1 is strictly above n^* and the switch from HP to LP is initiated with probability q^* when the system becomes empty.*

The proof of Proposition 1 is given in Section 2 of the Online Appendix. The parameter n^* is a threshold which defines a *deterministic* state-dependent priority for HP. The parameter q^* allows the system to provide something stronger than a simple priority. In the case $n^* = 0$ and $q^* = 1$, a strict priority is given to HP in a work-conserving situation. If $q^* = 0$, no LPs are treated in between HP in order to reserve the server for HP.

4.2.2 Analysis of the case $N = 3$ with caseload$= 1$

For such a setting, the optimal policy for the switch from LP to HP is simpler due to the constraint $0 \leq y + z \leq 1$. Moreover, with caseload$= 1$, Queue 2 has a strict priority over Queue 1. Therefore, Equation (2) can be rewritten as follows:

$$\begin{aligned} V_{k+1}(H,\vec{v}) &= \frac{\gamma(x+y)}{\lambda} + \lambda V_k(H,\vec{v}+\vec{e_1}) + z\mu_I V_k(H,\vec{v}-\vec{e_3}+\vec{e_2}) \\ &\quad + \mathbb{1}_{x>0,y=0,z=0}\mu_1 V_k(H,\vec{v}-\vec{e_1}+\vec{e_3}) + \mathbb{1}_{y>0}\mu_2 V_k(H,\vec{v}-\vec{e_2}) + \mathbb{1}_{y=0,(x=0\cup z=1)}\mu_{S_2}V_k^*(H,\vec{v}) \\ &\quad + (1-\lambda-z\mu_I-\mu_1\mathbb{1}_{x>0,y=0,z=0}-\mu_2\mathbb{1}_{y>0}-\mu_{S_2}\mathbb{1}_{y=0,(x=0\cup z=1)})V_k(H,\vec{v}), \text{ and,} \\ V_{k+1}(L,\vec{v}) &= \frac{\gamma(x+y)}{\lambda} - 1 + \lambda V_k(L,\vec{v}+\vec{e_1}) + z\mu_I V_k(L,\vec{v}-\vec{e_3}+\vec{e_2}) \\ &\quad + \mu_{S_1}V_k^*(L,\vec{v}) + (1-\lambda-z\mu_I-\mu_{S_1})V_k(L,\vec{v}). \end{aligned} \tag{4}$$

For the switch from LP to HP, we observe numerically that there exist three thresholds on the number of HPs in Queue 1 denoted by u_{I_1}, u_{I_2}, and u_B, depending on whether the server is in an interlude and an HP is with the non-focal server ($y = 0, z = 1$), the server is in an interlude and an HP is in Queue 2 ($y = 1, z = 0$), or no HP is with the non-focal server nor in Queue 2 (i.e., a server's break: $y = z = 0$), such that the decision to switch from LP to HP is taken if and only if the number of HPs in Queue 1 is above these thresholds. A switch from HP to LP, may occur either if $x = y = z = 0$ (switch during the break) or if $y = 0, z = 1$ and $x < \tilde{u}_I$ (switch during the interlude). In Proposition 2, we show the threshold structure of the optimal policy during the interlude for the switch from LP to HP and from HP to LP. Moreover, we show that $u_{I_1} \geq u_{I_2}$. The proof of Proposition 2 is given in Section 3 of the Online Appendix. Note that the threshold structure of the optimal policy for the switch from LP to HP during the break cannot be proven with this technique due to the caseload limitation which breaks the monotonicity properties of the value function.

Proposition 2 *For caseload$= 1$, the optimal policy during the interlude is of threshold type. There exists thresholds u_{I_1}, u_{I_2} and $\tilde{u}_I$ such that:*

- *The switch from LP to HP is initiated if and only if $x \geq u_{I_1}$ with $y = 0$ and $z = 1$, or $x \geq u_{I_2}$ with $y = 1$ and $z = 0$. Moreover, $u_{I_1} \geq u_{I_2}$.*
- *The switch from HP to LP during an interlude is initiated if and only if $x < \tilde{u}_I$.*

4.2.3 Analysis of the case $N = 3$ with infinite caseload and priority for Queue 1

With infinite (or unrestricted) caseload, some properties of the optimal policy can be proven. However, we need to simplify the model assumptions by giving strict priority to Queue 1 instead of optimizing the queue priority and by assuming $\mu_1 = \mu_2 = \mu_{S_2}$. Moreover, as the system is non-uniformizable with infinite caseload, the MDP approach can be implemented only if the real system is approximated by a sufficiently high finite caseload such that uniformization is possible and the effect of the caseload restriction can be neglected.

Even though the problem becomes three-dimensional with infinite caseload, as the switch from HP to LP can only be operated if $x = y = 0$, the decision to switch from HP to LP only depends on the state variable z. This simplifies the analysis as compared to a case with a finite caseload.

Under the aforementioned assumptions, in Proposition 3, we prove the threshold structure of the optimal policy in the variables y and z. Yet, we are not able to show the same result in the variable x as some of the monotonicity properties do not propagate via an induction step. However, by further replacing the expected wait $E(W)$ in Problem (1) by the expected wait in Queue 1, $E(W_1)$, the optimal policy can be fully proven. This objective is compatible with a priority to Queue 1. The proof of Proposition 3 is given in Section 4 of the Online Appendix. As reflected in this proposition, the optimal policy may be impracticable due to its complexity. Therefore, in Section 5, we simplify the analysis by considering non-state-dependent decisions during the interlude.

Proposition 3 *In the case $\mu_1 = \mu_2 = \mu_{S_2}$, without caseload restriction, with priority for Queue 1, the optimal policy is a threshold type policy with the following properties:*

- ***Switch from HP to LP.*** *There exists a threshold on the number of customers with the non-focal server, $\tilde{u}$, such that the server initiates a switch from HP to LP if and only if both queues are empty and the number of HPs with the non-focal server is strictly below this threshold; $z < \tilde{u}$.*
- ***Switch from LP to HP.*** *For each $x \geq 0$, there exists a threshold function $z = u_x(y)$, decreasing in y, such that if the server is working on LP in state (x, y, z) then it is optimal to switch from LP to HP if and only if $z \geq u_x(y)$.*

Under the same conditions, while considering $E(W_1)$ instead of $E(W)$ in the objective, the optimal policy is also of threshold type but the switch from LP to HP can be further characterized as follows:

Table 2: Computed thresholds with $N = 3$ and caseload= 1

	Exogenous parameters							Thresholds for the switch from LP to HP			Threshold for the switch from HP to LP	
	$\overline{\omega}$	λ	μ_1	μ_2	μ_I	μ_{S_1}	μ_{S_2}	u_{I_1}	u_{I_2}	u_B	$\tilde{u}_I$	$E(T)$
1	5	0.05	1	1	1	10	10	1	1	1	1	88.98%
2	5	0.1	1	1	1	10	10	1	1	1	1	78.89%
3	5	0.2	1	1	1	10	10	2	1	1	2	55.94%
4	5	0.05	1	1	1	1	1	1	1	1	1	87.24%
5	5	0.1	1	1	1	1	1	1	1	2	1	68.96%
6	5	0.2	1	1	1	1	1	2	1	2	1	34.97%
7	5	0.05	1	1	1	0.05	0.05	11	10	13	1	11.17%
8	5	0.1	1	1	1	0.05	0.05	5	2	3	1	6.48%
9	5	0.2	1	1	1	0.05	0.05	4	2	2	1	1.03%
10	10	0.05	1	1	1	1	1	2	1	2	1	88.86%
11	10	0.1	1	1	1	1	1	4	2	2	4	71.79%
12	10	0.2	1	1	1	1	1	6	4	6	5	39.20%
13	5	0.2	2	2	0.5	1	1	2	2	3	2	33.52%
14	5	0.2	0.71	0.71	5	1	1	2	1	1	1	32.29%
15	5	0.2	1	10	0.53	1	1	3	1	3	3	32.04%
16	5	0.2	10	1	0.53	1	1	2	1	3	2	31.81%
17	10	0.1	1	1	1	0.67	2	5	2	3	6	73.35%
18	10	0.1	1	1	1	2	0.67	3	2	2	3	67.31%

Switch from LP to HP. *There exists a threshold function $z = u_1(x, y)$, decreasing in the variables x and y, such that if the server is working on LP in state (x, y, z) then it is optimal to switch from LP to HP if and only if $z \geq u_1(x, y)$.*

4.3 Numerical analysis

We end this section with a numerical analysis to better understand how the optimal policy is influenced by the environmental conditions. First, in Section 4.3.1, we investigate the effect of the system parameters on the optimal policy in the case $N = 3$ and caseload= 1. Next, in Section 4.3.2, we evaluate the role of the caseload.

4.3.1 Effect of the system parameters

Table 2 tabulates the thresholds for the policy computed with Equation (4) (see Section 4.2.2) for different values of the system parameters under the stability condition $\lambda < \left(\mu_1^{-1} + \mu_I^{-1} + \mu_2^{-1}\right)^{-1}$. Note that the service rates are chosen such that the expected service time of an HP is constant (i.e., $\mu_1^{-1} + \mu_I^{-1} + \mu_2^{-1}$ is constant).

Consistent with Proposition 2, we observe that $u_{I_1} \geq u_{I_2}$. We also observe that $u_{I_1} \geq \tilde{u}_I$. This means that if a switch from HP to LP is decided in a given state, then in the same state the server should not initiate a switch back from LP to HP. In most cases, we also observe that $u_B \geq u_{I_1}$: For a given number of HPs in Queue 1, one more HP is present in the system during an interlude (either with the non-focal server or in Queue 2) than during a break. Hence, it is more urgent to switch from LP to HP during an interlude than during a break which is represented by $u_B \geq u_{I_1}$. Surprisingly, counterexamples can be found for low values of μ_S (see lines 8 and 9). The explanation comes from the long switching times. If a decision to treat LP during the interlude is taken (i.e., at state $(x, 0, 1)$), then the HPs already waiting in Queue 1 suffer from a first switch from HP to LP and from an additional one from LP to HP. The decision to treat LP during the break can only be taken in an empty system due to the priority for HP, so no HP is impacted by the switch from HP to LP. (Recall that an HP who arrives during a switch from HP to LP is directly served.) During the break, arriving HPs are only impacted by the switch from LP to HP. To reduce the negative effect of a double switch during the interlude compared to a single one during the break, the server should continue working on LP for a larger number of waiting HPs during the interlude than during the break. This behavior of the optimal policy explains why it is not possible to show the monotone behavior of the optimal policy as a function of the congestion in Queue 1 during the break. The preference for using the break or the interlude will be further investigated via a sensitivity analysis under policy $\mathcal{P}$ in Section 5.

Table 2 is organized in order to explore the impact of various system parameters.

- Comparison of lines 4-6 with lines 10-12 shows the role played by the service level objective, $\overline{\omega}$. We observe that the thresholds increase when the desired service level on HP is less strict, i.e., when these requests become less urgent, and consequently more time is dedicated to LP work as seen in the increased values of $E(T)$ in the last column.

- Comparison of lines 1-3 with lines 4-6 or with lines 7-9 shows the important role played by the switching times when switches from HP to LP and from LP to HP have the same expected duration. With long switching times, the server should postpone the switch from LP to HP. This is reflected by high thresholds for the switch from LP to HP (u_{I_1}, u_{I_2}, and u_B). The last column shows the high negative impact that switching times may have on the expected time spent on LP. In order to avoid making too frequent switches, we also could expect to have $\tilde{u_I}$ decreasing with the length of the switching time. However, since $\tilde{u_I}$ has reached its minimal value already with $\mu_{S_1} = \mu_{S_2} = 10$, no further reduction can be observed while increasing the switching rate.

- The examples in lines 13-14 highlight the impact of the interlude duration. A long interlude duration is an incentive to use this interlude to treat LP. This can be observed with u_B increasing from 1 to 3 when μ_I changes from 5 to 0.5, making the interlude duration ten times longer.

- Lines 15-16 provide examples where the two working phases have different duration. The observations here can only be made in cases where the rates μ_1 and μ_2 are significantly different. If $\mu_2 >> \mu_1$, then an HP does not represent a significant amount of remaining work for the server after the completion of the first working phase. Therefore, the decisions of the server during the interlude are close to those during the break as observed at line 15 ($u_B = u_{I_1}$). If $\mu_2 << \mu_1$, the important amount of remaining work after the completion of the first working phase tends to reduce the value of using the interlude and is reflected by smaller thresholds (u_{I_1} and $\tilde{u_I}$).

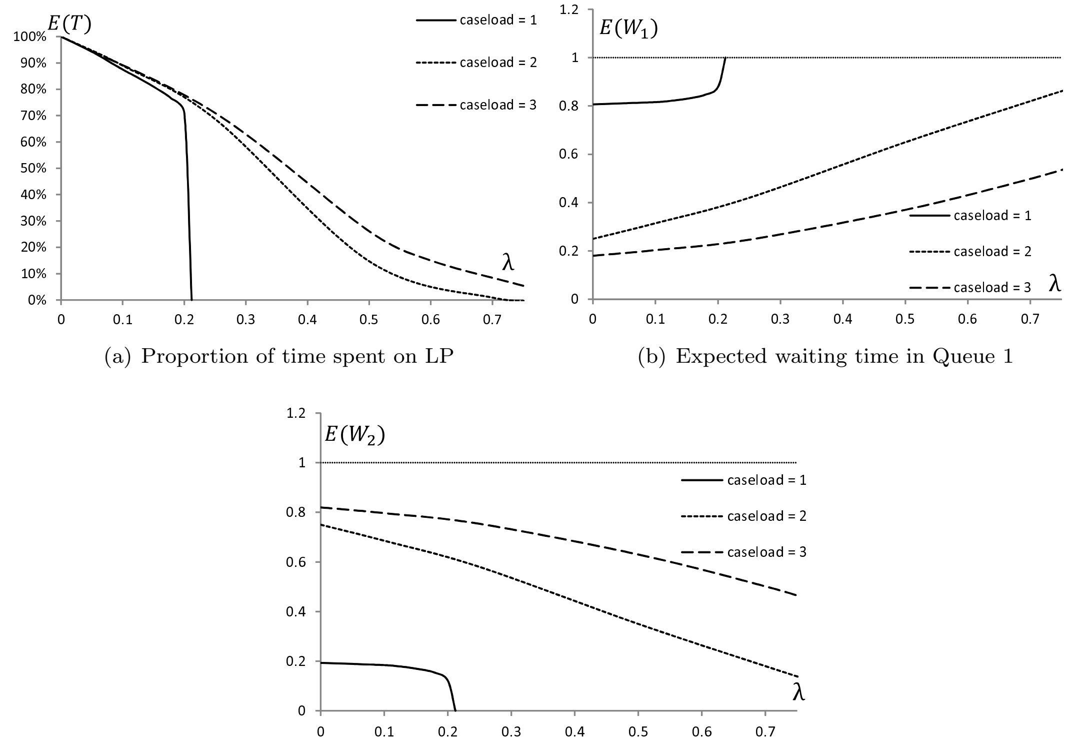
- Lines 11, 17 and 18 illustrate the impact of having different switching duration from LP to HP and from HP to LP. We choose $\frac{1}{\mu_{S_1}} + \frac{1}{\mu_{S_2}} = 2$ such that the sum of the expected duration of the switches is kept constant. We observe that a long switch from HP to LP is less problematic than a long switch from LP to HP. This observation is due to the beneficial possibility of preemption by an HP when the server is in switch from HP to LP while this preemption is not possible for the switch back from LP to HP. We observe in particular that the thresholds are higher at line 17 than at line 18 which reflects longer periods of time spent on LPs.

- The impact of the arrival rate is less straightforward. In some examples we observe that the thresholds increase with the arrival rate (see lines 1 to 3 or lines 4 to 6), while in others the threshold decrease (see lines 7 to 9). This may be the result of the competition between two phenomena. With high arrival rates, the proportion of time spent on HP should be important in order to attain the service level constraint for HP. This would imply decreasing thresholds as the arrival rate increases, as observed in lines 7-9. However, lines 1-3 and 4-6 contradict this conclusion. It is in fact not good to have high thresholds with low arrival rates (see lines 1 and 4). Consider a situation where the server is treating LP and one HP is at the head of Queue 1. The switch from LP to HP will be executed when the number of HPs in Queue 1 attains the threshold. With a low arrival rate, the time between two arrivals is long. Hence, high thresholds would lead to an excessive wait for the HP at the head of the queue. This explains why with a low arrival rate, it makes sense to also have low thresholds.

4.3.2 Impact of the caseload

A caseload restriction may be imposed either for managerial reasons (a quality concern in a healthcare setting) or to limit the overall service time of an HP. Clearly, the caseload creates an additional constraint for the optimization problem, which can only deteriorate the solution. We investigate the impact of the caseload on $E(W_1)$, $E(W_2)$ and $E(T)$. Using the computed policy with Equation (2), we determine the performance measures by adjusting the cost parameters in the value function.

In Figure 3, we present the performance measures as a function of the arrival rate λ for three values of the caseload (1, 2 and 3). We first observe that as the caseload increases, the interval for λ which allows the system to achieve $E(W) = \varpi$ increases. This observation can be related to the stability of the system which increases with the caseload. The second observation is that the proportion of time spent on LP is higher with a higher caseload. The difference can be significant for high arrival rates (see Figure 3(a)). So, clearly, an unrestricted caseload is the best choice for Problem (1).



(a) Proportion of time spent on LP

(b) Expected waiting time in Queue 1

(c) Expected waiting time in Queue 2

Figure 3: Impact of the caseload ($\mu_1 = \mu_2 = 2$, $\mu_I = 1$, $\mu_{S_1} = \mu_{S_2} = 4$, $\varpi = 1$)

Another observation is that although $E(W) = E(W_1) + E(W_2)$ is maintained constant, $E(W_1)$ (respectively, $E(W_2)$) increases (respectively, decreases) with the arrival rate and decreases (respectively, increases) with the caseload. Since the caseload limits the number of HPs in Queue 2, increasing the caseload allows more HPs to wait in Queue 2 and clearly leads to higher values for $E(W_2)$. The caseload limits the congestion in Queue 2. Hence, the congestion increases more in Queue 1 than in Queue 2, when the arrival rate increases. This explains why $E(W_1)$ is increasing in λ. The caseload restriction gives a state-dependent server reservation for Queue 2 (i.e., if too many HPs are either in Queue 2 or with the non-focal server, then the server is not allowed to initiate an HP from Queue 1). This explains why for a low caseload and a high arrival rate we have $E(W_2) < E(W_1)$. Having a small in-service wait, $E(W_2)$, can be important in some contexts. For example, in the after-sales technical service setting, it may be desirable to complete the repair of a critical item urgently, or the general practitioner may wish to complete the initial treatment of a serious patient in order to direct them to a specialist without delay. In a chat service context, customers may be more annoyed by an in-chat wait than a pre-chat wait. Despite the increase in performance for Problem (1) as the caseload is increased, we note that the improvement is decreasing in caseload, suggesting an eventual saturation. Taken together with the fact that the in-service waiting time increases, one is left with the recommendation that the caseload should not be increased beyond several customers.

5 Analysis for general multi-stage encounters under a restricted policy

In this section, we consider multi-stage service encounters like the earlier mentioned audit process or general practitioner examples, possibly having more than one interlude. Given the fixed number of stages assumption made, as stated in the modeling section, we are considering mostly processes with standard operating procedures or protocols. This in turn suggests that an exponential working phase may be too restrictive as an assumption. Therefore, we wish to consider general service times for the working phases. The analysis of Section 4 shows us that even in the presence of a single interlude, optimal policies during interludes are state-dependent, with state-dependent thresholds. This creates a very complex policy to implement. It is clear that for a setting with multiple interludes, policies will be even more complex. The desire to have general service times will further complicate the analysis, with policies that need to depend on the remaining service time distribution of HP. In addition to being analytically intractable, state-dependent policies for use of the interludes may also not be practical. Having a single policy for all interludes may be preferred. Thus, motivated by both tractability and managerial simplicity concerns, we limit the analysis of Problem (1) for general multi-stage encounters to a restricted class of policies called Policy $\mathcal{P}$. The idea is to keep the policy for use of the break state-dependent as shown to be optimal in Section 4.2.1, while making that for the interludes a static policy. Keeping the control during the breaks state-dependent is relatively easier, as this decision only depends on the number of customers in Queue 1. The benefit of using this simpler (non-optimal) policy is that we can characterize performance measures and some policy parameters analytically when caseload= 1. This in turn allows us to explore the general question of when one would be interested in using the interludes for back-office work in such general multi-stage service encounters.

This section is organized as follows. Section 5.1 defines Policy $\mathcal{P}$. Section 5.2 gives the explicit performance measures and partially characterizes the optimal control parameters in the case caseload= 1. Section 5.3 provides the results of the numerical investigation showing the main drivers for the use of the interlude or the break. Section 5.4 extends the analysis of Section 5.3 in order to answer managerial questions.

5.1 Definition of Policy $\mathcal{P}$

For Policy $\mathcal{P}$, we give a strict priority for Queue 2. The policy for the use of the break and the interlude is defined as follows:

The server's policy during a break. We propose a queue length dependent threshold priority for HP. We define a threshold n^* on the number of HPs in Queue 1 and a parameter q^* such that

- At a service completion of an HP, if the system is empty of HP (i.e., a break), then with probability q^* the server switches from HP to LP and with probability $1 - q^*$, she chooses to remain idle and thus available for directly serving HP (reservation). Recall that if an HP arrives before the end of the switching time, this HP is directly served.
- If the server is working on LP, the server initiates a switch from LP to HP if and only if the number of HPs in Queue 1 is strictly above n^*.

Note that this policy is identical to the one found in Proposition 1.

The server's policy during an interlude. During an interlude, we choose a non-state dependent policy. At the beginning of an interlude, the server automatically switches from HP to LP with probability p^* and works until an HP comes back. When the customer comes back,

- if the server is still in switch from HP to LP then the customer is directly served,
- if the server is working on LP then the server continues working on LP during t^* time units then switches from LP to HP before continuing the service of the HP customer.

The possibility to continue working on LP for a duration t^* allows the server to reduce the negative impact of the switching times by working longer on LP. Note that in the case $p^* = t^* = 0$, the server is reserved for HP, in the case $p^* = 1$ and $t^* = 0$, then a strict priority is given to HP but the server adopts a work-conserving strategy. The definition of this policy can be extended to a case with multiple interludes by setting a parameter p^* and t^* for each interlude.

The randomizing parameters q^* and p^* are useful by allowing exact attainment of the service level constraint for HP. The parameters p^* and q^* control the decision to switch from HP to LP whereas the parameters n^* and t^* control the decision to switch back from LP to HP. From a simulation study, we compare the performance of Policy $\mathcal{P}$ to the optimal policy for investigator systems ($N = 3$) in Section 5 of the Online Appendix. We observe that it performs well, and its performance may even coincide with the optimal one in some cases (for high and low arrival rates). However, in cases with long interlude durations, short switching times and a high workload its performance begins to deviate from the optimal one.

5.2 Analysis of Policy $\mathcal{P}$ with caseload= 1

Theorem 1 gives closed-form expressions for the performance measures of interest for Policy $\mathcal{P}$ with caseload= 1. In order to simplify the notation, we assume that the service encounter includes only one interlude with an exponential rate μ_I. We denote by R, the random service time of an HP. It is defined as the interval of time from the initiation of the first phase of service of an HP to the last service phase completion. It includes the working phases, the time spent with the non-focal server and the wait in Queue 2. Hence, the service time can be decomposed as the sum of two random variables, X and S, where X is the time actively spent with the server (the working phases) and S is the time spent without the server (the time spent with the non-focal server plus the time spent in Queue 2), $R = X + S$. The system performance measures are functions of $E(R)$ and cv_R defined as the expected service time and the coefficient of variation of the service time R, respectively. The random variable X is given by the distributions of the working phases. What remains to be characterized is the random variable S. Hence, for the performance evaluation, we determine the first and the second moments of S; denoted by $E(S)$ and $E(S^2)$, respectively. Finally, we denote by a_{S_i} the ratio λ/μ_{S_i}, for $i = 1, 2$. The stability condition in this model is identical to the one of an M/G/1 queue since HPs are served one by one (caseload= 1), i.e., $\lambda E(R) < 1$. Note that the performance measures can easily be extended to the case of more than one interlude; while the parts of the expressions of $E(T)$ and $E(W)$ depending on the break would remain identical, other parts corresponding to additional interludes should be added. These additional terms would be obtained in a similar way as the parts related to single interlude.

The approach to compute the performance measures first consists of providing a recursive formula for the computation of the stationary probabilities, using a state definition based on the residual service time for a given customer. From the recursive relations for the stationary probabilities, we derive the probability of an empty system, the expected waiting time and the expected time spent on LP as a function of the system parameters, the first and the second moments of the time spent in service (which includes the time spent with the server plus the time spent with the non-focal server plus the time spent in Queue 2), and the expected time spent on LP during the interlude. The complete detailed proof is given in Section 6 of the Online Appendix.

Theorem 1 *The expected proportion of time spent on LP, $E(T)$, is*

$$E(T) = p^*\lambda \frac{\mu_{S_1}}{\mu_I + \mu_{S_1}} \left(t^* + \frac{1}{\mu_I} - \frac{1}{\mu_{S_1}} + \frac{\mu_I e^{-\mu_{S_1} t^*}}{\mu_{S_1}(\mu_{S_1} + \mu_I)} \right) + \frac{(1 - \lambda E(R))(n^* + 1)}{n^* + \frac{1}{q^*} + \frac{1}{q^*}\left(a_{S_1} + q^* a_{S_2}\right)}. \tag{5}$$

The expected waiting time, $E(W)$, is

$$\begin{aligned} E(W) = {} & p^* \frac{\mu_{S_1}}{\mu_{S_1} + \mu_I} \left(\frac{1}{\mu_{S_2}} + t^* \right) + \frac{\lambda E(R)^2 (1 + cv_R^2)}{2(1 - \lambda E(R))} + \frac{n^* \left(n^* + 1 + 2a_{S_2}\right)}{2\lambda \left(n^* + \frac{1}{q^*} + \frac{1}{q^*}\left(a_{S_1} + q^* a_{S_2}\right) \right)} \\ & + \frac{1}{\mu_{S_2}} \cdot \frac{1 + a_{S_2}}{n^* + \frac{1}{q^*} + \frac{1}{q^*}\left(a_{S_1} + q^* a_{S_2}\right)}. \end{aligned} \tag{6}$$

The first and second moments of S are

$$E(S) = \frac{1}{\mu_I} + p^* \frac{\mu_{S_1}}{\mu_{S_1} + \mu_I} \left(\frac{1}{\mu_{S_2}} + t^* \right), \tag{7}$$

$$E(S^2) = \frac{2}{\mu_I^2} + \frac{p^* \mu_{S_1}}{\mu_{S_1} + \mu_I} \left(\left(t^* + \frac{1}{\mu_{S_2}} \right)^2 + \frac{1}{\mu_{S_2}^2} + \frac{2}{\mu_I} \left(t^* + \frac{1}{\mu_{S_2}} \right) \right). \tag{8}$$

The expected time spent on LP, as given in Equation (5), is the sum of the expected time spent on LP during the interlude, $E(T_I) = p^*\lambda \frac{\mu_{S_1}}{\mu_I + \mu_{S_1}} \left(t^* + \frac{1}{\mu_I} - \frac{1}{\mu_{S_1}} + \frac{\mu_I e^{-\mu_{S_1} t^*}}{\mu_{S_1}(\mu_{S_1} + \mu_I)} \right)$, and the expected time spent on LP during the break, $E(T_B) = \frac{(1 - \lambda E(R))(n^* + 1)}{n^* + \frac{1}{q^*} + \frac{1}{q^*}\left(a_{S_1} + q^* a_{S_2}\right)}$. The expected waiting time, as given in Equation (6), is the sum of four terms. The first one is the in-service wait in Queue 2, $E(W_2) = p^* \frac{\mu_{S_1}}{\mu_{S_1} + \mu_I} \left(\frac{1}{\mu_{S_2}} + t^* \right)$, and the second one is similar to the expected waiting time in an M/G/1 queue. This second term shows that the variability of the working phases negatively affects the solution of Problem (1). The last two terms may be more complicated to interpret. Note that in the case of $q^* = 1$ and $\mu_{S_1} = \mu_{S_2}$, the third term is exactly equal to $\frac{n^*}{2\lambda}$ which corresponds to the effect of n^* on the waiting time ($n^*/2$ is the average queue size when the server is on LP).

Computation of the optimal threshold parameters under Policy $\mathcal{P}$. We proceed by first assuming that the policy for the use of the interlude is fixed (i.e., the parameters p^* and t^* are fixed) in order to derive closed-form expressions for the optimal values of n^* and q^*. Next, the values of p^* and t^* will be computed numerically. Since the use of the interlude is fixed, the difference $K = \overline{w} - E(W_2) - \frac{\lambda E(R)^2 (1 + cv_R^2)}{2(1 - \lambda E(R))}$ is also fixed. We assume that $K \geq 0$, otherwise the waiting time constraint in Problem (1) cannot be satisfied. In Proposition 4, we state the optimal expressions of n^* and q^* in K and the system parameters. The proof is given in Section 7 of the Online Appendix.

Proposition 4 *The following holds.*

1. *If $a_{S_2} \geq \frac{1}{\sqrt{2}}$ and $\lambda K \leq \frac{\sqrt{2} a_{S_2} \left((\sqrt{2}+1) a_{S_2} - \frac{1}{2} \right)}{\frac{1 + a_{S_1}}{2} + (\sqrt{2}+1) a_{S_2} - 1}$, then the optimal couple (n^*, q^*) is $n^* = \sqrt{2} a_{S_2} - 1$ and $q^* = \frac{\frac{\lambda K}{2}(1 + a_{S_1})}{\sqrt{2} a_{S_2} (a_{S_2}(\sqrt{2}+1) - 1/2) - \lambda K (a_{S_2}(\sqrt{2}+1) - 1)}$.*

2. *If $a_{S_2} < \frac{1}{\sqrt{2}}$ and $\lambda K \leq \frac{a_{S_2}(1 + a_{S_2})}{1 + a_{S_1} + a_{S_2}}$, then the optimal couple (n^*, q^*) is $n^* = 0$ and $q^* = \frac{(a_{S_1} + 1)\lambda K}{a_{S_2}(1 + a_{S_2} - \lambda K)}$.*

3. *In the remaining cases, the optimal couple (n^*, q^*) is*
   $n^* = \lambda K - a_{S_2} - \frac{1}{2} + \frac{1}{2}\sqrt{2 + (2\lambda K + 2a_{S_1} + 1)^2 - (1 + 2a_{S_1})^2 - (1 + 2a_{S_2})^2}$ and $q^* = 1$.

It is not possible to obtain the parameters p^* and t^* in closed-form. To solve Problem (1), we first fix a policy for the interlude and determine q^* and n^* according to Proposition 4. Next, by an exhaustive search, we obtain the best combination of parameters which solve Problem (1).

5.3 Main drivers for the decisions to use the interlude and the break

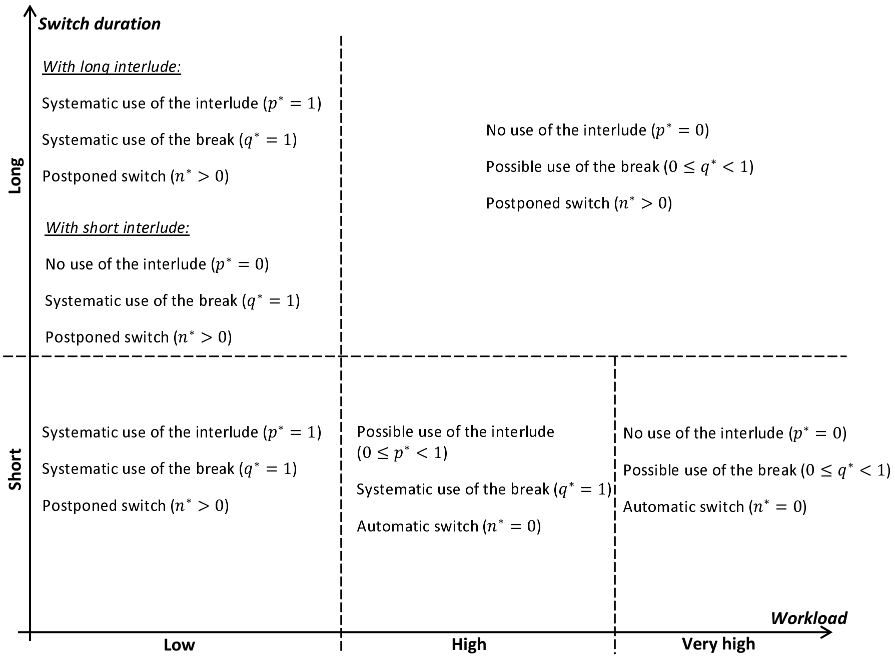
From numerical investigations, we observe that the switching times have a major impact on the optimal policy. For caseload= 1, from the expression of $E(T_I)$, the difference between $1/\mu_{S_1}$ and $1/\mu_I$ implies that if the expected time of the interlude is shorter than the expected switching time, then $E(T_I)$ decreases in p^*. So, in this case the interlude should not be used and $p^* = t^* = 0$. Second, the switching times also have a strong impact on the choice of n^* as shown in Proposition 4. In case of long switching times (first and last statement of the proposition), a high value for n^* may reduce the switches from HP to LP during the break. Finally in most cases, $t^* = 0$. It is almost never optimal to extend the interlude duration.

In Figure 4, we depict the possible cases generally observed in the case where $\mu_{S_1} = \mu_{S_2}$ (i.e., no distinction between the switches from HP to LP and from LP to HP). The characterization (long or short) of the switching time is relative to the service time, the characterization (high or low) of the workload is relative to the difficulty of satisfying the service level constraint and the characterization (long or short) of the interlude time is relative to the switching time. Considering an insurance or audit setting, one can envision a long switch if the server needs to change the software or system which they are using for HP and LP tasks, while a short switch may be seen in settings where the two tasks are not dramatically different and require a short mental adjustment. It is interesting to observe that in many cases very simple solutions should be implemented. For instance, with short switching times and high workload the combination $n^* = 0$ and $q^* = 1$ indicates that a strict priority should be given to HP. With even higher workload and short switching times, $p^* = 0$ means that it is a good strategy to reserve the server for HP during the interlude. On the contrary, in low workload situations, $p^* = 1$, suggesting to systematically treat LP during the interlude. More generally, we observe that many parameter combinations lead to a postponed switch policy, indicating that the server will continue working on LP despite some waiting HP tasks. Similarly, whenever p or q are equal to zero, the server is preferring to remain idle during the interlude or break, despite an infinite amount of LP work available.

Illustration. An illustrating instance is provided in Figures 5 and 6 for caseload= 1 and infinite caseload, respectively. For the comparison, the results for caseload= 1 are included in Figure 6 with the dotted curve. The results of Figure 5 are derived from the explicit expressions of the performance measures in Section 5.2 while those of Figure 6 are obtained from simulations. We propose here a simple procedure based on intuitive assumptions regarding the monotonicity properties of the performance measures. From simulations, we observe that increasing one of the parameters p^*, q^*, or n^* increases $E(W)$ and $E(T)$. This observation is intuitive as spending more time on LPs reduces the server's availability which in turn increases the wait. Based on this assumption, in what follows, we explain how we can restrict the number of simulations to a finite set of possible values for the control parameters.

As p^* and q^* may take an infinite set of values, we discretize their values. We introduce the parameter $M \in \mathbb{N}^+$, such that the values of p^* and q^* are restricted to $0, \frac{1}{M}, \frac{2}{M}, \cdots, \frac{M-1}{M}, 1$. Therefore, the couple (p^*, q^*) can take $(M+1)^2$ values. The parameter n^* can also take an infinite number of positive integer values. In the simulations, for each set of parameters p^* and q^*, we increase n^* by one until we reach $E(W) > \overline{w}$. For a given couple (p^*, q^*), the optimal value for n^* is the highest integer such that $E(W) \leq \overline{w}$. As $E(W)$ is unbounded in n^*, the optimal value for n^* can be determined after a finite number of simulation experiments. It may happen that for a given couple (p^*, q^*), we have $E(W) > \overline{w}$ for $n^* = 0$. In this case, the couple (p^*, q^*) should be excluded from the set of possible solutions as well as any couple with a higher value for p^* or q^*. This allows us to reduce the number of simulation experiments. Note also that with $p^* = q^* = 0\%$, n^* does not need to be specified.

Figures 5(a), 5(c), 6(a), 6(c) depict a case with a short switching time and Figures 5(b), 5(d), 6(b) and 6(d) a case with a long one, showing how switching times influence the blending decisions. Further, by comparing Figure 5(a) with Figure 5(c), Figure 6(a) with Figure 6(c), Figure 5(b) with Figure 5(d), or Figure 6(b) with Figure 6(d), we observe that $E(T)$ decreases with the variability of the service time. Moreover, the impact of the variability of the working phases is stronger when the interlude is not used (Figure 5(b) or 6(b) compared with Figure 5(d) or 6(d)). Two phenomena concur in this effect. First, if the interlude is long, then the working phases have a relatively small impact on the overall service length. Hence, the variability of these working phases has a small effect on the performance as shown in Figures 5(a), 5(c), 6(a) and 6(c). Second, if the interlude is long then it is also used for treating LP. This creates in-service switches which further reduce the relative importance of the working phases in the service process of an HP. Although the optimal policy derived in Section 4 differs from the one studied here for the use of the interlude, we observe the same drivers as in Section 4 for the switch from LP to HP (high congestion in Queue 1 or in Queue 2, short interludes, and short switching times).



With long interlude:
Systematic use of the interlude ($p^* = 1$)
Systematic use of the break ($q^* = 1$)
Postponed switch ($n^* > 0$)

With short interlude:
No use of the interlude ($p^* = 0$)
Systematic use of the break ($q^* = 1$)
Postponed switch ($n^* > 0$)

No use of the interlude ($p^* = 0$)
Possible use of the break ($0 \leq q^* < 1$)
Postponed switch ($n^* > 0$)

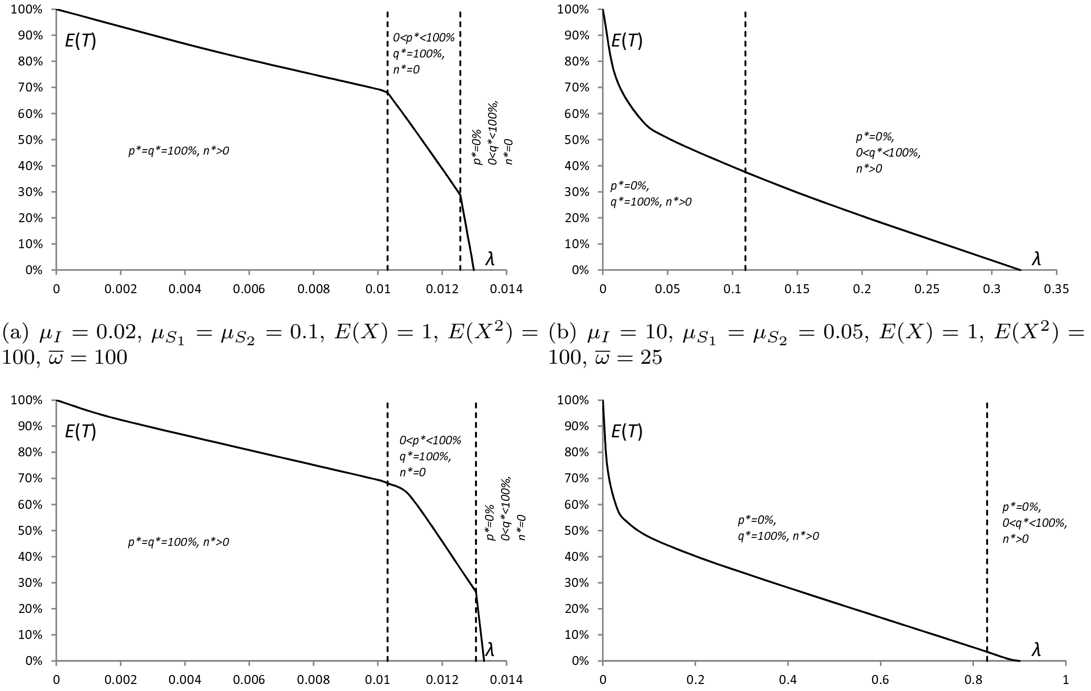
Systematic use of the interlude ($p^* = 1$)
Systematic use of the break ($q^* = 1$)
Postponed switch ($n^* > 0$)

Possible use of the interlude ($0 \leq p^* < 1$)
Systematic use of the break ($q^* = 1$)
Automatic switch ($n^* = 0$)

No use of the interlude ($p^* = 0$)
Possible use of the break ($0 \leq q^* < 1$)
Automatic switch ($n^* = 0$)

Figure 4: Summary of numerical study

The comparison between Figures 5 and 6 shows that our conclusions are consistent in the caseload. However, Figure 6 confirms the observation of Section 4.3.2, showing that the caseload has a strong impact on the performance measures. As mentioned in Section 4.3, the situation with infinite caseload leads to a better solution Problem 1 than the case with caseload= 1. However, the improvement differs strongly as functions of the environmental conditions. The comparison between Figures 6(a) and 6(c) with Figures 6(b) and 6(d) shows that a limited caseload has the most detrimental effect when the interlude is long. By restricting the caseload, the server is forced to idle while she/he could treat some HPs present in Queue 1. These unproductive states occur more frequently when the interlude is long. With infinite caseload, the role of the working phases' variability is stronger. This can be seen by the more important difference between Figures 6(a) and 6(c) as compared to the difference between Figures 5(a) and 5(c). For these figures, the interlude duration is significantly longer than the duration of the working phases. The detrimental effect of the long interlude is reduced with infinite caseload as the server is able to continue working on HPs. Therefore, by reducing the effect of the interlude duration, the working phases' variability becomes more influential.



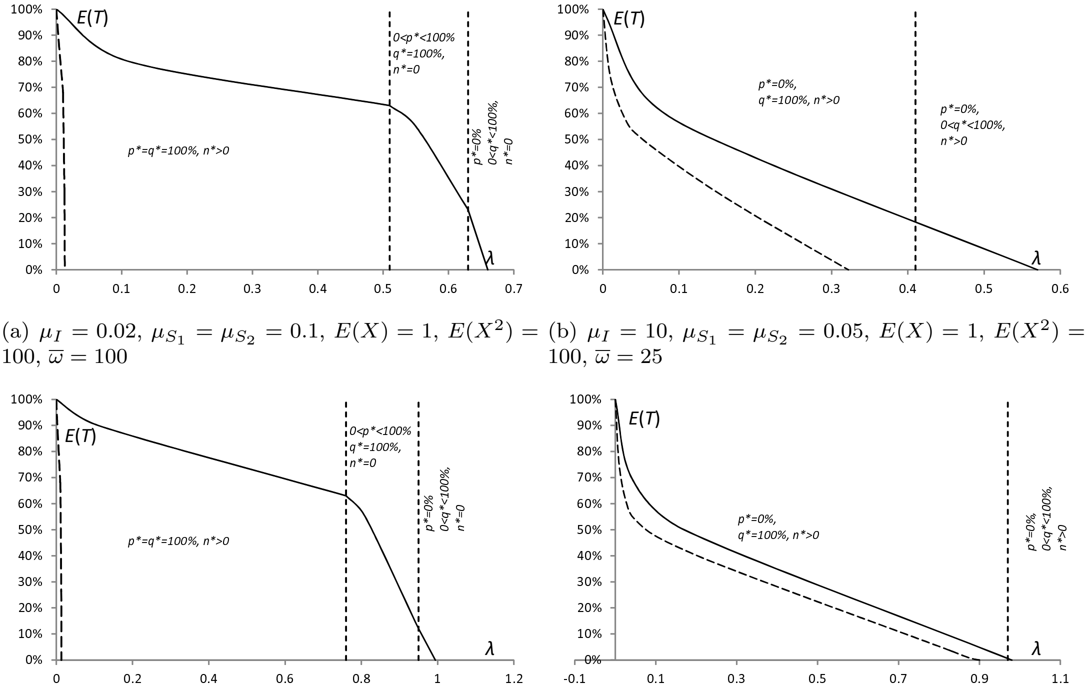
(a) $\mu_I = 0.02$, $\mu_{S_1} = \mu_{S_2} = 0.1$, $E(X) = 1$, $E(X^2) = 100$, $\overline{\omega} = 100$

(b) $\mu_I = 10$, $\mu_{S_1} = \mu_{S_2} = 0.05$, $E(X) = 1$, $E(X^2) = 100$, $\overline{\omega} = 25$

(c) $\mu_I = 0.02$, $\mu_{S_1} = \mu_{S_2} = 0.1$, $E(X) = 1$, $E(X^2) = 1$, $\overline{\omega} = 100$

(d) $\mu_I = 10$, $\mu_{S_1} = \mu_{S_2} = 0.05$, $E(X) = 1$, $E(X^2) = 1$, $\overline{\omega} = 25$

Figure 5: $E(T)$ as a function of λ for short (a),(c) and long (b),(d) switching times for caseload= 1



(a) $\mu_I = 0.02$, $\mu_{S_1} = \mu_{S_2} = 0.1$, $E(X) = 1$, $E(X^2) = 100$, $\overline{\omega} = 100$

(b) $\mu_I = 10$, $\mu_{S_1} = \mu_{S_2} = 0.05$, $E(X) = 1$, $E(X^2) = 100$, $\overline{\omega} = 25$

(c) $\mu_I = 0.02$, $\mu_{S_1} = \mu_{S_2} = 0.1$, $E(X) = 1$, $E(X^2) = 1$, $\overline{\omega} = 100$

(d) $\mu_I = 10$, $\mu_{S_1} = \mu_{S_2} = 0.05$, $E(X) = 1$, $E(X^2) = 1$, $\overline{\omega} = 25$

Figure 6: $E(T)$ as a function of λ for short (a),(c) and long (b),(d) switching times with $M = 10$ for infinite caseload

5.4 Managerial questions

We end this section by focusing on specific questions related to our optimization problem. The analysis in the section is made with the performance measures obtained for caseload= 1 but are checked to be valid with higher caseload. We first explore the question of when to use the interlude and when to prefer the break for LP work. To this end, we compare p^* and q^*. We observe in Figure 5(a) that as the workload increases, we should first decrease p^* and then q^*. This is consistent with the results of Section 4 (see Table 2) where in most cases $u_B \geq u_{I_1}$. The second observation is $t^* = 0$. Apparently, extending the interlude duration is not a good strategy. The last observation is that the switch from HP to LP has a worse impact than the switch from LP to HP. Finally, when more than one interlude is involved should we either spread the work on LPs over different interludes or should we instead concentrate this work on fewer ones? Are these observations generalizable? Theorem 1 allows us to explore these questions analytically and to understand the intuition behind preferences for use of the interlude compared to the break.

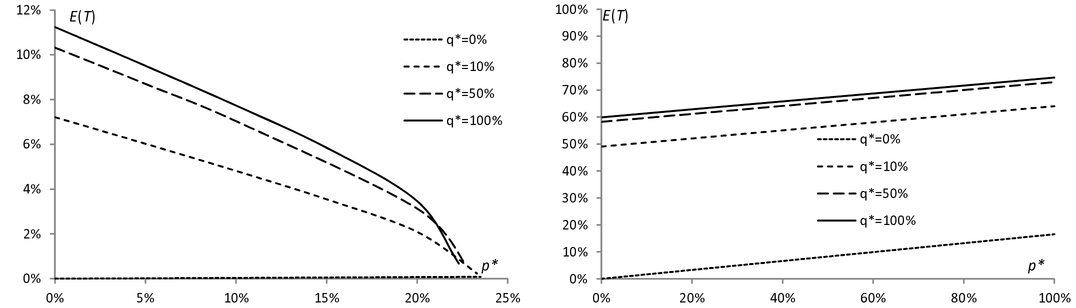
5.4.1 Which duration to use more frequently for back-office work: interlude or break?

We consider a situation where $t^* = 0$ (i.e., no possibility of increasing the interlude duration) and explore whether the interlude should be used for LP work (i.e., switch from HP to LP). We first consider Problem (1) without the waiting time constraint. We compare an increase in p^* with an increase in q^* in order to get an insight on the eventual preference for a more frequent use of the interlude or the break for LP work. After some algebra, we obtain

$$\begin{aligned}&\frac{\partial E(T)}{\partial p^*} - \frac{\partial E(T)}{\partial q^*}\\ &= \frac{-(a_{S_1}+1)(n^*+1)(1-\lambda E(R))}{(1+n^*q^*+a_{S_1}+q^*a_{S_2})^2} + a_I \frac{n^*q^*(a_I^2 - a_I a_{S_2} - a_{S_1}a_{S_2}) + q^* a_{S_2}(a_I^2 - a_I - a_{S_1}) + a_I^2(1+a_{S_1})}{(a_I+a_{S_1})^2(1+n^*q^*+a_{S_1}+q^*a_{S_2})}.\end{aligned}$$

If this difference is positive then increasing the use of the interlude is preferred and vice-versa. To make this difference positive, one should either increase a_I, $E(R)$ or decrease a_{S_1} or a_{S_2}. In other words, for treating LP, the interlude is preferred to the break when the service duration is long, the interlude time is long, or the switch times are short.

The service level constraint is imposed next. Figure 7, depicts a case with a high workload in (Figure 7(a)) and a case with a low workload in (Figure 7(b)). It illustrates that, as the workload increases, the impact of p^* on $E(T)$ varies from a positive effect to a negative one. The negative effect of p^* under a high workload situation can be understood through the following rewriting of the optimization problem at saturation of the service level constraint



(a) Case 1 ($\lambda = 0.43$, $\mu_I = 1$, $\mu_{S_1} = \mu_{S_2} = 0.1$, $E(X) = 1$, $E(X^2) = 2$, $\overline{w} = 50$)

(b) Case 2 ($\lambda = 0.2$, $\mu_I = 1$, $\mu_{S_1} = \mu_{S_2} = 10$, $E(X) = 1$, $E(X^2) = 100$, $\overline{w} = 100$)

Figure 7: $E(T)$ as a function of p^* and q^*

(i.e., $E(W)=\overline{w}$):

$$\begin{cases} \text{Maximize } E(T)=E(T_I)+(1-\lambda E(R))\left[\overline{w}-E(W_2)-\frac{\lambda E(R)^2(1+cv_R^2)}{2(1-\lambda E(R))}+\frac{1}{\mu_{S_2}}\frac{1+n^*+a_{S_2}}{n^*+\frac{1}{q^*}+\frac{1}{q^*}\left(a_{S_1}+q^*a_{S_2}\right)}\right] \\ \text{subject to } E(W)=\overline{w}. \end{cases} \tag{9}$$

In the expression of $E(T)$, the parameter p^* appears in $E(T_I)$ with a positive effect, and in the terms $E(W_2)$ and $\frac{\lambda E(R)^2(1+cv_R^2)}{2(1-\lambda E(R))}$ with a negative one. As the workload increases, the impact of the terms related to the wait increase in the expression of $E(T)$. So the negative effect of an increase in p^* can overcome its positive effect on $E(T_I)$. Thus, in low workload situations, the interlude is a real opportunity to increase the time spent on LP, but under high workload situations it might not be preferred in comparison with spending more time on LP during breaks.

5.4.2 Which duration to extend for more back-office work: interlude or break?

We compare two strategies: increase the service time (more time for LP during the interlude) or increase the busy period (more time for LP during the break) in order to treat more LP? Consider a situation where it is optimal to choose $p^*=q^*=1$. Such a situation may occur with a low arrival rate or a non-restrictive service level constraint. What is more effective: increasing the LP working time of the server during the break with parameter n^* or during the interlude with parameter t^*? Since the nature of n^* and t^* is different, we consider the variable λt^* instead of t^*, enabling a comparison with n^*. First, observe that

$$\begin{aligned} \frac{\partial E(T)}{\partial n^*} &= \frac{(a_{S_1}+a_{S_2})(1-\lambda E(R))}{(n^*+1+a_{S_1}+a_{S_2})^2}, \text{ and,} \\ \frac{\partial E(T)}{\partial \lambda t^*} &= \frac{a_I(a_{S_1}+a_{S_2})}{(a_{S_1}+a_I)(n^*+1+a_{S_1}+a_{S_2})} - \frac{a_{S_1}a_I}{(a_{S_1}+a_I)^2}e^{-\lambda t^*/a_{S_1}}. \end{aligned}$$

Clearly $\frac{\partial E(T)}{\partial n^*}>0$. When one disregards the waiting time constraint, increasing the time spent on LP between HP therefore cannot be counterproductive. The same cannot be said for the time spent on LP during the interlude; $\frac{\partial E(T)}{\partial \lambda t^*}<0$ is equivalent to

$$(a_I+a_{S_1})(a_{S_1}+a_{S_2}) < a_{S_1}(n^*+1+a_{S_1}+a_{S_2})e^{-\lambda t^*/a_S}. \tag{10}$$

Starting from a situation where $t^*=0$, Inequality (10) becomes $\frac{a_I(a_{S_1}+a_{S_2})}{a_{S_1}}<n^*+1$. Since $a_I<1$ due to stability reasons then Inequality (10) is often met and extending the duration of the service time would not make sense in most cases (except if the workload is very low, the interlude duration is very short or the threshold on the queue length is very low). The manager is then in general more tempted to increase n^* rather than t^*. This explains why $t^*=0$ is optimal in most cases. Yet, counterexamples can be found (see Section 8 of the Online Appendix).

5.4.3 Which is the worst switch?

We question here the impact of the switching times by differentiating the switch from HP to LP to the switch from LP to HP. In Table 3, using the results of Theorem 1 and Proposition 4, we give the optimal control parameters and the optimal value for $E(T)$ for different values of the system parameters. We choose $\lambda=0.2, 0.35$, and, 0.37 in order to reflect low, moderate, and, high workload situations. We also assume that $\frac{1}{\mu_{S_1}}+\frac{1}{\mu_{S_2}}=1$, such that, the cumulative expected duration of the two switches is kept constant.

In all cases, we observe that the duration of the switch from LP to HP (i.e. decrease of μ_{S_2}) has a worst effect than the switch from HP to LP (i.e. decrease of μ_{S_1}). This confirms the observation of Table 2. When the server is in switch from HP to LP, an arriving HP in Queue 1 or in Queue 2 can directly start service. Therefore, HP jobs seems to only be delayed by the switch from LP to HP. This explains why, when $\mu_{S_2}=\infty$, the server should never

Table 3: Performance evaluation ($\mu_I = 1$, $E(X) = 1$, $E(X^2) = 10$, $\frac{1}{\mu_{S_1}} + \frac{1}{\mu_{S_2}} = 1$, $\overline{w} = 10$)

	μ_{S_1}	μ_{S_2}	n^*	p^*	q^*	$E(T)$
$\lambda = 0.2$	1	∞	3.21	100%	100%	62.28%
	1.33	4	2.97	100%	100%	60.93%
	2	2	2.64	100%	100%	59.44%
	4	1.33	2.15	100%	100%	57.94%
	∞	1	1.31	100%	100%	56.81%
$\lambda = 0.35$	1	∞	1.47	100%	100%	35.03%
	1.33	4	0.00	77%	100%	28.13%
	2	2	1.12	0%	100%	25.75%
	4	1.33	0.92	0%	100%	25.37%
	∞	1	0.70	0%	100%	24.88%
$\lambda = 0.37$	1	∞	0.04	100%	100%	28.42%
	1.33	4	0.00	0%	18.32%	3.62%
	2	2	0.00	0%	7.60%	1.65%
	4	1.33	0.00	0%	4.40%	1.04%
	∞	1	0.00	0%	2.90%	0.75%

idle (i.e., $p^* = q^* = 100\%$). We also observe that the sensitivity to the switching rates increases with the workload. In high workload situations, more HPs are present in the system. Therefore, the server can more frequently switch from HP to LP during the interlude or the break. Hence, this increases the sensitivity to the switch parameters.

In fact, the switch from HP to LP even has a beneficial effect on the wait of HPs. The reason is that with a long switch from HP to LP, the server has less chance to start serving LPs before an HP arrives in Queue 1 or in Queue 2. Since HPs can preempt a switch from HP to LP, having a long switch means having a long period of time during which an arriving HP can be served without being delayed. This can be proven using the results of Theorem 1. Consider the expression of the expected wait in Equation (6) with $t^* = 0$. The first part, which represents $E(W_2)$, can be rewritten as $E(W_2) = \frac{p^*}{\lambda}\frac{a_I a_{S_2}}{a_I + a_{S_1}}$. Since a_{S_1} is only in the denominator of this expression, $E(W_2)$ is decreasing in a_{S_1} which means that the wait in Queue 2 decreases with the length of the switch from HP to LP. For the same reason, the third and the fourth parts of the expression of $E(W)$ are also decreasing in a_{S_1}. The second part of $E(W)$, $\frac{\lambda E(R)^2(1+cv_R^2)}{2(1-\lambda E(R))}$, can be rewritten as $\frac{1}{\lambda}\frac{\lambda^2\frac{E(X^2)}{2}+\lambda E(X)\left(a_I+p^*\frac{a_I a_{S_2}}{a_I+a_{S_1}}\right)+a_I\left(a_I+p^*\frac{a_{S_2}(a_{S_2}+a_I)}{a_I+a_{S_1}}\right)}{1-\lambda E(X)-a_I-p^*\frac{a_I a_{S_2}}{a_I+a_{S_1}}}$. This leads to

$$\frac{\partial\left(\frac{\lambda E(R)^2(1+cv_R^2)}{2(1-\lambda E(R))}\right)}{\partial a_{S_1}} = -\frac{1}{\lambda}\frac{p^* a_{S_2} a_I\left(\frac{\lambda^2 E(X^2)}{2} + (1 - a_I - \lambda E(X))(\lambda E(X) + a_{S_2}) + a_I\right)}{\left((a_I + a_{S_1})(1 - \lambda E(X) - a_I) - p^* a_I a_{S_2}\right)^2}.$$

This expression is negative as $1 - \lambda E(X) - a_I > 0$ for stability reasons. Therefore, we have proven that $E(W)$ decreases with the length of the switch from HP to LP. The expected time spent on LPs during the interlude, $E(T_I) = \frac{a_I^3}{\left(a_I+a_{S_1}\right)^2}$, is clearly decreasing in a_{S_1}. The expected time spent on LPs during the break, $E(T_B) = \frac{(n^*+1)\left(1-\lambda E(X)-a_I-p^*\frac{a_I a_{S_2}}{a_I+a_{S_1}}\right)}{n^*+\frac{1}{q^*}+\frac{1}{q^*}\left(a_{S_1}+q^* a_{S_2}\right)}$ is not always decreasing in a_{S_1}. After deriving the derivative of $E(T_B)$ we show that $E(T_B)$ is increasing in a_{S_1} if and only if $p^* a_{S_2} a_I(1 + n^* q^* + 2a_{S_1} + a_I + q^* a_{S_2}) > (a_I + a_{S_1})^2\,(1 - \lambda E(X) - a_I)$. This condition is complex to analyze but it qualitatively means that the length of the switch from HP to LP can be beneficial for the time spent on LPs during the break when the interlude and the break are highly used for LPs (i.e. high value of p^*, q^* and n^*), and when the switch from LP to HP is long. In such cases, having a long switch from HP to LP allows to have shorter interlude duration by not having enough time to initiate LPs during the interlude. This reduces the expected service time of HPs, $E(R)$, and allows the server to have less HPs present in the system. Consequently, the breaks can be longer and the time spent on LPs during the breaks can also be longer.

5.4.4 How to deal with more than one interlude?

An important question is how the server should organize her work when multiple interludes are present in the service encounter? Each time LP work is attempted during an interlude, the server will incur the switching time twice. As shown earlier, this affects system performance both in terms of expected duration and variability of the service process. Intuitively, this would point to a policy of concentrating LP work on fewer interludes. Yet a more myopic policy that considers each interlude independently from another and attempts LP work whenever an interlude is long enough relative to the switching time might also be efficient and may lead to a choice of using more of the interludes for LP work. We explore the performance implications of these two options. A policy of concentration further requires a choice of which interludes to use for LP work. In an environment with a constraint on the expected waiting time of HP, it makes sense intuitively to delay the LP work to later interludes, at which point there are a small number of remaining working phases for HP. To provide rules of thumb for managers addressing these questions, we consider a situation with only two interludes with interlude rates, μ_{I_1} and μ_{I_2}. We optimize the use of the interludes under Policy $\mathcal{P}$. Under this policy, the position of an interlude during service does not influence the optimal decision since the use of the interlude is non state-dependent. In Section 9 of the Online Appendix, to further explore the role of the position of an interlude, using an MDP approach similar to the one developed in Section 4, we characterize more complex properties that the optimal policy for the use of the interludes may have. We show that the optimal policies provide similar insights to those derived under Policy $\mathcal{P}$ in this subsection.

Without loss of generality for Policy $\mathcal{P}$, we assume here that $\mu_{I_2} > \mu_{I_1}$. So, the second interlude is shorter than the first one. Clearly, there is a preference for using the first interlude. We assume that it is not optimal to extend the interlude duration ($t^* = 0$). The server has the choice between two policies. Either using only the first interlude with control parameter p_a^* (concentration on one interlude) or a bit of both interludes with control parameters p_b^* and p_c^* (spreading of the work between the two interludes). If the two policies achieve the same expected time on LP during service, using the expression of $E(T_I)$, we may write

$$p_a^* = p_b^* + p_c^* \frac{\mu_{I_1}(\mu_S + \mu_{I_1})}{\mu_{I_2}(\mu_S + \mu_{I_2})}.$$

We now compare the expected waiting time in Queue 2 under the two policies by computing the difference $E(W_{2,A}) - E(W_{2,B})$ where $E(W_{2,A})$ is the expected supplementary waiting time when the work is concentrated on one interlude and $E(W_{2,B})$ is the expected supplementary waiting time when the work is spread on the two interludes. We have $E(W_{2,A}) - E(W_{2,B}) = \frac{p_c^*}{\mu_S + \mu_{I_2}} \left(\frac{\mu_{I_1}}{\mu_{I_2}} - 1 \right) < 0$. Thus the supplementary waiting time is higher when the work is spread on two interludes. Similarly, we show that the expected service time is also longer when the work is spread on two interludes. We now consider the difference $E(S_A^2) - E(S_B^2)$ which represents the difference between the second order moments of the interlude times in the two situations. We have $E(S_A^2) - E(S_B^2) = \frac{2p_c^*}{\mu_S(\mu_S + \mu_{I_2})} \left(\frac{\mu_{I_1}}{\mu_{I_2}} - 1 \right) < 0$. Again concentrating the work on one interlude has a better effect in terms of reducing variability, than spreading it over more than one interlude.

The intuitive conclusion here is that if a given quantity of time can be dedicated to LP during the interludes then it is better to concentrate the work on a limited number of interludes without extending any of these. The chosen interludes should be the longest. This avoids having too many switching times.

6 Concluding remarks

A lot of service front-office work has intervals of random length during and between the treatment of different customers, when the server is not needed. In this paper, we have explored the use of these interludes and breaks to treat back-office tasks. The latter tasks are different in nature from the front-office tasks, and imply switching times as the server alternates between these in their work. Our analysis shows that these switching times are important to model and have a significant effect on the structure of optimal choices regarding when to treat back-office tasks. The server may prefer to make customers wait while continuing to work on back-office due to long switching times. Alternatively, the server may prefer to remain idle despite an infinite amount of back-office tasks to process, due to the long switching times otherwise incurred. Optimal policies for use of the interludes and breaks are state-dependent and are controlled by several thresholds. Due to its dependence on random arrivals by customers both pre-process and in-process, the control of the interlude blending decision is more difficult relative to the one during the break. Furthermore the two interact. In the absence of automation, the optimal policies may be hard to implement by a server. For such cases, simpler non state-dependent policies are proposed. Both type of policies lead to similar insights regarding the drivers to use the interlude times for back-office blending: a low or moderate workload, long interlude times, low sensitivity of customers to waiting (as captured by the waiting service level constraint). These findings suggest that managers need to understand specific process features well, before determining ideal blending approaches. It is shown that blending of front-back office work is not just meaningful during breaks between customers but also in interludes within service, despite the switching times that need to be incurred in doing so. In the presence of multiple interludes, the rule of thumb developed is to make use of the longer and later interludes in a service encounter.

An alternative to the front-back office work blending to make use of server idle times, is working on several customers (front-office tasks) simultaneously as in case-manager systems. Studied in different contexts, earlier work has shown that limiting the caseload in such systems may be preferred due to quality or efficiency related reasons. The analysis herein shows the effect of an increasing caseload on in-process waits. When coupled with the observation of diminishing benefits to increasing the caseload, this suggests to limit caseloads in front-back office work blending settings to low numbers.

In future research, it would be interesting but challenging to extend the analysis to cases with more than two tasks done by alternation. Different assumptions regarding switching times may be needed to analyze settings where the multitasking is done between different types of customer tasks that are all front-office (requiring customer interaction) instead of a front-office versus back-office task. Another challenging extension is to investigate other objectives for front-office tasks like wait percentiles or the average excess wait. Choosing other objectives would lead one to reconsider fundamental assumptions like the priority for front-office tasks at their service completion. New technologies allow service systems to keep track of past performance. So, we could also envision to rethink the priority rules from the second queue based on the past performance in the first queue. Going one step beyond the current framework where the length of the interlude is modeled as an exogenous parameter (exponential with given rate), one can consider this parameter to be the outcome of a coproductive service design (Roels, 2014). This is relevant for settings where the interlude is a self-service task. Endogenous choice of the interlude duration would correspond to the determination of the extent of self-service in a service encounter. This would enable creating settings where the interlude length is controlled via service design.

References

Altman, E. (1999). *Constrained Markov decision processes*, volume 7. CRC Press.

Andradóttir, S., Ayhan, H., and Down, D. (2001). Server assignment policies for maximizing the steady-state throughput of finite queueing systems. *Management Science*, 47(10):1421–1439.

Armony, M. and Maglaras, C. (2004). Contact centers with a call-back option and real-time delay information. *Operations Research*, 52(4):527–545.

Bhulai, S. and Koole, G. (2003). A queueing model for call blending in call centers. *IEEE Transactions on Automatic Control*, 48(8):1434–1438.

Brandt, A. and Brandt, M. (1999). On a two-queue priority system with impatience and its application to a call center. *Methodology and Computing in Applied Probability*, 1(2):191–210.

Campello, F., Ingolfsson, A., and Shumsky, R. (2017). Queueing models of case managers. *Management Science*, 63(3):882–900.

Charron, S. and Koechlin, E. (2010). Divided representation of concurrent goals in the huma frontal lobes. *Science*, 328(5976):360–363.

Cui, L. and Tezcan, T. (2016). Approximations for chat service systems using many-server diffusion limits. *Mathematics of Operations Research*, 41(3):775–807.

Dai, J., Hasenbein, J., and Vate, J. (2004). Stability and instability of a two-station queueing network. *The Annals of Applied Probability*, 14(1):326–377.

Dai, J. and Weiss, G. (1996). Stability and instability of fluid models for reentrant lines. *Mathematics of Operations Research*, 21(1):115–134.

Deslauriers, A., L'Ecuyer, P., Pichitlamken, J., Ingolfsson, A., and Avramidis, A. (2007). Markov chain models of a telephone call center with call blending. *Computers & Operations Research*, 34(6):1616–1645.

Dobson, G., Lee, H., Sainathan, A., and Tilson, V. (2012). A queueing model to evaluate the impact of patient batching on throughput and flow time in a medical teaching facility. *Manufacturing & Service Operations Management*, 14(4):584–599.

Dobson, G., Tezcan, T., and Tilson, V. (2013). Optimal workflow decisions for investigators in systems with interruptions. *Management Science*, 59(5):1125–1141.

Duenyas, I., Gupta, D., and Olsen, T. (1998). Control of a single-server tandem queueing system with setups. *Operations Research*, 46(2):218–230.

Dux, P., Tombu, M., Harrison, S., Rogers, B., Tong, F., and Marois, R. (2009). Training improves multitasking performance by increasing the speed of information precessing in human prefontal cortex. *Neuron*, 63(1):127–138.

Gans, N. and Zhou, Y. (2003). A call-routing problem with service-level constraints. *Operations Research*, 51(2):255–271.

Gladstones, W., Regan, M., and Lee, R. (1989). Division of attention: The single-channel hypothesis revisited. *Quarterly Journal of Experimental Psychology Section A: Human Experimental Psychology*, 41(1):1–17.

Gromoll, H. C., Robert, P., and Zwart, B. (2008). Fluid limits for processor-sharing queues with impatience. *Mathematics of Operations Research*, 33(2):375–402.

Gurvich, I. and Van Mieghem, J. (2017). Collaboration and multitasking in networks: Prioritization and achievable capacity. *Management Science*, 64(5):2390–2406.

Gurvich, I. and Van Mieghem, J. A. (2014). Collaboration and multitasking in networks: Architectures, bottlenecks, and capacity. *Manufacturing & Service Operations Management*, 17(1):16–33.

Hasenbein, J. (1997). Necessary conditions for global stability of multiclass queueing networks. *Operations research letters*, 21(2):87–94.

Iravani, S., Posner, M., and Buzacott, J. (1997). A two-stage tandem queue attended by a moving server with holding and switching costs. *Queueing Systems*, 26(3-4):203–228.

Johri, P. and Kateiiakis, M. (1988). Scheduling service in tandem queues attended by a single server. *Stochastic Analysis and Applications*, 6(3):279–288.

KC, D. (2013). Does multitasking improve performance? Evidence from the emergency department. *Manufacturing & Service Operations Management*, 16(2):168–183.

Keblis, M. and Chen, M. (2006). Improving customer service operations at amazon.com. *Interfaces*, 36(5):433–445.

Koole, G. and Righter, R. (1998). Optimal control of tandem reentrant queues. *Queueing Systems*, 28(4):337–347.

Legros, B. and Jouini, O. (2019). On the scheduling of operations in a chat contact center. *European Journal of Operational Research*, 274(1):303–316.

Legros, B., Jouini, O., and Koole, G. (2015). Adaptive threshold policies for multi-channel call centers. *IIE Transactions*, 47(4):414–430.

Legros, B., Jouini, O., and Koole, G. (2016). Optimal scheduling in call centers with a callback option. *Performance Evaluation*, 95:1–40.

Lohr, S. (2007). Slow down, brave multitasker, and don't read this in traffic. *The New York Times*, 25.

Neuts, M. (1981). *Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach.* Johns Hopkins University Press, Mineola.

Pang, G. and Perry, O. (2014). A logarithmic safety staffing rule for contact centers with call blending. *Management Science*, 61(1):73–91.

Pichitlamken, J., Deslauriers, A., L'Ecuyer, P., and Avramidis, A. (2003). Modeling and simulation of a telephone call center. *Proceedings of the 37th Conference on Winter Simulation, New Orleans, LA*, pages 1805–1812.

Puterman, M. (1994). *Markov Decision Processes.* John Wiley and Sons.

Roels, G. (2014). Optimal design of coproductive services: Interaction and work allocation. *Manufacturing & Service Operations Management*, 16(4):578–594.

Rosen, C. (2008). The myth of multitasking. *The New Atlantis*, 20(Spring):105–110.

Shae, Z., Garg, D., Bhose, R., Mukherjee, R., Guven, S., and Pingali, G. (2007). Efficient internet chat services for help desk agents. In *Services Computing, 2007. SCC 2007. IEEE International Conference on*, pages 589–596.

Srinivasan, M. and Gupta, D. (1996). When should a roving server be patient? *Management Science*, 42(3):437–451.

Stolyar, A. L. (2004). Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *Annals of Applied Probability*, 14:1–53.

Tezcan, T. and Zhang, J. (2014). Routing and staffing in customer service chat systems with impatient customers. *Operations Research*, 62(4):943–956.

Yom-Tov, G. and Mandelbaum, A. (2014). Erlang-R: A time-varying queue with reentrant customers, in support of healthcare staffing. *Manufacturing & Service Operations Management*, 16(2):283–299.