

# Scheduling of multi-class multi-server queueing systems with abandonments \*

U. Ayesta<sup>b,c,e,f</sup>, P. Jacko<sup>d</sup>, V. Novak<sup>a</sup>

<sup>a</sup>CERGE-EI, a joint workplace of Charles University in Prague and  
the Economics Institute of the Czech Academy of Sciences,  
Politických veznu 7, Prague 1, 111 21, Czech Republic

<sup>b</sup>CNRS, LAAS, 7 avenue du colonel Roche, 31400 Toulouse, France

<sup>c</sup>IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain

<sup>d</sup>Lancaster University, Department of Management Science, Lancaster, LA1 4YX, UK

<sup>e</sup>UPV/EHU, Univ. of the Basque Country, 20018 Donostia, Spain

<sup>f</sup>Univ. de Toulouse, INP, INSA, LAAS, 31400 Toulouse, France

This is the author final version of the article published in Journal of Scheduling, online  
on 11 November 2015, DOI 10.1007/s10951-015-0456-7.

## Abstract

Many real-world situations involve queueing systems in which customers may abandon if service does not start sufficiently quickly. We study a comprehensive model of multi-class queue scheduling accounting for customer abandonment, with the objective of minimizing the total discounted or time-average sum of linear waiting costs, completion rewards and abandonment penalties of customers in the system. We assume the service times and abandoning times are exponentially distributed. We solve analytically the case in which there is one server and there are one or two customers in the system and obtain an optimal policy. For the general case, we use the framework of restless bandits to analytically design a novel simple index rule with a natural interpretation. We show that the proposed rule achieves near-optimal or asymptotically-optimal performance both in single- and multi-server cases, both in overload and underload regimes, and both in idling and non-idling systems.

## 1 Introduction

Abandonment (aka renegeing) is a ubiquitous phenomenon in a multitude of systems. It may happen due to customers' impatience, mobility, perishability, obsolescence, or other reasons. For instance, customers can abandon after waiting for too long in a queue, Internet users may give up a transfer if the connection is slow, or external factors may cause customers not to be amenable to service anymore. Abandonment is especially important in call centers ([Gans et al. 2003](#), [Aksin et al. 2007](#)) and health care ([Argon et al. 2010](#)). Other application areas in which abandonment may have a significant impact are real-time systems where data received after certain hard deadline is useless ([Buttazzo 2011](#)), or inventory systems with perishable goods ([Graves 1984](#)).

---

\*An early version of this paper appeared in [Ayesta et al. \(2011\)](#).

Customer abandonment has a very negative impact on performance both from the customer's and the system's perspective. A customer who decides to abandon perceives that the system is poorly managed or not well dimensioned, which may lead to switching to a competitor. In health care, abandonment may be used to model irreversible deterioration of patients' health. From the system's point of view the abandonment might imply that resources have been wasted by allocating them to a customer that decided to abandon anyway. For this reason it is important to design efficient scheduling policies in the presence of abandonment.

Mathematically, abandonment can be studied by queueing models, but as a consequence of the complexity, the problem of how to schedule impatient customers is not completely understood. For instance, very little is known about how to schedule customers in order to optimize a performance criteria that may depend on the number of customers waiting and on the number of customers that abandon the queue. This is an important question, for which no general solution is known, and it is likely to be computationally infeasible. The main focus of the present paper is to design a simple implementable scheduling rule for multi-class systems that shows a nearly-optimal performance.

The literature on models that incorporate customer abandonment is rich, and there has been a surge in recent years motivated by their application in health-care, call-centers and the Internet. As an illustration of the latter we can mention the recent Special Issue in Queueing Systems devoted to queueing systems with abandonments [Hasenbein and Perry \(Eds\)](#). An important stream of works investigate the performance of single-server systems in the presence of abandonment (see for example [Iravani and Balcioglu 2008](#), [Baccelli et al. 1984](#), [Brandt and Brandt 2004](#), [Brill and Posner 1977](#), [Ata and Tongarlak 2013](#)), and there is also a significant body of literature dealing with the multi-server case (see for example [Boxma and de Waal 1994](#), [Boots and Tijms 1999](#), [Whitt 2004](#)). We also refer to [Dai and He \(2012\)](#) for a recent survey on multi-server systems with abandoning customers. Abandonment systems have also been looked at from a game-theoretic perspective (see [Hassin and Haviv 2003](#), Chapter 5, and references therein). More relevant to our work are the papers that deal with the control of systems in the presence of abandonment (see, for example, [Harrison and Zeevi 2004](#), [Glazebrook et al. 2004](#), [Argon et al. 2010](#), [Down et al. 2011](#), [Jouini et al. 2010](#), [Atar et al. 2010](#)). The most common objectives are either service completion reward maximization or waiting cost and abandonment penalty minimization. In [Glazebrook et al. \(2004\)](#) the authors consider three different models for service completion reward maximization, and derive heuristic rules using a variety of techniques. [Argon et al. \(2010\)](#) study how to schedule optimally in a patient triage problem. [Down et al. \(2011\)](#) deals with a two-class system and derives sufficient conditions for a classic priority rule to be optimal. In [Jouini et al. \(2010\)](#) the authors derive scheduling policies for call centers with two classes of customers.

## 1.1 Problem Description

In this paper we aim at solving the problem of multi-class multi-server customer scheduling for a system in which we allow for abandonment. The objective is to maximize the total discounted or time-average revenue from customers in the system, where the revenue is defined as the sum of service completion rewards minus waiting costs and abandonment penalties. In contrast to [Atar et al. \(2010\)](#) and most of the previous literature, *the customers in service are also charged a waiting cost*, which as we will see, will have significant consequences.

Let  $M$  denote the number of servers, or machines. There are  $K$  classes of customers, which arrive to the system according to a Poisson arrival process with mean  $\lambda_k$  for class  $k \in \mathcal{K} := \{1, 2, \dots, K\}$ . Every customer incurs a waiting cost while in the system, an abandonment penalty in case of abandonment, and yields a service completion reward in case of service completion.

Service completion times and abandonment times are exponentially distributed. Thus, the class- $k$  customer's behavior is completely characterized by the mean service time  $1/\mu_k$  (with service rate  $\mu_k > 0$  while served), the mean abandonment time  $1/\theta_k$  (with abandonment rate  $\theta_k > 0$  while not served), the waiting cost rate  $c_k$  (possibly negative), the abandonment penalty  $d_k$  (possibly negative), and the service completion reward  $r_k$  (possibly negative). Future costs, penalties and rewards are discounted with exponential discount rate  $\alpha \geq 0$ .

We emphasize that we assume that the customers in service cannot abandon, though our approach could easily be adapted to the other possibility. We also assume that a customer can only receive service from one server at a time. Customers are assumed independent of each other, i.e., we assume that at any moment and independently of everything else, every customer (except for those in service, if any) may abandon the system, while a customer in service leaves the system if her service is completed.

It is further permitted to allocate each server to an *alternative task* (such as idling, battery recharging or service maintenance), for which we obtain an alternative-task reward rate  $\kappa$ . Thus we assume there are  $M$  alternative tasks — one for each server. For instance, the role of this alternative task with a positive  $\kappa$  could be to allocate a server to it when all the customers in the system have a relatively low profitability (e.g., low service completion reward, low waiting cost, and low abandonment penalty). On the other hand, by setting this parameter to a large negative value we may force a server to be non-idling (whenever there are waiting customers). If  $\kappa = -\infty$ , then the alternative task will never be engaged, and we call this the *non-idling variant*. Finally, if  $\kappa = 0$ , then the alternative task represents the classic possibility of server idling and we call this the *idling variant*.

The joint goal is to maximize the expected aggregate rate of service completion rewards and alternative-task rewards minus the waiting costs and abandonment penalties, over an infinite horizon. The server is assumed to be *preemptive* (i.e., the service of a customer or of the alternative task can be interrupted at any moment). Thus, the servers continuously decide which customer (if any) to serve.

The most relevant paper for our work is [Atar et al. \(2010\)](#), where the authors investigate this model under the time-average criterion and introduce the  $c\mu/\theta$ -rule for the *in-queue* cost minimization problem (i.e., the customers in service do not incur the waiting cost):

**Rule 1 ( $c\mu/\theta$ -rule)** Assign to each customer of class  $k$  in the system the following index rate:  $\frac{(c_k + d_k \theta_k) \mu_k}{\theta_k}$  and allocate the servers to the  $M$  customers with the highest rates.

It is shown in [Atar et al. \(2010\)](#) (and extended in [Atar et al. \(2011\)](#) to a general service distribution for a non-preemptive server) that the  $c\mu/\theta$ -rule minimizes asymptotically (in the multi-server fluid limit sense) the time-average *in-queue* waiting cost in the overload case (i.e., when the system would be unstable if no abandonment were possible) under Poisson arrivals.

If there is no abandonment (i.e.,  $\theta_k = 0$  for all classes  $k$ ), then this problem recovers the classic job scheduling problem considered in [Cox and Smith \(1961\)](#), [Smith \(1956\)](#), [Fife \(1965\)](#), [Buyukkoc et al. \(1985\)](#), for which the following rule is optimal in a single-server setting ( $M = 1$ ) under Poisson arrivals, for both the discounted and time-average criterion:

**Rule 2 ( $c\mu$ -rule)** Assign to each customer of class  $k$  in the system the following index rate:  $c_k \mu_k$  and allocate the servers to the  $M$  customers with the highest rates.

For a given class  $k$ ,  $c_k \mu_k$  measures the expected savings in waiting costs if a customer of class  $k$  is served. Thus, the  $c\mu$ -rule allocates the server to the customers who contribute most efficiently to the minimization of the expected aggregate waiting cost. If  $M \geq 2$ , this rule may not be optimal, but the suboptimality gap is usually very small.

## 1.2 Main Contributions

The most important contributions of the paper are as follows:

- We develop a new *versatile modeling framework to optimize the performance of non-work-conserving queueing systems*, by formulating the system with abandonments as a multi-armed restless bandit problem, which is relatively easily generalizable to include more complex system features;
- Using this stochastic optimization framework, we derive a *novel, robust, and simple scheduling rule*, and provide its intuitive interpretation;
- We show that the proposed rule achieves *near-optimal and asymptotically-optimal performance* both in single- and multi-server cases, both in overload and underload regimes, and both in idling and non-idling systems, which apart from its own sake validates the framework.

The most interesting case for practical implementation is the time-average criterion on which we focus next. Let us denote by  $C_k := r_k + d_k - c_k (1/\mu_k - 1/\theta_k)$  the *expected total profit*, since it is the difference between the expected total revenue if serving the customer always ( $r_k - c_k/\mu_k$ ) and the expected total revenue if not serving her at all ( $-d_k - c_k/\theta_k$ ). Our stochastic optimization approach yields the following Whittle Index rule (WI rule) that we propose to be implemented in practice:

**Rule 3 (WI rule)** *Assign to each customer of class  $k$  in the system the following index rate:  $C_k\mu_k$ , if  $C_k \geq 0$  and  $C_k\theta_k$ , otherwise, and to each alternative task  $k$  the following index rate:  $\kappa$ , and allocate the servers to the  $M$  options with the highest rates.*

There is a simple interpretation of the WI index rate, as measuring the *expected profit rate*, since the expected total profit  $C_k$  is divided by the expected time the customer stays in the system, which is  $1/\mu_k$  if it is preferable to serve and  $1/\theta_k$  if it is preferable not to serve.

We also emphasize the fundamental differences between our approach and results and those of the existing literature on control of systems with abandonments, to the best of our knowledge:

- this is the *first paper that tries to solve the original stochastic problem*. Previous literature focused on solving simpler limiting regimes such as deterministic fluid limits, equilibria in overload, or diffusions;
- this is the *first paper that considers the discounted criterion*, and we derive our rule under both the discounted and the time-average criterion. Moreover, our rule is optimal if  $\theta_k \rightarrow 0$  (it reduces to the  $c\mu$  rule) and gives a simple priority index in general;
- this is the *first paper that relaxes the assumption of non-idling policies*. Note that though, for many scheduling problems, idling is not optimal when preemption is permitted, because we assume no abandonment during service and the waiting cost is paid during service, we may want to idle, e.g., when the abandonment rate is high and the penalty is low. In prior work either the model assumptions are such that idling is not optimal, or policies are restricted to be non-idling. Both of these approaches may be inadequate in practice, and are taken to make the problem tractable. We show simple examples in which idling may be optimal for our modelling assumptions. Thus, our rule will be superior to existing non-idling rules in these cases. Moreover, our rule is often superior *even when idling is not optimal*.

### 1.3 Paper Structure

The problem as described above is hard to solve for optimality, and therefore we approach it by solving it approximately by using several levels of problem relaxation. [Section 2](#) contains the formulation of the problem without arrivals as a Markov Decision Process after its uniformization and discretization. This formulation can be seen as a generalization of the job sequencing problem with geometrically distributed service times formulated in [Cox and Smith \(1961\)](#), who showed that the index rule based on the  $c\mu$  index is optimal. The modeling framework is known as the multi-armed restless bandit problem, an extremely difficult optimization problem that has been proven PSPACE-hard ([Papadimitriou and Tsitsiklis 1999](#)), that is, its complexity grows exponentially in time and in memory requirements. The term restless refers to the fact that, as a consequence of abandonment, the state of all customers in the system varies in time regardless of whether they are served or not. Restless problems can be solved analytically only in a few cases (typically with largely restricted dynamics), and this explains to some extent why optimality results on scheduling in systems with abandonment are so scarce in queueing theory.

In [Section 3](#) we solve analytically the case in which there are one or two customers in the system. The case with one customer gives a fundamental insight into the problem and provides an interpretation of  $C_k$ . For the case with two customers we characterize in closed form the optimal switching curve which gives rise to a simple rule. This rule is not an index rule because it is not separable - the decision to serve one customer cannot be made by just comparing indices for each customer where the index for one customer depends only on that customer's parameters and not those of the other one.

To approach the intractable case with more than two customers, we follow the approach of [Whittle \(1988\)](#) which allows us to derive a scheduling rule for any number of customers and servers by calculating the Whittle indices as described in [Niño-Mora \(2007\)](#). In [Section 4](#) we introduce the relaxed formulation of the original problem. The main idea is to relax the sample path constraint (that imposes that exactly  $M$  options be served at a time) by letting the time-average/discounted number of options served in a period be  $M$ . This relaxation significantly simplifies the problem. The optimal policy of the relaxed formulation becomes now separable across options, that is, we can calculate for each customer the Whittle index value (that depends only on the customer's parameters and on whether she is still waiting or not), and the optimal scheduler serves in every period the customers with actual Whittle index value higher than a threshold (the value of the threshold ensures that the time-average number of customers served is precisely  $M$ ).

[Section 5](#) contains the main contribution of this paper, that is, the analysis of the relaxed problem and the heuristic rule for the original stochastic optimization formulation. The optimal policy for the relaxed problem need not be feasible for the original problem, yet it allows us to calculate the Whittle index values, which can be interpreted as dynamic shadow prices, in closed form, and derive their equivalents for the continuous-time problem in order to construct a heuristic Whittle index rule for the original problem with arrivals. The Whittle index rule is feasible (at most  $M$  customers are served at a time) and is typically reported to have an extremely good performance ([Niño-Mora 2007](#)). In addition, it was shown in [Weber and Weiss \(1990\)](#) that such a heuristic approaches optimality in the limit as the number of customers and the number of servers grows (under certain additional assumptions).

In [Section 6](#) we perform computational experiments for a wide variety of scenarios and we compare the performance of various scheduling policies. In the experiments section we consider a more realistic scenario where customers arrive randomly according to a Poisson process. Simulation results suggest that the performance of the WI rule is extremely good, and is optimal for a

broad range of parameters and scenarios. From this section we can conclude that: (i) the WI rule outperforms or is equivalent to the  $c\mu/\theta$ -rule, (ii) when there is a class of customers with  $\theta$  larger than  $\mu$ , or when  $c$ 's differ across classes, then the WI rule can significantly outperform  $c\mu/\theta$ , (iii) in cases in which the optimal scheduling policy chooses to idle instead of serving, the WI rule is much better than  $c\mu/\theta$  or  $c\mu$ , (iv) in many scenarios the WI rule is equal to the optimal policy for a broad setting of the parameter values.

Proofs are deferred to the appendix to streamline the paper.

## 2 MDP Formulation

Since the  $c\mu$ -rule is optimal both under Poisson arrivals and under no arrivals, we set out to analyze the continuous-time model without arrivals. Our aim is to obtain a rule accounting for abandonment whose performance in the continuous-time model with arrivals we later evaluate by means of numerical experiments. We set the model in the framework of the dynamic and stochastic resource allocation problem, closely following Jacko (2009). We assume there are initially  $K$  customers, i.e. one of each class.

We will uniformize and discretize the problem described in the previous section. Let us consider any sufficiently large uniformization rate (per second)  $\gamma$ , which can be interpreted as the number of periods per second. In other words,  $1/\gamma$  is the expected duration in seconds until the next event happens, which will be taken in the discretization as the duration of a discrete-time period. The uniformization and discretization is described in Table 1. In the following we suppose that the parameters are uniformized and discretized.

**Remark 1** *In our model without arrivals we require  $\gamma \geq M \max_{k \in \mathcal{K}} \mu_k + K \max_{k \in \mathcal{K}} \theta_k$  for both the discounted and time-average problem. One might consider including the rate  $\alpha$  in the uniformization in order to use a common  $\gamma$  for both cases and then to set  $\gamma = 1$  without loss of generality, so that the same parameters can be interpreted as probabilities in the discrete-time, uniformized version. This would be perfectly valid in the undiscounted case. However, we were unable to use similar tricks that would work for the discounted case (which is central to our analysis). For instance, defining  $\gamma \geq M \max_{k \in \mathcal{K}} \mu_k + K \max_{k \in \mathcal{K}} \theta_k + \alpha$  and setting  $\gamma = 1$  prohibits consideration of the myopic case (in which  $\alpha \rightarrow +\infty$ ). Even when we ignored the myopic case and tried to redo the whole analysis, we did not obtain the same index. One of the important issues is that the index in the discretized case is a value “per period”, while in the continuous case it is rate “per second”. We make this distinction in subsection 5.4 and relate it to the Lagrangian parameter  $\nu$ , which is the discretized work charge, and  $\xi$ , the corresponding rate.*

After the problem has been uniformized, we formulate the problem as a discrete time Markov Decision Process (MDP). Consider the time slotted into epochs  $t \in \mathcal{T} := \{0, 1, 2, \dots\}$  at which decisions can be made. The time epoch  $t$  corresponds to the beginning of the time period  $t$ . At  $t = 0$  there are  $K$  customers (labeled by  $k \in \mathcal{K}$ ) and  $M$  alternative tasks (labeled by  $k \in \mathcal{M} := \{K + 1, \dots, K + M\}$ ) awaiting service. Thus, there are  $K + M$  competing options, labeled by  $k \in \mathcal{K}^+ := \mathcal{K} \cup \mathcal{M}$ . At each time epoch  $t$ , the decision-maker allocates each server to exactly one (distinct) option.

### 2.1 Customers

Every customer  $k \in \mathcal{K}$  can be allocated either no or one server. We denote by  $\mathcal{A} := \{0, 1\}$  the action space, i.e., the set of allowable levels of server allocation. Here, action 0 means allocating



Parameter per second	Parameter per period
arrival rate $\lambda_k$	arrival probability $\lambda'_k := \lambda_k/\gamma$
service completion rate $\mu_k$	service completion probability $\mu'_k := \mu_k/\gamma$
abandonment rate $\theta_k$	abandonment probability $\theta'_k := \theta_k/\gamma$
exponential discount rate $\alpha$	geometric discount factor $\beta := \gamma/(\alpha + \gamma)$
alternative task reward rate $\kappa$	alternative task reward $\kappa' := \kappa/(\alpha + \gamma)$
waiting cost rate $c_k$	waiting cost $c'_k := c_k/(\alpha + \gamma)$
work charge rate $\xi$	work charge $\nu := \xi/(\alpha + \gamma)$
abandonment penalty $d_k$	abandonment penalty $d_k$
service completion reward $r_k$	service completion reward $r_k$

Table 1: Uniformization and discretization of continuous-time parameters for the discrete-time model with uniformization factor  $\gamma$ .

zero servers (i.e., “not serving”), and action 1 means allocating one server (i.e., “serving”). This action space is the same for every customer  $k$ .

Each customer  $k$  is defined independently of other customers as the tuple

$$(\mathcal{N}_k, (\mathbf{W}_k^a)_{a \in \mathcal{A}}, (\mathbf{R}_k^a)_{a \in \mathcal{A}}, (\mathbf{P}_k^a)_{a \in \mathcal{A}}),$$

where

- $\mathcal{N}_k := \{0, 1\}$  is the *state space*, where state 0 represents a service already completed or abandoned, and state 1 means that the service is uncompleted and not abandoned;
- $\mathbf{W}_k^a := (W_{k,n}^a)_{n \in \mathcal{N}_k}$ , where  $W_{k,n}^a$  is the (expected) one-period capacity consumption, or *work* required by customer  $k$  at state  $n$  if action  $a$  is decided at the beginning of a period; in particular, for any  $n \in \mathcal{N}_k$ ,

$$W_{k,n}^1 := 1, \quad W_{k,n}^0 := 0;$$

- $\mathbf{R}_k^a := (R_{k,n}^a)_{n \in \mathcal{N}_k}$ , where  $R_{k,n}^a$  is the expected one-period *revenue* earned by customer  $k$  at state  $n$  if action  $a$  is decided at the beginning of a period; in particular,

$$\begin{aligned} R_{k,0}^1 &:= 0, & R_{k,1}^1 &:= -c'_k, \\ R_{k,0}^0 &:= 0, & R_{k,1}^0 &:= -c'_k - \beta d_k \theta'_k; \end{aligned}$$

- $\mathbf{P}_k^a := (p_{k,n,m}^a)_{n,m \in \mathcal{N}_k}$  is the customer- $k$  stationary one-period *state-transition probability matrix* if action  $a$  is decided at the beginning of a period, i.e.,  $p_{k,n,m}^a$  is the probability of moving to state  $m$  from state  $n$  under action  $a$ ; in particular, we have

$$\mathbf{P}_k^1 := \begin{matrix} & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{pmatrix} 1 & 0 \\ \mu'_k & 1 - \mu'_k \end{pmatrix} \end{matrix}, \quad \mathbf{P}_k^0 := \begin{matrix} & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{pmatrix} 1 & 0 \\ \theta'_k & 1 - \theta'_k \end{pmatrix} \end{matrix}.$$

The dynamics of customer  $k$  is thus captured by the *state process*  $X_k(\cdot)$  and the *action process*  $a_k(\cdot)$ , which correspond to state  $X_k(t) \in \mathcal{N}_k$  and action  $a_k(t) \in \mathcal{A}$  at all time epochs  $t \in \mathcal{T}$ . As a result of deciding action  $a_k(t)$  in state  $X_k(t)$  at time epoch  $t$ , the customer  $k$  consumes the allocated capacity, earns the revenue, and evolves its state for the time epoch  $t + 1$ .

**Remark 2** *The attentive reader might wonder why to allow serving in state 0. This is assumed for compatibility with the methodology derived in the previous literature. It turns out that the index value of state 1 we obtain below would be the same even if serving in state 0 was not allowed because the one-period revenue is zero in this absorbing state under both actions. Moreover, the heuristic we propose below only relies on the index value of state 1.*

Note that we have defined the customer with a zero completion reward. This is without loss of generality, because the customers with a non-zero completion reward can be normalized to a cost-only customer as follows. The following proposition follows trivially from the model description.

**Proposition 1** *The expected one-period revenue for a customer with service rate  $\tilde{\mu}_k$ , abandonment rate  $\tilde{\theta}_k$ , service completion reward  $\tilde{r}_k$ , waiting cost  $\tilde{c}_k$  and abandonment penalty  $\tilde{d}_k$  is equal to the one-period revenue for a customer with service rate  $\mu_k = \tilde{\mu}_k$ , abandonment rate  $\theta_k = \tilde{\theta}_k$ , revenue  $r_k = 0$ , waiting cost  $c_k = \tilde{c}_k - \beta \tilde{r}_k \mu_k$ , and abandonment penalty  $d_k = \tilde{d}_k + \tilde{r}_k \frac{\mu_k}{\theta_k}$ .*

## 2.2 Alternative Tasks

We model each alternative task as a *static  $\kappa$ -customer* with a single state 0 and with reward  $\kappa'$  if served, i.e., such a task  $k \in \mathcal{M}$  is defined by  $\mathcal{N}_k := \{0\}$ ,  $W_{k,0}^a := a$ ,  $R_{k,0}^a := \kappa' a$ ,  $p_{k,0,0}^a := 1$  for all  $a \in \mathcal{A}$ .

## 2.3 Unified Optimization Criterion

Before describing the problem we first define an averaging operator that will allow us to discuss the infinite-horizon problem under the traditional  $\beta$ -discounted criterion and the time-average criterion in parallel. Let  $\Pi_{X,a}$  be the set of all the policies that for each time epoch  $t$  decide (possibly *randomized*) action  $a(t)$  based only on the state-process history  $X(0), X(1), \dots, X(t)$  and on the action-process history  $a(0), a(1), \dots, a(t-1)$  (i.e., *non-anticipative*). Let  $\mathbb{E}_\tau^\pi$  denote the expectation over the state process  $X(\cdot)$  and over the action process  $a(\cdot)$ , conditioned on the state-process history  $X(0), X(1), \dots, X(\tau)$  and on policy  $\pi$ .

Consider any expected one-period quantity  $Q_{X(t)}^{a(t)}$  that depends on state  $X(t)$  and on action  $a(t)$  at any time epoch  $t$ . For any policy  $\pi \in \Pi_{X,a}$ , any initial time epoch  $\tau \in \mathcal{T}$ , and any *discount factor*  $0 \leq \beta \leq 1$  we define the infinite-horizon  $\beta$ -average quantity as<sup>1</sup>

$$\mathbb{B}_\tau^\pi \left[ Q_{X(\cdot)}^{a(\cdot)}, \beta, \infty \right] := \lim_{T \rightarrow \infty} \frac{\sum_{t=\tau}^{T-1} \beta^{t-\tau} \mathbb{E}_\tau^\pi \left[ Q_{X(t)}^{a(t)} \right]}{\sum_{t=\tau}^{T-1} \beta^{t-\tau}}. \quad (1)$$

The  $\beta$ -average quantity recovers the traditionally considered quantities in the following three cases:

<sup>1</sup>For definiteness, we consider  $\beta^0 = 1$  for  $\beta = 0$ .



- *expected time-average quantity* when  $\beta = 1$ .
- *expected total  $\beta$ -discounted quantity*, scaled by constant  $1 - \beta$ , when  $0 < \beta < 1$ ;
- *myopic quantity* when  $\beta = 0$ .

Thus, when  $\beta = 1$ , the problem is formulated under the *time-average criterion*, whereas when  $0 < \beta < 1$  the problem is considered under the  *$\beta$ -discounted criterion*. The remaining case when  $\beta = 0$  reduces to a static problem and hence is considered in order to define a *myopic policy*. In the following we consider the discount factor  $\beta$  to be fixed and the horizon to be infinite, therefore we omit them in the notation and write briefly  $\mathbb{B}_\tau^\pi \left[ Q_{X(\cdot)}^{a(\cdot)} \right]$ .

## 2.4 Optimization Problem

We now describe in more detail the problem we consider. Let  $\Pi_{X,a}$  be the space of randomized and non-anticipative policies depending on the joint state-process  $\mathbf{X}(\cdot) := (X_k(\cdot))_{k \in \mathcal{K}^+}$  and deciding the joint action-process  $\mathbf{a}(\cdot) := (a_k(\cdot))_{k \in \mathcal{K}^+}$ , i.e.,  $\Pi_{X,a}$  is the *joint policy space*.

For any discount factor  $\beta$ , the problem is to find a joint policy  $\pi$  maximizing the *value function* given by the  $\beta$ -average aggregate revenue starting from the initial time epoch 0 subject to the family of *sample path* allocation constraints, i.e.,

$$V(\mathbf{X}(0)) := \max_{\pi \in \Pi_{X,a}} \mathbb{B}_0^\pi \left[ \sum_{k \in \mathcal{K}^+} R_{k, X_k(\cdot)}^{a_k(\cdot)} \right] \quad (\text{P})$$

$$\text{subject to } \mathbb{E}_t^\pi \left[ \sum_{k \in \mathcal{K}^+} a_k(t) \right] = M, \text{ for all } t \in \mathcal{T}$$

Note that the constraint could equivalently be expressed in words as that for all  $t \in \mathcal{T}$ :  $\sum_{k \in \mathcal{K}^+} a_k(t) = M$  under policy  $\pi$  and for any possible joint state-process history  $\mathbf{X}(0), \mathbf{X}(1), \dots, \mathbf{X}(t)$ .

We say that a policy is Markovian deterministic if it chooses the vector  $\mathbf{a}(t)$  according to a fixed rule based solely on the current system state  $\mathbf{X}(t)$ . Let  $\Pi^{\text{MD}}$  denote the set of Markovian deterministic policies. From standard results in Markov decision processes (Puterman 2005, Chapter 6) it follows that there is no loss of generality in restricting our attention to the policies in  $\Pi^{\text{MD}}$ .

The solution to problem (P) can be found by determining the unique solution  $V(\cdot)$  to Bellman's equation, which for  $\beta < 1$  is

$$V(\mathbf{x}) = \max_{\mathbf{a} \text{ s.t. } \sum_{k \in \mathcal{K}^+} a_k = M} \left\{ \sum_{k \in \mathcal{K}^+} R_{k, x_k}^{a_k} + \beta \sum_{\mathbf{y}} q_{\mathbf{x}, \mathbf{y}}^{\mathbf{a}} V(\mathbf{y}) \right\} \quad (2)$$

where  $q_{\mathbf{x}, \mathbf{y}}^{\mathbf{a}}$  denotes the probability of having a transition from state  $\mathbf{x}$  to state  $\mathbf{y}$  in one time-period under action profile  $\mathbf{a}$ . As a consequence of the uniformization procedure, it follows that there can be at most one transition in every one-time period, that is, either a service completion, an abandonment, or a dummy transition due to the uniformization.

### 3 Optimal Solution for Special Cases

Problem (P) is hard to solve in its whole generality, but we have identified special cases that admit an analytical solution of the Bellman equation, summarized in this section. To the best of our knowledge these particular have not been considered previously in the literature.

Let us first denote by

$$\begin{aligned} C_k &:= \beta \frac{c'_k(\mu'_k - \theta'_k) + d_k \theta'_k (1 - \beta + \beta \mu'_k)}{(1 - \beta + \beta \theta'_k)(1 - \beta + \beta \mu'_k)} \\ &= \frac{c_k(\mu_k - \theta_k) + d_k \theta_k (\alpha + \mu_k)}{(\alpha + \theta_k)(\alpha + \mu_k)}. \end{aligned} \quad (3)$$

**Proposition 2**  $C_k$  is the difference between the expected total revenue if serving the customer always and the expected total revenue if not serving her at all.

We remark that in the undiscounted case  $C_k$  simplifies as follows:

$$C_k = d_k - c'_k \left( \frac{1}{\mu'_k} - \frac{1}{\theta'_k} \right) = d_k - c_k \left( \frac{1}{\mu_k} - \frac{1}{\theta_k} \right). \quad (4)$$

#### 3.1 Single Customer at a Single Server

In the single server case with a single customer competing with a single alternative task, we introduce the following index value (1U):

$$\nu_k^{1U} := C_k. \quad (5)$$

This index value thus inherits the interpretation from  $C_k$ , so it is positive if and only if serving the customer always is more profitable than not serving her at all.

**Proposition 3** Let  $K = 1$ ,  $M = 1$  and the alternative task reward  $\kappa' = 0$ . The solution to Problem (P) is:

- (i) If  $\nu_1^{1U} \geq 0$ , then it is optimal to serve customer 1;
- (ii) If  $\nu_1^{1U} \leq 0$ , then it is optimal to allocate the server to the alternative task ( $k = 2$ ).

It is important to note that the above result means that idling may be optimal for a very impatient customer.

#### 3.2 Two Customers at a Single Server

In the case of two customers competing among themselves (2U) for a single server, due to the technical complexity of the problem we focus on the undiscounted case ( $\alpha = 0$ , i.e.,  $\beta = 1$ ). We introduce the following *non-separable index* value for customer  $k = 1, 2$  with respect to the other customer ( $3 - k$ ):

$$\nu_k^{2U} := \frac{C_k \theta'_k}{\theta'_k + \mu'_{3-k}}. \quad (6)$$

We note that this expression is not separable since it depends on the other customer's parameters, though only through the service rate  $\mu'_{3-k}$ , so does not lead to an index rule in the traditional sense.

We obtain the following characterization of the optimal policy in this case, under the additional assumption that there is no alternative task, or equivalently that  $\kappa' = -\infty$ .

**Proposition 4** Suppose that  $K = 2$  and  $C_k \geq 0$  for  $k \in \mathcal{K}$ . The following holds for problem (P) with  $M = 1$ ,  $\alpha = 0$ , and the alternative task reward  $\kappa' = -\infty$ :

- (i) If  $\nu_1^{2U} \geq \nu_2^{2U}$ , then it is optimal to serve customer 1;
- (ii) If  $\nu_1^{2U} \leq \nu_2^{2U}$ , then it is optimal to serve customer 2.

## 4 Relaxations and Decomposition

For larger values of  $K$  and  $M$ , the problem is analytically intractable, and therefore we approach it in a different way. The main idea is to solve a relaxed version of the problem (P). It turns out that the relaxation will in fact allow decomposition of the problem, and the optimal solution to the relaxed problem can be obtained by solving parametric single-option subproblems. In Section 5 we will then show how the solution of the relaxed problem can be used to construct a nearly-optimal heuristic for the original problem (P).

### 4.1 Relaxations

We can use the fact that  $W_{k, X_k(t)}^{a_k(t)} = a_k(t)$  (cf. definitions in Section 2) and instead of the constraints in (P), for notational reasons, we consider the sample path *consumption* constraints  $\mathbb{E}_t^\pi \left[ \sum_{k \in \mathcal{K}^+} W_{k, X_k(t)}^{a_k(t)} \right] = M$ , for all  $t \in \mathcal{T}$ . These constraints imply the *epoch- $t$  expected consumption* constraints,

$$\mathbb{E}_0^\pi \left[ \sum_{k \in \mathcal{K}^+} W_{k, X_k(t)}^{a_k(t)} \right] = M, \text{ for all } t \in \mathcal{T} \quad (7)$$

requiring that the capacity be fully allocated at every time epoch if conditioned on  $\mathbf{X}(0)$  only. Finally, we may require this constraint to hold only on  $\beta$ -average, as the  *$\beta$ -average capacity consumption constraint*

$$\mathbb{B}_0^\pi \left[ \sum_{k \in \mathcal{K}^+} W_{k, X_k(\cdot)}^{a_k(\cdot)} \right] = \mathbb{B}_0^\pi [M]. \quad (8)$$

Using  $\mathbb{B}_0^\pi [M] = M$ , we obtain the following *relaxation* of problem (P),

$$\begin{aligned} & \max_{\pi \in \Pi_{\mathbf{X}, \alpha}} \mathbb{B}_0^\pi \left[ \sum_{k \in \mathcal{K}^+} R_{k, X_k(\cdot)}^{a_k(\cdot)} \right] \\ & \text{subject to } \mathbb{B}_0^\pi \left[ \sum_{k \in \mathcal{K}^+} W_{k, X_k(\cdot)}^{a_k(\cdot)} \right] = M. \end{aligned} \quad (\mathbf{P}^W)$$

This relaxation was introduced in Whittle (1988), and it gives us the following result.

**Proposition 5** Problem  $(\mathbf{P}^W)$  is a relaxation of problem (P). Moreover,  $(\mathbf{P}^W)$  provides an upper bound for the optimal value of problem (P).

The *Whittle relaxation* ( $\mathbf{P}^W$ ) can be approached by traditional Lagrangian methods, introducing a Lagrangian parameter, say  $\nu$ , interpreted as the *work charge*, to dualize the constraint, obtaining the following Lagrangian relaxation,

$$\max_{\pi \in \Pi_{\mathbf{X},a}} \mathbb{B}_0^\pi \left[ \sum_{k \in \mathcal{K}^+} R_{k, X_k(\cdot)}^{a_k(\cdot)} - \nu \sum_{k \in \mathcal{K}^+} W_{k, X_k(\cdot)}^{a_k(\cdot)} \right] + \nu M. \quad (\mathbf{P}_\nu^L)$$

The classic Lagrangian result is the following:

**Proposition 6** *For any  $\nu$ , problem ( $\mathbf{P}_\nu^L$ ) is a relaxation of problem ( $\mathbf{P}^W$ ), and further a relaxation of problem ( $\mathbf{P}$ ). Moreover, ( $\mathbf{P}_\nu^L$ ) for every  $\nu$  provides an upper bound for the optimal value of both problem ( $\mathbf{P}^W$ ) and problem ( $\mathbf{P}$ ).*

## 4.2 Decomposition into Single-Option Subproblems

We now set out to decompose the optimization problem ( $\mathbf{P}_\nu^L$ ) as is standard for Lagrangian relaxations, considering  $\nu$  as a parameter. Notice that any joint policy  $\pi \in \Pi_{\mathbf{X},a}$  defines a set of single-option policies  $\tilde{\pi}_k$  for all  $k \in \mathcal{K}^+$ , where  $\tilde{\pi}_k$  is a randomized and non-anticipative policy depending on the *joint* state-process  $\mathbf{X}(\cdot)$  and deciding the *customer- $k$*  action-process  $a_k(\cdot)$ . We will write  $\tilde{\pi}_k \in \Pi_{\mathbf{X},a_k}$ . We will therefore study the customer- $k$  subproblem

$$\max_{\tilde{\pi}_k \in \Pi_{\mathbf{X},a_k}} \mathbb{B}_0^{\tilde{\pi}_k} \left[ R_{k, X_k(\cdot)}^{a_k(\cdot)} - \nu W_{k, X_k(\cdot)}^{a_k(\cdot)} \right]. \quad (9)$$

## 5 Solution

In this section we explain our approach in order to derive a nearly-optimal scheduling discipline for the original problem ( $\mathbf{P}$ ). In [subsection 5.1](#) we will identify a set of optimal policies for problem (9), which will be used in [subsection 5.2](#) to solve problem ( $\mathbf{P}^W$ ). In [subsection 5.3](#) we will then construct a joint policy that is feasible but not necessarily optimal for problem ( $\mathbf{P}$ ), while [subsection 5.4](#) is dedicated to designing policies for the original, continuous-time problem.

### 5.1 Optimal Solution to Single-Option Subproblem via the Whittle Index

We will identify a set of optimal policies  $\tilde{\pi}_k^*$  for (9) for all options  $k$ . Problem (9) falls into the framework of *restless bandits* and can be optimally solved by assigning a set of Whittle index values  $\nu_{k,n}$  to each state  $n \in \mathcal{N}_k$  under certain conditions ([Niño-Mora 2007](#)), which we prove valid for our problem.

Let us denote for customer  $k \in \mathcal{K}$ ,  $\nu_{k,0} := 0$ , and

$$\nu_{k,1} := \begin{cases} C_k(1 - \beta + \beta\mu'_k), & \text{if } C_k \geq 0, \\ C_k(1 - \beta + \beta\theta'_k), & \text{if } C_k < 0, \end{cases} \quad (10)$$

where  $C_k$  is given in (3), and for alternative task  $k \in \mathcal{M}$ ,  $\nu_{k,0} := \kappa'$ . The following is the main theoretical result of the paper.

**Theorem 1** *For problem (9), the following hold (where, in the case of equality, both actions are optimal):*

- (i) *it is optimal to serve waiting customer  $k \in \mathcal{K}$  if and only if  $\nu \leq \nu_{k,1}$ ;*

- (ii) it is optimal to serve customer  $k \in \mathcal{K}$  when it is already completed or abandoned if and only if  $\nu \leq \nu_{k,0}$ ;
- (iii) it is optimal to serve the alternative task  $k \in \mathcal{M}$  if and only if  $\nu \leq \nu_{k,0}$ ;

The proof of this proposition is based on establishing indexability of the problem and computing the index values, following the survey [Niño-Mora \(2007\)](#). The full proof is presented in [Appendix D](#).

## 5.2 Optimal Solution to Relaxations

The vector of policies  $\pi^* := (\tilde{\pi}_k^*)_{k \in \mathcal{K}^+}$  identified in [Theorem 1](#) is formed by mutually independent single-option optimal policies. Therefore this vector is an optimal policy to the Lagrangian relaxation  $(P_\nu^L)$ .

Since a finite-state MDP admits an LP formulation using the standard *state-action frequency* variables (as observed in [Niño-Mora \(2007\)](#)), strong LP duality implies that there exists  $\nu^*$  (possibly depending on the joint initial state) such that the Lagrangian relaxation  $(P_{\nu^*}^L)$  achieves the same objective value as  $(P^W)$ . Further, if  $\nu^* \neq 0$ , then LP complementary slackness ensures that the  $\beta$ -average capacity constraint (8) is satisfied by any optimal solution to  $(P_{\nu^*}^L)$ .

## 5.3 Whittle Index Rule for Discretized Problem

Since the discrete-time problem (P) requires us to allocate each server to exactly one option (a customer or an alternative task), [Whittle \(1988\)](#) proposed the following heuristic.

**Rule 4 (Whittle Index Rule):** Assign to each customer of class  $k$  in the system and to each alternative task  $k$  the Whittle index value  $\nu_{k, X_k(t)}$ , and allocate the servers to the  $M$  options with the highest actual Whittle index values.

## 5.4 Index Rules for Original Continuous-Time Problem

For the original continuous-time problem, we obtain the continuous-time *Whittle index rates* (per second)  $\xi_{k,n}$  from the discretized Whittle index values (per event)  $\nu_{k,n}$  multiplying them by the rate  $\alpha + \gamma$ , i.e.,  $\xi_{k,n} := \nu_{k,n}(\alpha + \gamma)$ , as now these come from the model with work charge rate  $\xi$  rather than the work charge  $\nu$ . The obtained Whittle index rates thus become independent of the uniformization rate. For customer  $k \in \mathcal{K}$ ,

$$\xi_{k,1} := \begin{cases} \left[ \frac{d_k \theta_k}{\alpha + \theta_k} - c_k \left( \frac{1}{\alpha + \mu_k} - \frac{1}{\alpha + \theta_k} \right) \right] (\alpha + \mu_k), & \text{if } C_k \geq 0, \\ \left[ \frac{d_k \theta_k}{\alpha + \theta_k} - c_k \left( \frac{1}{\alpha + \mu_k} - \frac{1}{\alpha + \theta_k} \right) \right] (\alpha + \theta_k), & \text{if } C_k < 0. \end{cases} \quad (11)$$

and for alternative task  $k \in \mathcal{M}$ ,  $\xi_{k,0} := \kappa$ . Note that the Whittle index rule remains consistent, as it maintains the order of the Whittle index rates the same as the order of the Whittle index values.

In the undiscounted case ( $\alpha = 0$ ), we obtain the time-average version of the Whittle index rate of customer  $k$ , which we call *WI* throughout the paper

$$\xi_{k,1}^{\text{WI}} := \begin{cases} C_k \mu_k = [d_k - c_k (1/\mu_k - 1/\theta_k)] \mu_k, & \text{if } C_k \geq 0 \\ C_k \theta_k = [d_k - c_k (1/\mu_k - 1/\theta_k)] \theta_k, & \text{if } C_k < 0 \end{cases} \quad (12)$$

where we have used  $C_k = d_k - c_k(1/\mu_k - 1/\theta_k)$ . As already mentioned earlier, this allows to interpret that WI is positive if and only if the expected total revenue from serving forever is greater than the expected total revenue from not serving ever, i.e.,  $C_k > 0$ . In [Section 6](#) we consider this undiscounted index rule and investigate its performance in systems with customer arrivals through computational experiments.

We remark that any class without abandonment (i.e., having  $\theta_k = 0$  as in the classic job scheduling problem) has the Whittle index that reduces to the Gittins index (see [Gittins 1989](#))

$$\nu_{k,1}^{c\mu} := \frac{c'_k \mu'_k}{1 - \beta}, \quad \xi_{k,1}^{c\mu} := \frac{c_k \mu_k}{\alpha}, \quad (13)$$

which is just the classic  $c\mu$  index scaled by a constant. The WI of such a class is  $+\infty$ , hence this class gets an absolute priority over any class with positive abandonment rate under the time-average criterion.

We further note that  $\alpha \rightarrow +\infty$ , i.e.,  $\beta \rightarrow 0$  gives rise to the myopic version of the Whittle index, which is

$$\nu_{k,1}^{\text{Myopic}} := 0, \quad \xi_{k,1}^{\text{Myopic}} := d_k \theta_k. \quad (14)$$

## 5.5 Relation among Index Policies

For completeness, we denote the 2U (for two users) rate by

$$\xi_k^{2U} = \frac{C_k \theta_k}{\theta_k + \mu_{3-k}}. \quad (15)$$

Recall from [Proposition 4](#) that the 2U rule is optimal in the non-idling system with two customers satisfying  $C_k \geq 0$  for  $k = 1, 2$ .

We further remark that it was shown in [Atar et al. \(2010\)](#) that for a non-zero abandonment penalty, the  $c\mu/\theta$  index rate is

$$\xi_{k,1}^{c\mu/\theta} := [d_k + c_k/\theta_k] \mu_k. \quad (16)$$

We also note that the WI index rate of equation (12) can equivalently be written as  $\xi_{k,1}^{WI} = [d_k + c_k/\theta_k] \mu_k - c_k$  if  $C_k \geq 0$ , and  $\xi_{k,1}^{WI} = [d_k - c_k/\mu_k] \theta_k + c_k$  if  $C_k < 0$ . In the case  $C_k \geq 0$  both the WI and the  $c\mu/\theta$  rules are very close. If in addition  $c_k = 1$  for all  $k$ , then the two rules are identical. However, both rules can also yield very different behavior. For example let us assume  $C_k < 0$  and set  $d_k := 0$ , which implies that the WI index becomes  $\xi_{k,1}^{WI} = c_k(-\theta_k/\mu_k + 1)$ , as compared to  $\xi_{k,1}^{c\mu/\theta} = c_k \mu_k / \theta_k$ . With  $c\mu/\theta$ , the server always chooses serving rather than idling, whereas under the WI rule, the server might prefer to idle if  $\theta_k > \mu_k$  since the index value becomes negative (there is no benefit in serving a customer that is more likely to abandon than complete service).

Although we obtain an index rule different from the  $c\mu/\theta$  rule, our main contribution rather lies in developing a *framework* based on the restless bandit problem using Whittle's approach to design a heuristic that can handle abandonments and that is relatively easily *generalizable* to more complex features. For instance, if we modify the model so customers in service do not incur a holding cost, the WI index becomes equivalent to the  $c\mu/\theta$  rule as defined in (16). Similarly, if the customer in service can also abandon we obtain two additional indices: if all customers incur a holding cost the WI index is again given by (16), and if only customers in the queue incur a waiting cost, then the index becomes  $[d_k + c_k/\theta_k] \mu_k + c_k$ .



As a final remark we note that in [Atar et al. \(2010\)](#) the  $c\mu/\theta$  rule was obtained as the rule that optimizes a (deterministic) fluid-model of a multi-server queue in overload. Our approach allows us to obtain index rules *without assuming an overload regime*. This is relevant because the fluid-based technique does not apply under this regime, since in the underload case the fluid converges to 0 regardless of the scheduling policy.

## 6 Computational Experiments

In this section we report on an exhaustive study of numerical experiments of the continuous-time model. We consider a system with two classes of customers. Each class  $k$  is characterized by a set of values for the parameters  $\mu_k$ ,  $\theta_k$ ,  $c_k$  and  $d_k$  as before, and the mean rate  $\lambda_k$  of Poisson arrivals.

**Remark 3** *Even though the WI rule is obtained by solving a model without arrivals, existing literature gives strong evidence to support the claim that the WI rule may perform very well also in the presence of arrivals, particularly if the arrival process is memory-less (Bernoulli or Poisson). Indeed, it has been shown in a wide variety of models that the optimal scheduling policy with a fixed number of customers is also optimal in the case of arrivals. See for example [Sevcik \(1974\)](#) and [Gittins \(1989, Theorem 3.28\)](#) for the M/G/1 queue, [Smith \(1956\)](#) and [Fife \(1965\)](#), [Buyukkoc et al. \(1985\)](#) for the  $c\mu$ -rule, [Meilijson and Weiss \(1977\)](#) for a single-server queue with feedback and [Gittins and Jones \(1974\)](#), [Gittins \(1989\)](#) and [Whittle \(1981\)](#) for the multi-armed bandit problem.*

We truncate the state space by allowing a maximum number of customers in each class, and we then use the uniformization technique in order to obtain a discrete-time representation of the model. Note that all the policies are independent of the uniformization factor. Using value iteration ([Puterman 2005](#)) we obtain numerically the optimal policy, and we then calculate the relative suboptimality gap produced by the WI,  $c\mu/\theta$ ,  $c\mu$ , 2U and Myopic rules (see [subsection 5.4](#) for the definitions). Recall that, using our notation, the  $c\mu/\theta$  index of [Atar et al. \(2010\)](#) becomes  $\frac{(c_k + d_k \theta_k) \mu_k}{\theta_k}$ . In our study we make sure that the truncation levels are large enough so that the optimal policy obtained in this way is almost surely the optimal policy in the original non-truncated problem.

We will evaluate the performance of the above policies both for idling and non-idling systems, where, in non-idling systems, the server is not allowed to idle when customers are present. Before doing so, we first define the performance evaluation measures we use to report the experimental results.

### 6.1 Performance Evaluation Measures

Let  $D^\pi$  denote the time-average objective value when policy  $\pi$  is employed in every period. Further, by solving (P) optimally by value iteration we obtain the maximizing policy, which yields the optimal objective value  $D^{\max}$ . The *relative suboptimality gap* of policy  $\pi$ , often used in literature, is defined as

$$\text{rsg}(\pi) = \frac{D^{\max} - D^\pi}{|D^{\max}|}. \quad (17)$$

Clearly, as long as  $D^{\max} \neq 0$ , we have  $0 \leq \text{rsg}(\pi) \leq \frac{D^{\max} - D^{\min}}{|D^{\max}|}$ , where  $\text{rsg}(\pi) = 0$  is obtained by the maximizing policy. However, the worst-case, minimizing policy may have a non-zero objective value. We can therefore conclude that this measure may overestimate the quality of  $\pi$  by reporting a small value even for the worst-case policy.

	$\mu_1$	$\mu_2$	$\theta_1$	$\theta_2$	$c_1$	$c_2$	$d_1$	$d_2$	$\lambda_1 = \lambda_2$
Scenario 1	0.7	0.3	[0, 1]	0.2	27	27	1	1	1
Scenario 2	0.4	0.59	[0, 2]	4	1	1	1	1	1
Scenario 3	0.8	0.7	1.2	2.7	1	1	[0, 3]	1	1
Scenario 4	0.8	0.7	1.2	2.7	1	1	[0, 10]	5	1
Scenario 5	0.4	0.1	0.5	0.8	1	[0.01, 20]	0.035	0.035	1
Scenario 6	0.4	0.22	0.1	0.2	1	[0.01, 40]	1	1	1
Scenario 7	0.4	0.3	0.001	0.03	1	[0.01, 60]	1	1	1
Scenario 8	0.4	0.3	0.001	0.03	1	1	1	1	[0, 2]

Table 2: Parameters for all the scenarios in numerical experiments.

In our experiments we have also considered another measure, the *adjusted relative suboptimality gap* of policy  $\pi$ , defined as

$$\text{arsg}(\pi) = \frac{D^{\max} - D^\pi}{D^{\max} - D^{\min}}, \quad (18)$$

where  $D^{\min}$  is the worst-case (minimizing) policy. With this measure we always have  $0 \leq \text{arsg}(\pi) \leq 1$  (as long as  $D^{\max} - D^{\min} \neq 0$ ), and both limiting values can be achieved.

The results we have obtained for both measures do not differ from each other substantially, and we thus only report on the results for  $\text{rsg}(\pi)$ .

## 6.2 Single-server Idling System

First we consider a system that can idle ( $\kappa = 0$ ). This assumption has an influence on the characterization of the optimal policy, the worst-case policy and also the WI and 2U rules, where idling gets priority over any customer with a negative index rate. It does not affect the policies defined by the rest of the rules ( $c\mu$ ,  $c\mu/\theta$  and Myopic) because these rules choose some customer whenever there is any.

Given the number of parameters that we can choose from, the number of scenarios one can construct is virtually unbounded. We have investigated a wide range of settings for the parameters in hundreds of scenarios, and we report here the results of six representative scenarios in order to provide a global panorama. We further present two additional scenarios with unique and peculiar results. In most of the scenarios, only a single parameter is varied in order to easily illustrate the effect. In [Table 2](#) we present the parameters considered in each of the scenarios. We say that the system is in *overload* if

$$\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} \geq 1,$$

which implies that abandonment is required in order to stabilize the system. Note that the system is in overload (as in [Atar et al. 2010](#)) in Scenarios 1–7. The system in Scenario 8 is in underload for smaller values of  $\lambda$  and in overload for larger values of  $\lambda$ .

There are some general conclusions that we can draw:

- WI is optimal for the majority of values of the varied parameter in all the scenarios;
- WI is almost always equivalent to or outperforms  $c\mu/\theta$ , which is in turn almost always equivalent or outperforms  $c\mu$ ;

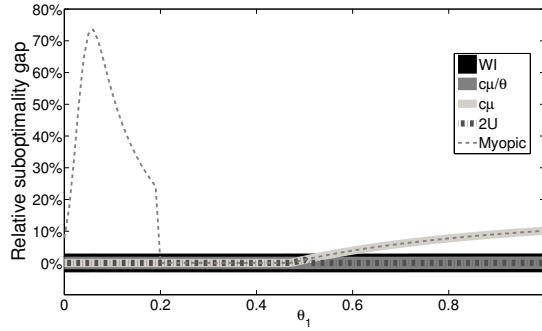


Figure 1: Relative suboptimality gap in Scenario 1.

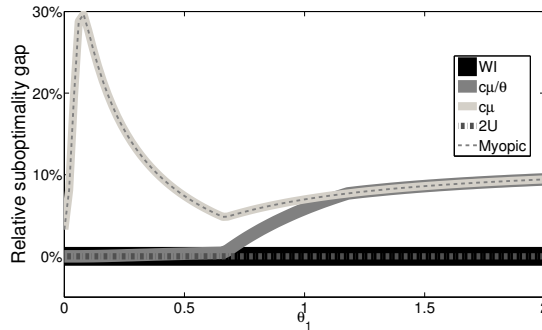


Figure 2: Relative suboptimality gap in Scenario 2.

- In cases in which the optimal policy chooses to idle instead of serving, WI is much better than  $c\mu/\theta$  or  $c\mu$ ;
- The switching point of 2U is often very close to WI, but usually its suboptimality region is larger;
- If both the 2U and WI policies give priority to class 1, then it is almost always optimal to serve class 1.

The first six scenarios illustrate these general conclusions.

**Scenario 1.** (Figure 1) In this scenario the WI and  $c\mu/\theta$  rules are optimal and the performance of the 2U rule is equivalent except for small values of  $\theta_1$ . The performance of  $c\mu$  is good only for lower values of  $\theta_1$ , and the difference with respect to WI and  $c\mu/\theta$  is the switching point where the other policies start to serve class-2 customers while  $c\mu$  does not, which makes its suboptimality grow.

**Scenario 2.** (Figure 2) Because of class-2 high abandonment rates, it is never optimal to serve class 2 and it is optimal to serve class 1 only for lower values of  $\theta_1$ . The 2U and WI policies capture this feature and are optimal. The  $c\mu$ -rule always gives priority to class 2, while the  $c\mu/\theta$ -rule correctly prioritizes class 1 for lower values of  $\theta_1$ , but then switches to class 2, which leads to poor performance.

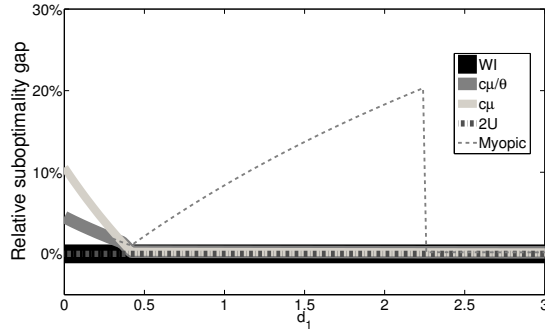


Figure 3: Relative suboptimality gap in Scenario 3.

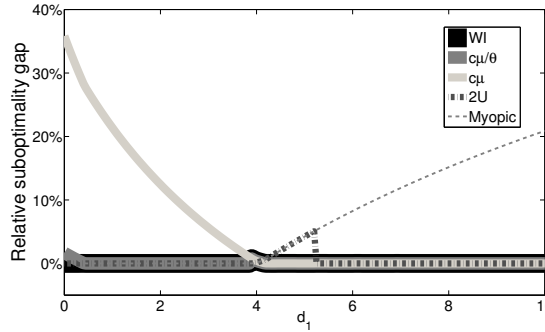


Figure 4: Relative suboptimality gap in Scenario 4.

**Scenario 3.** (Figure 3) For values of  $d_1$  smaller than approximately 0.42, the optimal policy does not serve any customer, which is captured by WI and 2U. The other policies however perform dramatically worse. For larger values it is optimal to serve class 1 customers (or to idle if no class 1 customer is present). All the policies give priority to class 1, with the only difference being that WI and 2U idle if there is no class 1 customer, whereas  $c\mu$  and  $c\mu/\theta$  serve class 2 in such a case.

**Scenario 4.** (Figure 4) Though the only difference between this scenario and the previous one is that the value of  $d_2$  is now 5 instead of 1, there is a significant difference in the results. Even though the  $\theta$ 's are still larger than the  $\mu$ 's, in this case the performance of 2U and WI differ during a non-negligible range of values for  $d_1$ . The optimal policy is to serve class 2 and then switch to serve class 1. The  $c\mu$ -rule always serves class 1, the Myopic rule always serves class 2, and the other policies start serving class 2 (as a consequence of high  $d_2$ ), and switch to serve class 1 (first WI together with  $c\mu/\theta$ , and afterwards 2U) as the value of  $d_1$  grows.

**Scenario 5.** (Figure 5) The optimal policy is to idle. Policies WI and 2U give priority to idling. The policies  $c\mu$  and  $c\mu/\theta$  for small values of  $c_2$  serve class 1. As the value of  $c_2$  increases, the policies  $c\mu$  and  $c\mu/\theta$  switch to give priority to class 2, what makes a sudden increase in the cost function. The performance of WI and 2U is optimal. The key difference is that WI depends on the difference  $\mu_2 - \theta_2$  (which is negative), and thanks to this it chooses not to serve class 2 regardless of the value of  $c_2$ .

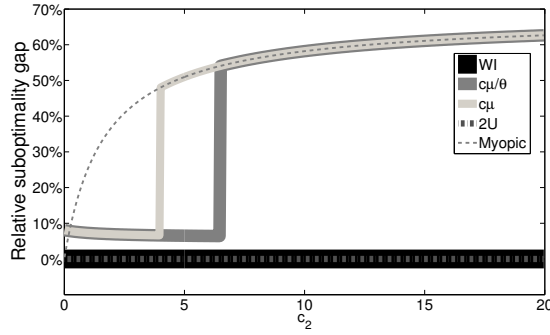


Figure 5: Relative suboptimality gap in Scenario 5.

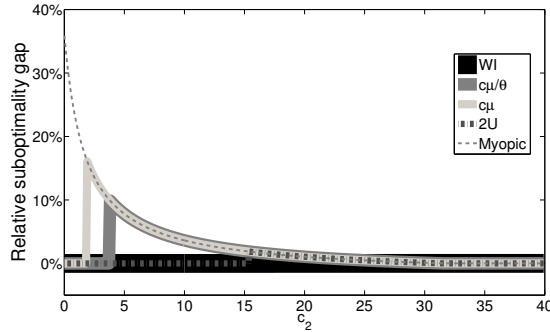


Figure 6: Relative suboptimality gap in Scenario 6.

**Scenario 6.** (Figure 6) This is a particularly interesting scenario. Service rates are larger than abandonment rates, and the policy WI has again better performance than the other policies. In fact, WI is optimal for all values of  $c_2$  with the exception of a small range around 32. The optimal policy starts serving class 1 in almost all joint states, but it starts serving class 2 in more joint states as the value of  $c_2$  increases. The WI policy serves class 1 with strict priority for values of  $c_2$  smaller than 32, and class 2 from that moment on. The upward jump for the other indices happens when they start giving priority to class 2 (first  $c_\mu$  switches, then  $c_\mu/\theta$  and then 2U).

The following two scenarios illustrate some specific and uncommon phenomena we have found in our experiments.

**Scenario 7.** (Figure 7) In this scenario the abandonment rate is very small, say negligible. We recall that without abandonment, the  $c_\mu$ -rule is optimal (Cox and Smith 1961, Buyukkoc et al. 1985). In the numerical experiments we see that, with a rather surprising exception for  $c_2 = 1$ , the  $c_\mu$ -rule is indeed optimal in this case and the 2U rule is equivalent to  $c_\mu$ . Policies WI and  $c_\mu/\theta$  start serving class 1, and switch later on to class 2 when the value of  $c_2$  becomes sufficiently large.

We emphasize that this is the only scenario we have found where the decision pattern of 2U and WI differs completely and also the only one in which  $c_\mu/\theta$  outperforms WI (for values of  $c_2$  between 40 and 44).

**Scenario 8.** (Figure 8) We have tested a wide variety of scenarios in underload; the qualitative properties of WI do not differ notably from overloaded systems, and there was no sharp jump in performance when crossing the frontier. For illustration of a rare phenomenon, we have selected

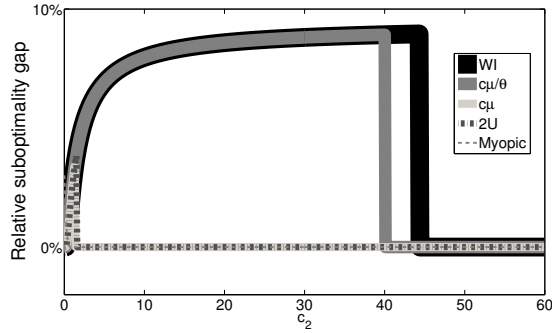


Figure 7: Relative suboptimality gap in Scenario 7.

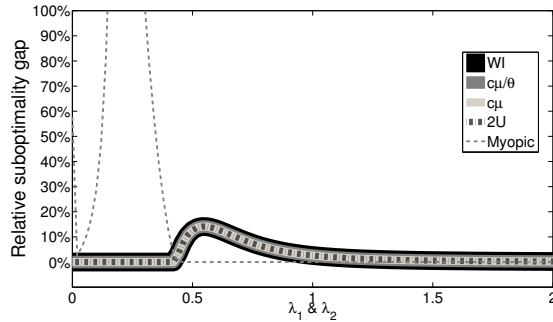


Figure 8: Relative suboptimality gap in Scenario 8.

a scenario, in which all the rules are optimal in underload, but suboptimal in overload, which is in contrast with, e.g., the results of asymptotic optimality of  $c\mu/\theta$ .

This scenario is in underload at the beginning and becomes overloaded as  $\lambda_1$  and  $\lambda_2$  grow. The optimal policy is to give priority to class 1 and then switch to class 2 in some joint states. The Myopic rule always serves class 2 and all the other rules always serve class 1. This scenario thus illustrates that the arrival rates should be taken into account for optimal scheduling, though it is very uncommon to observe such a huge effect.

### 6.3 Single-server Non-idling System

In this case we do not allow the system to idle, that is, if there are customers in the system, the scheduler must necessarily select one to serve. The policies  $c\mu$ ,  $c\mu/\theta$  and Myopic have the same priorities as before, but the optimal policy, the worst-case policy and the WI and 2U rule may behave differently.

We have performed numerical experiments for the same scenarios as in the idling case (see Table 2). The qualitative performance of the policies is analogous, but whenever there is suboptimality of any policy, it is in general higher than in the idling case. To illustrate this we discuss the results obtained for Scenarios 2 and 5. In these two scenarios it was optimal to idle in the idling system, which is not allowed now.

**Non-idling Scenario 2.** (Figure 9) The WI rule is no longer optimal for larger values of  $\theta_1$ , since it prioritizes class 1 until approximately 1.8, when it switches to class 2. Nevertheless for a non-



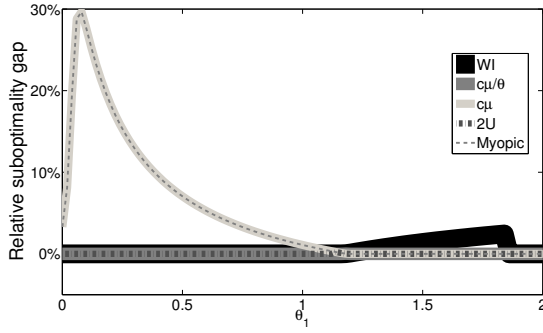


Figure 9: Relative suboptimality gap in non-idling Scenario 2.

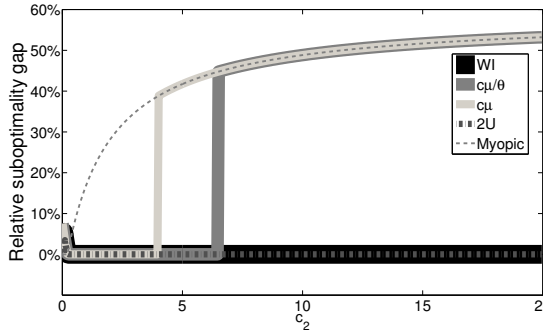


Figure 10: Relative suboptimality gap in non-idling Scenario 5.

negligible interval the WI rule is better than or equivalent to the  $c\mu$ ,  $c\mu/\theta$  and Myopic rules. When its performance is inferior to the other policies, the suboptimality in absolute terms is small.

**Non-idling Scenario 5.** (Figure 10) WI and 2U are no longer optimal for a small interval of  $c_2$  values, but they still outperform the  $c\mu/\theta$ ,  $c\mu$  and Myopic rule, because they do not switch to prioritize class 2, while all the other policies do.

## 6.4 Multi-server Idling System

In order to study the Whittle index rule in the multi-server case, we evaluate the performance of all the rules and the optimal policy for 2 servers. We believe that this sheds light on the expected performance of these rules even for more servers. Due to the curse of dimensionality we have not evaluated the optimal policy for more servers.

We investigated the same scenarios (see Table 2) as previously. Performance of the rules in the two-server case is very similar qualitatively to the single-server system. The WI rule is again almost always optimal, and the eventual suboptimality is small. However, the main difference is that the relative suboptimality gap of the other rules becomes more severe. We illustrate this in Figure 11 for Scenario 2 and in Figure 12 for Scenario 5, where the relative suboptimality gap approximately doubled.

Therefore, we believe that in the multi-server case the observations we have drawn for the single-server case are still valid and the benefit of proposed WI rule should rise with the number of servers.

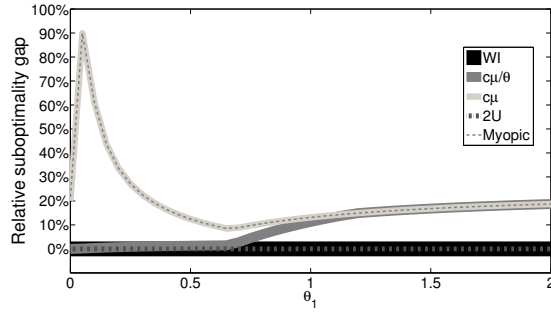


Figure 11: Multi-server relative suboptimality gap in idling Scenario 2.

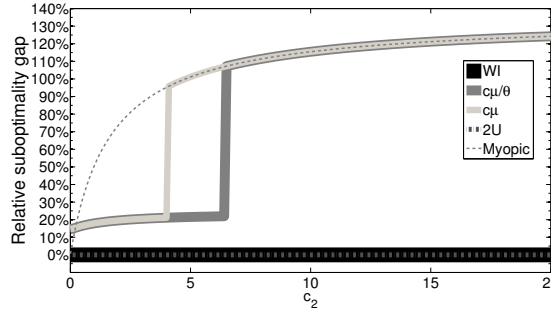


Figure 12: Multi-server relative suboptimality gap in idling Scenario 5.

## 7 Conclusion

We have investigated the problem of scheduling of customers with abandonment. We have proposed a comprehensive model accounting for completion rewards, linear waiting costs and abandonment penalties. For the problem with one or two customers in the system, we have obtained an optimal solution, which is not in the form of an index rule. For the more general case with multiple customers, we have applied Whittle’s relaxation methodology to derive the WI rule, a heuristic scheduling rule which has a very simple structure.

Numerical experiments indicate that WI performs exceptionally well: it is often optimal, and if not, then its suboptimality is small. They also show that in most cases the WI rule outperforms the alternative  $c\mu/\theta$  and  $c\mu$  rules. This is a robust result, that holds in both idling and non-idling systems, independently of system overload conditions, and for the vast majority of scenarios that we have tested. The biggest improvement of WI over  $c\mu/\theta$  and  $c\mu$  is achieved when it is sometimes optimal to idle, for instance when the abandonment rate is large relative to the service rate. In this case WI may be optimal while the  $c\mu/\theta$  can be performing as the worst policy. Important differences can also be observed when the waiting costs differ across classes. Interestingly, our scheduling rule is well-grounded, as it recovers known optimal policies in some special cases of the problem. For instance, our rule becomes the  $c\mu$ -rule if there is no abandonment. In many cases the performance of 2U is comparable to the performance of WI, however, 2U cannot be easily extended to more than two classes because it is not a simple index rule.

The performance advantage of the WI rule over the  $c\mu/\theta$  rule is primarily due to the fact that the WI rule is designed for our model, where waiting costs are paid while customers are in service as well as in queue, and abandonment does not occur during service, whereas the  $c\mu/\theta$  rule is designed for a model in which waiting costs only apply to the time in the queue. This

suggests that much care is needed in appropriately modeling and defining the objective function for a particular system.

An important question for future research is to determine under what conditions the WI rule is optimal or asymptotically optimal. Based on our numerical experiments, we believe that if the WI index values are sufficiently different across classes, then the WI rule is optimal.

The multi-class and/or multi-server problem is much more complex and time- and memory-consuming to simulate due to the curse of dimensionality. We believe that the performance of the WI rule should improve with the number of servers, as suggested by the asymptotic optimality of  $c\mu/\theta$  in overloaded multi-server systems. Moreover, the Whittle and Lagrangian relaxations apply to the multi-server case as well, and the Whittle index rule was shown to be asymptotically optimal when the number of servers grows (under certain technical conditions not satisfied by our model) in [Weber and Weiss \(1990\)](#).

The WI rule is likely to be generalizable to other distributions of service and abandonment, and non-linear waiting costs. The WI index can be interpreted as the expected profit rate, since it is given as the ratio of the expected total profit and the expected service time (or the expected abandonment time).

## Acknowledgements

Research partially supported by the French “Agence Nationale de la Recherche (ANR)” through the project ANR JCJC RACON.

## References

- Aksin, Z., Armony, M., and Mehrotra, V. (2007). The modern call center: A multi-disciplinary perspective on operations management research. *Production and Operations Management*, 16(6):665–688.
- Argon, N., Ziya, S., and Righter, R. (2010). Scheduling impatient jobs in a clearing system with insights on patient triage in mass-casualty incidents. *Probability in the Engineering and Informational Sciences*, 22(3):301–332.
- Ata, B. and Tongarlak, M. H. (2013). On scheduling a multiclass queue with abandonments under general delay costs. *Queueing Systems*, 74(1):65–104.
- Atar, R., Giat, C., and Shimkin, N. (2010). The  $c\mu/\theta$  rule for many-server queues with abandonment. *Operation Research*, 58(5):1427–1439.
- Atar, R., Giat, C., and Shimkin, N. (2011). On the asymptotic optimality of the  $c\mu/\theta$  rule under ergodic cost. *Queueing Systems*, 67(2):127–144.
- Ayesta, U., Jacko, P., and Novak, V. (2011). A nearly-optimal index rule for scheduling of users with abandonment. In *IEEE Infocom 2011*.
- Baccelli, F., Boyer, P., and Hebuterne, G. (1984). Single-server queues with impatient customers. *Advances in Applied Probability*, 16:887–905.
- Boots, N. and Tijms, H. (1999). A multiserver queueing system with impatient customers. *Management Science*, 45(3):444–448.
- Boxma, O. and de Waal, P. (1994). Multiserver queues with impatient customers. In *In Proceedings of ITC-14*, pages 743–756.
- Brandt, A. and Brandt, M. (2004). On the two-class M/M/1 system under preemptive resume and impatience of the prioritized customers. *Queueing Systems*, 47:147–168.
- Brill, P. and Posner, M. (1977). Level crossings in point processes applied to queues: single-server case. *Operations Research*, 25(4):662–574.

- Buttazzo, G. C. (2011). *Hard real-time computing systems: predictable scheduling algorithms and applications*, volume 24 of *Real-time Systems*. Springer, third edition.
- Buyukkoc, C., Varaya, P., and Walrand, J. (1985). The  $c\mu$  rule revisited. *Adv. Appl. Prob.*, 17:237–238.
- Cox, D. R. and Smith, W. L. (1961). *Queues*. Methuen & Co.
- Dai, J. and He, S. (2012). Many-server queues with customer abandonment: A survey of diffusion and fluid approximations. *Journal of Systems Science and Systems Engineering*, 21:1–36.
- Down, D., Koole, G., and Lewis, M. (2011). Dynamic control of a single server system with abandonments. *Queueing Systems*, 67:63–90.
- Fife, D. (1965). Scheduling with random arrivals and linear loss functions. *Management Science*, 11(3):429–437.
- Gans, N., Koole, G., and Mandelbaum, A. (2003). Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management*, 5(2):79–141.
- Gittins, J. (1989). *Multi-armed Bandit Allocation Indices*. Wiley, Chichester.
- Gittins, J. and Jones, D. (1974). A dynamic allocation index for the sequential design of experiments. In Gani, J., editor, *Progress in Statistics*, pages 241–266. North-Holland.
- Glazebrook, K., Ansell, P., Dunn, R., and Lumley, R. (2004). On the optimal allocation of service to impatient tasks. *Journal of Applied Probability*, 41(1):51–72.
- Graves, S. (1984). The application of queueing theory to continuous perishable inventory systems. *Management Science*, 28:401–406.
- Harrison, J. M. and Zeevi, A. (2004). Dynamic scheduling of a multiclass queue in the Halfin-Whitt heavy traffic regime. *Operations Research*, 52(2):243–257.
- Hasenbein, J. and Perry(Eds), D. (2013). Special issue on queueing systems with abandonments. *Queueing Systems*, 75(2–4):111–384.
- Hassin, R. and Haviv, M. (2003). *To Queue or not to Queue: Equilibrium Behavior in Queueing Systems*. Kluwer Academic Publishers, Boston etc.
- Iravani, F. and Balcioglu, B. (2008). On priority queues with impatient customers. *Queueing Systems*, 58:239–260.
- Jacko, P. (2009). Adaptive greedy rules for dynamic and stochastic resource capacity allocation problems. *Medium for Econometric Applications*, 17(4):10–16.
- Jouini, O., Pot, A., Koole, G., and Dallery, Y. (2010). Online scheduling policies for multiclass call centers with impatient customers. *European Journal of Operational Research*, 207(1):258–268.
- Meilijson, I. and Weiss, G. (1977). Multiple feedback at a single server station. *Stochastic Processes and Applications*, 5:195–205.
- Niño-Mora, J. (2007). Dynamic priority allocation via restless bandit marginal productivity indices. *TOP*, 15(2):161–198.
- Papadimitriou, C. and Tsitsiklis, J. (1999). The complexity of optimal queueing network. *Mathematics of Operations Research*, 24(2):293–305.
- Puterman, M. L. (2005). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.
- Sevcik, K. (1974). Scheduling for minimum total loss using service time distributions. *Journal of the ACM*, 21:66–75.
- Smith, W. (1956). Various optimizers for single stage production. *Naval Res. Logist. Quart.*, 3:59–66.
- Weber, R. and Weiss, G. (1990). On an index policy for restless bandits. *Journal of Applied Probability*, (27):637–648.
- Whitt, W. (2004). Efficiency-driven heavy-traffic approximations for many-server queues with abandonments. *Management Science*, 50:1449–1461.
- Whittle, P. (1981). Arm-acquiring bandits. *Annals of Probability*, 9(2):284–292.
- Whittle, P. (1988). Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, 25:287–298.

## A Proof of Proposition 2

**Proof.** It is straightforward to obtain that the expected total revenue if serving the customer always equals

$$-\frac{c'_k}{1 - \beta + \beta\mu'_k},$$

and that the expected total revenue if not serving her at all equals

$$-\frac{c'_k + \beta d_k \theta'_k}{1 - \beta + \beta\theta'_k}.$$

By taking the difference and rearranging we obtain

$$\frac{c'_k}{1 - \beta + \beta\theta'_k} + \frac{\beta d_k \theta'_k}{1 - \beta + \beta\theta'_k} - \frac{c'_k}{1 - \beta + \beta\mu'_k},$$

which can be further rewritten as

$$\beta \frac{c'_k(\mu'_k - \theta'_k) + d_k \theta'_k(1 - \beta + \beta\mu'_k)}{(1 - \beta + \beta\theta'_k)(1 - \beta + \beta\mu'_k)} = C_k. \quad \blacksquare$$

## B Proof of Proposition 3

**Proof.** There are only two possible states of the system: (0) meaning that the system is empty, and (1), meaning that the customer is in the system. For this particular case, from (2) we get  $V(0) = 0$ , and

$$V(1) = \max\{-c'_1 + \beta(1 - \mu'_1)V(1); \\ -c'_1 - \beta d_1 \theta'_1 + \beta(1 - \theta'_1)V(1)\},$$

where the first term refers to serving the customer, and the second term to serve the alternative task. It follows that serving the customer is better than or equivalent to serving the alternative task if and only if

$$-V(1)(\mu'_1 - \theta'_1) + d_1 \theta'_1 \geq 0. \quad (19)$$

We solve the Bellman equation and evaluate the value function assuming that serving the customer is optimal, and we get

$$V(1) = \frac{-c'_1}{1 - \beta + \beta\mu'_1}. \quad (20)$$

Substituting (20) into (19) yields (i).

Claim (ii) is obtained analogously. The only difference is that now serving the alternative task is optimal, and thus the value function is equal to:

$$V(1) = \frac{-c'_1 - \beta d_1 \theta'_1}{1 - \beta + \beta\theta'_1}. \quad \blacksquare$$

## C Proof of Proposition 4

**Proof.** Since  $\kappa' = -\infty$ , it will never be optimal to engage the alternative task (as long as there is at least one customer in the system), and thus it will have no effect on the characterization of the optimal policy. Further, we can assume that the problem is over once the system is empty, i.e.  $V(0, 0) = 0$ . Since there are two customers, equation (2) becomes

$$\begin{aligned} V(1, 1) = & \max\{-c'_1 - c'_2 + \mu'_1 V(0, 1) + \theta'_2(-d_2 + V(1, 0)) \\ & + (1 - \mu'_1 - \theta'_2)V(1, 1); \\ & -c'_1 - c'_2 + \theta'_1(-d_1 + V(0, 1)) + \mu'_2 V(1, 0) \\ & + (1 - \theta'_1 - \mu'_2)V(1, 1)\} \end{aligned} \quad (21)$$

In states in which there is only one customer present, that is  $(1, 0)$  and  $(0, 1)$ , the optimal action is to serve the customer present because the alternative task reward  $\kappa' = -\infty$  and  $C_k \geq 0$ . From Bellman's equation we directly obtain that  $V(1, 0) = -c'_1/\mu'_1$  and  $V(0, 1) = -c'_2/\mu'_2$ , as in the proof of Proposition 3. Substituting these equalities in the above expression we get

$$\begin{aligned} 0 = -c'_1 - c'_2 + \max\left\{ & -\mu'_1 \frac{c'_2}{\mu'_2} - \theta'_2 \left(d_2 + \frac{c'_1}{\mu'_1}\right) \right. \\ & - (\mu'_1 + \theta'_2)V(1, 1); \\ & -\theta'_1 \left(d_1 + \frac{c'_2}{\mu'_2}\right) - \mu'_2 \frac{c'_1}{\mu'_1} \\ & \left. - (\theta'_1 + \mu'_2)V(1, 1)\right\} \end{aligned}$$

When serving customer 1 is optimal, then the first term within the brackets is equal to the left-hand side and greater than or equal to the second term. It is straightforward to obtain that this is equivalent to

$$\frac{c'_1(\mu'_1 - \theta'_1) + d_1\theta'_1\mu'_1}{\mu'_1(\theta'_1 + \mu'_2)} \geq \frac{c'_2(\mu'_2 - \theta'_2) + d_2\theta'_2\mu'_2}{\mu'_2(\theta'_2 + \mu'_1)}.$$

We have thus obtained that the index value  $\nu_k^{2U}$  given in (6) characterizes which customer it is optimal to serve. ■

## D Proof of Theorem 1

**Proof.** The proof of this theorem is based on establishing indexability of the problem and computing the index values following the survey Niño-Mora (2007). We will rewrite our minimization problem as a maximization one. Indexability is in fact equivalent to existence of the quantities with stated properties, and is valid because any binary-state MDP is indexable.

Let us denote the optimal value function by  $V_{k,n}$  for customer  $k$  in state  $n$ . The Bellman equation for state 1 and customer  $k \in \mathcal{K}$ , after plugging in the definitions of the action-dependent parameters for a state becomes:

$$\begin{aligned} V_{k,1} = & \max\{R_{k,1}^1 - \nu W_{k,1}^1 + \beta[\mu'_k V_{k,0} + (1 - \mu'_k)V_{k,1}]; \\ & R_{k,1}^0 - \nu W_{k,1}^0 + \beta[\theta'_k V_{k,0} + (1 - \theta'_k)V_{k,1}]\}, \end{aligned}$$



After plugging in the formulas for expected one-period revenues and expected one-period capacity consumption, we obtain:

$$V_{k,1} = \max\{-c'_k - \nu + \beta[\mu'_k V_{k,0} + (1 - \mu'_k)V_{k,1}]; \quad (22)$$

$$-c'_k - \beta d_k \theta'_k + \beta[\theta'_k V_{k,0} + (1 - \theta'_k)V_{k,1}]\},$$

where the first term in the curly brackets corresponds to serving and the second term corresponds to not serving the customer. Note that the Bellman equation for  $V_{k,0}$  is:

$$V_{k,0} = \max\{R_{k,0}^1 - \nu W_{k,0}^1 + \beta V_{k,0}; R_{k,0}^0 - \nu W_{k,0}^0 + \beta V_{k,0}\}$$

$$= \max\{-\nu + \beta V_{k,0}; \beta V_{k,0}\}.$$

For  $\nu \geq 0$ , it is straightforward to obtain that  $V_{k,0} = 0$ . Analogously for  $\nu < 0$  we obtain:

$$V_{k,0} = -\frac{\nu}{1 - \beta}.$$

Statement (ii) follows immediately. Next we prove (i), dividing the proof into two cases.

**Case  $\nu \geq 0$**  In statement (i), we want to show when it is optimal and when it is not optimal to serve waiting customer  $k$ . We derive the value of  $V_{k,1}$ . When we choose to serve the customer, we get:

$$V_{k,1} = -c'_k - \nu + \beta[\mu'_k V_{k,0} + (1 - \mu'_k)V_{k,1}],$$

and it is straightforward to obtain:

$$V_{k,1} = -\frac{c'_k + \nu}{1 - \beta + \beta\mu'_k}.$$

We substitute the expressions  $V_{k,0}$  and  $V_{k,1}$  in (22) to obtain, after rearranging, a condition for serving customer  $k$ :

$$\beta \frac{c'_k(\mu'_k - \theta'_k) + d_k \theta'_k(1 - \beta + \beta\mu'_k)}{1 - \beta + \beta\mu'_k} \geq \nu.$$

Since  $\nu \geq 0$ , this is equivalent to  $\nu_{k,1} \geq \nu$ .

Analogously to above we can show that if it is optimal not to serve waiting customer  $k$ , then  $\nu_{k,1} \leq \nu$ .

**Case  $\nu < 0$**  We can similarly prove that statement (i) also holds if  $\nu < 0$ . All the steps will be similar, but note that we now have  $V_{k,0} = -\frac{\nu}{1 - \beta}$ .

If serving is optimal, from equation (22), we have

$$V_{k,1} = \frac{-c'_k - \nu - \beta\mu'_k \frac{\nu}{1 - \beta}}{1 - \beta + \beta\mu'_k}. \quad (23)$$

We next substitute the expressions of  $V_{k,0}$  and  $V_{k,1}$  in the equation (22). We want to derive a condition to determine when it is optimal to serve. I.e., when the first term in (22) is greater or equal to the second. After rearranging, we obtain:

$$\nu \leq \beta \frac{c'_k(\mu'_k - \theta'_k) + d_k \theta'_k(1 - \beta + \beta\mu'_k)}{1 - \beta + \beta\mu'_k}.$$

Since  $\nu < 0$ , this is equivalent to  $\nu_{k,1} \geq \nu$ .

**Statement (iii)** The proof for the last statement can be done formally, but can better be seen in an intuitive way. We know that  $\nu$  is a cost and  $\nu_{k,0} = \kappa'$  is a reward for using a server during one period. It is obvious that when  $\nu \leq \nu_{k,0}$ , we choose to allocate the server to the alternative task  $k$ . Analogously, when  $\nu \geq \nu_{k,0}$ , we choose not to allocate the server to the alternative task  $k$ . ■