https://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH

Vol. 00, No. 0, Xxxxx 0000, pp. 000–000

Submitted to *Operations Research*

ISSN 0030-364X, EISSN 1526-5463

Improving Upon the generalized $c\mu$ rule: a Whittle approach

Zhouzi Li, Keerthana Gurushankar, Mor Harchol-Balter
Computer Science Department, Carnegie Mellon University, zhouzil@cs.cmu.edu, kgurusha@cs.cmu.edu, harchol@cs.cmu.edu

Alan Scheller-Wolf
Tepper School of Business, Carnegie Mellon University, awolf@andrew.cmu.edu

Abstract. Scheduling a stream of jobs whose holding cost changes over time is a classic and practical problem. Specifically, each job is associated with a holding cost (penalty), where a job's instantaneous holding cost is some non-decreasing function of its class and current age (the time it has spent in the system since its arrival). The goal is to schedule the jobs to minimize the time-average total holding cost across all jobs.

The seminal paper on this problem, by Van Mieghem in 1995, introduced the generalized c-mu rule for scheduling jobs. Since then, this problem has attracted significant interest but remains challenging due to the absence of a finite-dimensional state space formulation. Consequently, subsequent works focus on more tractable versions of this problem.

This paper returns to the original problem for a k-class M/M/1 system. We derive a heuristic that empirically improves upon the generalized c-mu rule and all existing heuristics. Our key idea is to first translate the holding cost minimization problem to a novel Restless Multi-Armed Bandit (R-MAB) problem with a finite number of arms, where each arm's state corresponds to the age of the oldest job in one class. Based on our R-MAB, we next derive a novel Whittle Index policy, which is both elegant and intuitive.

Funding: This research was supported by NSF-CIF-2403194, NSF-III-2322973, and NSF-CMMI-2307008.

Key words: generalized $c\mu$ rule, dynamic scheduling, holding cost minimization, Whittle Index, restless Multi-Armed Bandit, index policy, convex delay-based holding cost

1. Introduction

Since the seminal paper by Van Mieghem (1995), the problem of scheduling jobs with Time-Varying Holding Cost (the TVHC problem) has been an important topic in the operations literature. The

TVHC problem assumes a single-server multi-class system with k classes of jobs, where each job incurs a (time-varying) holding cost for every unit of time it remains in the system, and the goal of the scheduling policy is to minimize the time-average total holding cost across all jobs. Specifically, define the *age* of a job to be the time it has spent in the system since it arrived. For a job of class i, let $c_i(t)$ be the instantaneous holding cost when the job's age is t. We allow different classes to have different holding-cost functions (see Figure 1), but assume these functions are non-decreasing. Note that non-decreasing instantaneous holding cost is equivalent to convex accumulated holding cost, which is assumed in Van Mieghem (1995) and all its follow-on works. Also, throughout this paper, we assume that job sizes are exponentially distributed unless otherwise specified, and we assume that the job arrival process is Poisson. Let λ_i and μ_i, respectively, denote the arrival rate and the completion rate of class i jobs.

In the special case when the holding cost of each class is a constant function (where $c_i(t) = c_i$), the optimal policy is the famous $c\mu$ rule, which always (preemptively) runs the job with the highest product $c_i \cdot \mu_i$ (Cox (2020), Buyukkoc et al. (1985)). However, in the general case, this problem is much more complicated and the optimal policy is not known. In Van Mieghem (1995), an analogue of the $c\mu$ rule is proposed, which is now commonly referred to as the generalized $c\mu$ rule: The priority of a job is given by the product of its instantaneous holding cost, $c_i(t)$, and its instantaneous failure rate $\mu_i(t)$. We discuss the generalized $c\mu$ rule in Section 1.1.

Note that both the generalized $c\mu$ rule and the $c\mu$ rule are index policies: Each job has an index (in both cases the value $c_i(t)\mu_i(t)$) which is a function of only the job's state, and the policy always serves that job with the highest index. Index policies are both simple and powerful. This paper aims to solve the TVHC problem within the class of preemptive index policies, yielding a heuristic solution to the problem in general. We now review the literature, and motivate our approach.

1.1. The generalized $c\mu$ rule

The generalized $c\mu$-rule has been shown to be asymptotically optimal for M/G/1 queues in the diffusion limit regime (Van Mieghem (1995)), where both the arrival and service rates go to infinity (and the total load goes to 1). Under the same diffusion limit, its analogues have also been shown to be asymptotically optimal for more complicated settings such as the multi-server setting and systems with abandonment (Mandelbaum and Stolyar (2004), Atar et al. (2010), Long et al. (2020)). However, outside of the diffusion limit regime, it is known that the generalized $c\mu$-rule can perform poorly. Here we give an example to illustrate this.

Consider a system with two classes of jobs: one with deadlines and one without. For deadline-based jobs, the holding cost is zero before their deadline but becomes significantly higher once the deadline is passed, while jobs without deadlines have a constant holding cost (see Figure 2). Assume that both classes have job sizes following the same Exp(μ) distribution. Under the generalized $c\mu$ rule, the system grants priority to deadline-based jobs only after they have missed their deadlines. Intuitively, however, it might make more sense to prioritize these jobs before their deadline is reached. Simulation results show that by prioritizing the deadline-based jobs before their deadline, the overall holding cost can be substantially reduced in normal-traffic scenarios (see Figure 8b).

Thus, the generalized $c\mu$ rule can be highly suboptimal under normal traffic (even with exponential job sizes), underscoring the need for a better scheduling policy.

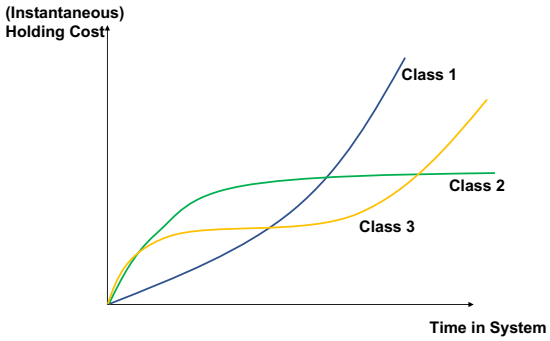


Figure 1 Classes with different holding costs.

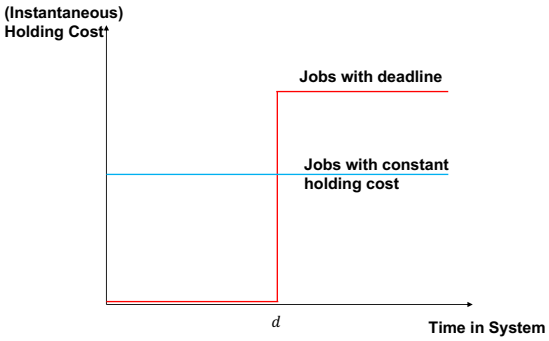


Figure 2 Example: generalized $c\mu$ rule is suboptimal.

1.2. Prior attempts to improve upon the generalized $c\mu$ rule

One of the difficulties in solving the TVHC problem lies in the fact that a job's holding cost depends on its age. Hence, we need to track the ages of all jobs in the system, which requires an infinite-dimensional state space. As a result, the existing literature has considered more tractable versions of the TVHC problem that have a finite-dimensional state space. Such a finite-dimensional state space allows the authors to represent their problem as a finite-dimensional Markov Decision Problem (MDP), or, more specifically, a Restless Multi-Armed Bandit problem (R-MAB) with a finite number of arms. We provide a brief tutorial on R-MABs in Section 2.

One way to create a finite-dimensional state space for the TVHC problem is to assume a static setting, where all n jobs are present at time 0 and there are no new arrivals. This is the approach taken in Anand and de Veciana (2018), Aalto (2024, 2026). By limiting the number of jobs, the problem can now be translated to an n-arm R-MAB. From this R-MAB the authors then derive a Whittle

index, which determines which job to run at every moment in time (see Electronic Companions, EC. 1 for a tutorial on the Whittle index). Figure 3 shows a road-map of the solution. The drawback of the static version setting is that the arrival rate (and consequently the load) of each class cannot be incorporated in the policy. This is unfortunate, because, for example, in Figure 2, it is reasonable to expect that if load is higher, we might want to start working on the deadline-oriented jobs sooner than we would under lower load. [1]

Figure 3 A road-map to derive the Whittle Index policy for the static version of our problem: The static version is first translated to a R-MAB problem. Then the Whittle Index is derived based on the R-MAB problem.

A finite-dimensional state space also exists for a complementary setting of the TVHC problem where individual jobs no longer have a holding cost. Instead, the total holding cost for class i is a function of the number of class i jobs present, see Gurvich and Whitt (2009), Atar et al. (2004), Bispo (2013), Larrañaga et al. (2016), Ansell et al. (2003), Larrañaga et al. (2014), Glazebrook et al. (2003). With this change, one only needs to track the number of jobs within each class, enabling the problem to be translated into a finite-dimensional state space MDP or a k-arm R-MAB. Several papers (Ansell et al. (2003), Larrañaga et al. (2014), Glazebrook et al. (2003)) next follow the road-map in Figure 4 where they use the k-arm R-MAB to derive a Whittle index policy. This queue-length holding cost setting is complementary to our age-based holding cost setting, as the policy derived for one setting cannot be translated into a policy in the other setting.

Figure 4 A road-map used to derive the Whittle Index policy for the queue-length holding cost setting.

1.3. Our approach and contributions

The Whittle Index policy is known to be a good heuristic for R-MAB problems (see Niño-Mora (2007) for a discussion). Like the prior work of Figures 3 and 4, we adopt a similar road-map to derive the Whittle Index for our age-based holding cost minimization problem. Unfortunately, in

[1] In the static case, Aalto (2026) generalizes their previous work (Aalto (2024)) and is able to derive the Whittle index for general job size distributions.

our problem we have an infinite-dimensional state space which makes things much harder.[2] First, while the translation to a tractable R-MAB is straightforward in prior works, it is hard to reduce our problem to a finite-arm R-MAB. Second, as in all the prior works, even after reducing to an R-MAB problem, we still need to derive the Whittle Index from the R-MAB, which requires overcoming two obstacles: *(i)* establishing indexability, a key property of the R-MAB ensuring that the Whittle Index is well-defined; and *(ii)* performing the actual derivation of the Whittle index, which can be intricate.

In overcoming these difficulties, our paper makes two primary contributions, outlined in Figure 5: First, Theorem 1 (see Section 3) translates the TVHC problem to a novel R-MAB problem. Second, Theorem 2 (see Section 4, 5 and 6) proves indexability and derives the Whittle Index for our R-MAB problem, thus yielding a Whittle-based heuristic policy for our problem.

Figure 5 Using Theorem 1 and Theorem 2, we follow this road-map to derive the Whittle Index policy.

Our final result is a Whittle-based policy which always preemptively runs the job with the highest Whittle Index. The Whittle Index has an elegant and intuitive formulation given in Corollary 1 (see Section 6) and repeated here: Our Whittle Index for a class i job with age t, $W_i(t)$, is given by

$$W_i(t) = \mu_i \cdot \mathbf{E}\left[c_i(t+X)\right], \quad X \sim \text{Exp}(\mu_i - \lambda_i). \tag{1}$$

Intuitively, while the generalized $c\mu$ rule focuses on the current holding cost $c_i(t)$ and schedules according to the index $\mu_i c_i(t)$, our Whittle index looks a little further into the future to age $t+X$. Note that if the load is high (λ_i is close to μ_i), then we look further into the future. This aligns with the motivating example in Figure 2, where it is preferable to prioritize a job before its deadline, particularly when there are likely to be many jobs in the system.

It is also interesting to contrast our policy with the heuristic given in Aalto (2024), which is the Whittle index in the static version. Under exponential job sizes, their index has the form $\mu_i \cdot \mathbf{E}\left[c_i(t+Y)\right]$, where $Y \sim \text{Exp}(\mu_i)$. Note that our index given in (1) matches theirs when $\lambda_i = 0$, where our setting degenerates to the static setting. In Section 7 we compare our policy with all the existing heuristics (including Aalto (2024)). Our simulations show that our policy outperforms all the other heuristics.

[2] In fact, both papers on the static version (Anand and de Veciana (2018), Aalto (2024)) note the analytical intractability of the dynamic age-based TVHC setting.

2. Tutorial on (Markov) R-MAB Problem

In this paper, when we say MAB, we always refer to the Markov MAB: In a Markov MAB problem, each arm corresponds to a Markov process with states that evolve based on an underlying state transition model. The agent chooses to pull an arm at each time step, incurring a cost determined by the states of the arms. The objective is to minimize the cumulative cost over time.

Mathematically, let k denote the number of arms. Each arm i is associated with a Markov Decision Process (MDP), which is defined by a state space, the action space $\{active, passive\}$, the transition probabilities given states and actions, and a cost function $c_i(s)$ representing the cost incurred when arm i is in state s. At each time step t, the agent observes the current states of all arms and selects at most one arm to pull (pulling means choosing the action $active$). Cost is incurred for each arm, and the state of each arm evolves according to its respective MDP and actions. The objective is to minimize the long-run average cost over an infinite time horizon.

A discounted MAB is defined exactly the same except for the objective. Instead of the long-run average cost, a discounted MAB aims to minimize the cumulative discounted cost, where the cost at time t is multiplied by $e^{-\alpha t}$, and $\alpha > 0$ is called the discount factor.

A MAB is called restful if the state of an arm does not evolve when the arm is not pulled (i.e., when the action is $passive$). In contrast, an MAB is called restless if the state evolves whether the arm is pulled or not. For a restful MAB, the optimal policy is the Gittins index policy (Gittins (1979), Scully and Harchol-Balter (2021)), while the optimal policy for restless MAB (R-MAB) is open. For R-MAB, the Whittle Index policy is known to be a good heuristic (Whittle (1988), see EC. 1 in the Electronic Companion for a tutorial). The TVHC problem is intrinsically restless.

3. The TVHC problem and its translation into an R-MAB problem

The main goal of this section is to complete the first step in the road-map (Figure 5), which is the translation from the TVHC problem to a discounted R-MAB problem. First in Section 3.1, we restate the TVHC problem setting, and prove that one should serve jobs within each class in FCFS order. This fact allows us to only consider index functions that enforce FCFS within each class, which we will see is essential for the translation to an R-MAB problem. Second in Section 3.2, we prove our theorem which translates the TVHC problem to an R-MAB problem. Finally in Section 3.3, a discount factor is introduced to the R-MAB problem. The discounted R-MAB problem serves as the starting point of the typical Whittle Index approach. The Whittle Index for the R-MAB without discounting is obtained by taking the limit on the discount factor (see Section 3.3).

3.1. The TVHC problem

In this subsection, we first formally define the TVHC problem, and then clarify our focus on index policies.

Problem setting: We briefly restate the TVHC problem setting as follows: In a single-server system, there are k types of jobs. For each type i, we assume a Poisson arrival process with rate λ_i. We also assume that the job size distribution is exponentially distributed with rate μ_i. For a type i job of age $t \geq 0$, its instantaneous holding cost is $c_i(t)$, which is a non-decreasing and smooth[3] function. We define c_i' to be the right-hand derivative of the holding cost function c_i, i.e., $c_i'(t) := \lim_{\delta \to 0^+} \frac{c_i(t+\delta)-c_i(t)}{\delta}$. We extend the definition of c_i and c_i' to include negative value ranges by defining $c_i(t) = c_i'(t) = 0$ for any $t < 0$.

The objective of the scheduler is to schedule the jobs in order to minimize the time-average mean holding cost. Specifically, Let $c_{total}(s)$ denote the sum of the holding costs of all jobs currently in the system at time s. Then the objective is to minimize $\mathbf{E}$ [Holding Cost], where

$$\mathbf{E}\left[\text{Holding Cost}\right] = \lim_{t\to\infty} \frac{1}{t} \int_{s=0}^{t} c_{total}(s) ds.$$

To make the problem well-defined, we assume that there exists a policy to make the time-average mean holding cost converge.

Index functions: We only focus on index policies in this paper. These are policies where a job's index (priority level) only depends on its own state, not on the state of other jobs in the system. Specifically, since the job sizes are exponential, a job's index only depends on the job's type and age and does not depend on its attained service. Thus an index policy is specified by a set of functions $\{U_i(\cdot)\}_{i=1}^k$, where $U_i(t)$ is the index of a job of type i with age t.

FCFS within each class: Intuitively, since the holding cost is non-decreasing and the job sizes are exponentially distributed, to minimize mean holding cost, we should schedule jobs within each type in FCFS (First Come First Serve) order, since earlier arriving jobs have higher cost. Mathematically, we have the following lemma:

LEMMA 3.1 **(FCFS within each type is optimal)**. *The optimal policy must serve jobs within each type in FCFS order.*

Proof See Electronic Companions, section EC.2.

[3] We need smoothness to simplify our derivation. However, our policy applies to any holding cost function that can be arbitrarily closely approximated by a smooth function (e.g., a deadline function, or any continuous function).

Motivated by Lemma 3.1, throughout this paper, we focus on index policies that enforce FCFS order within each class. This is equivalent to having a non-decreasing index function for each class with a FCFS tie-breaking rule. In Section 3.2, we will leverage the fact that the index functions enforce FCFS within each class to substantially reduce the dimensionality of the state space.

3.2. The corresponding R-MAB problem (Theorem 1)

In this section, we translate the TVHC problem into an R-MAB problem. Intuitively, our key idea is to let each arm of the R-MAB problem track the age of the oldest job within each class in the TVHC problem. At first this seems insufficent, because we're not capturing the state of all the other jobs within each class. However we will show how the FCFS ordering within each class ensures that the distribution of the ages of younger jobs can be effectively captured through the stochastic behavior of the Poisson arrival process. Our construction of the R-MAB problem is as follows:

DEFINITION 3.1 (OUR R-MAB). Our proposed R-MAB has k arms, each representing a class. There are two actions for each arm (active or passive), where arm i is active means the oldest class i job is served, and passive means the job served at this moment is of some other type. The arm states, transition probability, constraint, cost functions and objective are defined below:

- Arm state $A_i(t)$: The ith arm's state is $A_i(t) \in \mathbb{R}$. The arm state can be interpreted in the TVHC problem as the age of the oldest type i job at time t. If there is no type i job in the system, $A_i(t)$ is negative, which means the next type i arrival happens $(-A_i(t))$ time later. Note that for arm i, the action active is only allowed at time t when $A_i(t)$ is positive. [4]

- Transition Probability: If the action for arm i is passive, the ith arm state grows with rate 1. Otherwise if arm i is active, the ith arm state may drop an $\text{Exp}(\lambda_i)$ amount according to a Poisson process (when completions happen). Mathematically, the transition function is

$$\text{passive: } dA_i(t) = dt,$$
$$\text{active: } dA_i(t) = dt - I \cdot dN_{\mu_i}(t),$$

where $I \sim \text{Exp}(\lambda_i)$, and N_{μ_i} is a Poisson counting process with rate μ_i.

[4] We restrict the active action in negative states because, in the original scheduling problem, those states correspond to situations in which no class-i job is present, so taking the active action is not meaningful. Similar restrictions on the action space for certain states also appear in Ansell et al. (2003), Glazebrook et al. (2003), Larrañaga et al. (2014).

This transition function can be interpreted in the TVHC problem as follows: A passive action means that the oldest class i job is not in service and its age grows with rate 1. If the action is active, then in the next dt time period, the oldest class i job is served. There is a probability of $\mu_i dt$ that the job is completed and leaves the system, in which case the oldest class i job in the system becomes the previously second-oldest class i job. Since the inter-arrival time I follows the distribution $\text{Exp}(\lambda_i)$, the age of the oldest job drops by an $\text{Exp}(\lambda_i)$ amount.

- Constraint: The number of active arms at any time is at most 1.
- Cost Function $r_i(s)$: Arm i incurs a cost of $r_i(s)$ at state s, where $r_i(s)$ is defined to be

$$r_i(s) := c_i(s) + \mathbf{E}\left[\sum_{j=1}^{N_i} c_i(Y_j)\right], \tag{2}$$

where $Y_j = \sum_{m=1}^{j} I_m$, $I_m \sim \text{Exp}(\lambda_i)$, and N_i is the random variable such that $Y_{N_i} < s$ and $Y_{N_i+1} \geq s$. Thus $N_i \sim Pois(\lambda_i \cdot s)$. For $s < 0$, define $r_i(s) = 0$.

To interpret (2), observe that $r_i(s)$ represents the expected total instantaneous holding cost of all class i jobs given that the age of the oldest class i job is s: Since the index policy serves class i jobs in FCFS order, no "young" class i jobs (class i jobs younger than the current oldest one) have been completed. Since the inter-arrival times are distributed as $\text{Exp}(\lambda_i)$, their ages are distributed as

$$s - I_1,\ s - I_1 - I_2,\ \ldots,\ s - \sum_{m=1}^{N_i} I_m.$$

Thus the expected total instantaneous cost of all young class i jobs is $\mathbf{E}\left[\sum_{j=1}^{N_i} c_i(s - Y_j)\right]$, which is equal to $\mathbf{E}\left[\sum_{j=1}^{N_i} c_i(Y_j)\right]$ as the N_i arrivals are distributed as uniform order statistics in $(0, s)$.

- Objective: The objective is to minimize the long-run expected cost. Mathematically,

$$\text{Cost} = \mathbf{E}\left[\lim_{x\to\infty} \frac{1}{x} \int_0^x \sum_{i=1}^{k} r_i(A_i(t))dt \;\middle|\; A_i(0) = 0\right].$$

In deriving the R-MAB, we utilize the property that, conditional on the oldest job having age T, the jobs remaining in the queue can be viewed as the realizations of a Poisson process on the interval $(0, T)$. This representation allows us to calculate the expected costs without needing to track explicit arrival times. This technique is loosely inspired by Stanford et al. (2014), who also make use of a similar characterization of a Poisson process to analyze a very different problem called the accumulated priority queue.

Theorem 1 (TVHC is equivalent to our R-MAB) *For any set of non-decreasing index functions* $\{U_i(\cdot)\}_{i=1}^k$*, the corresponding index policies (breaking ties by FCFS) in the TVHC problem and our R-MAB problem incur the same cost.*

Proof [Proof Sketch] Through the construction and the corresponding interpretation in the TVHC problem, it is straightforward to see a coupling between the ith arm state and the age of the oldest class i job in the TVHC problem. However, although intuitive, it is delicate to rigorously prove that characterizing the holding cost of all young jobs by expectation does not change the expected long-run mean cost. We refer to Electronic Companions, section EC. 3 for the rigorous proof, where we use a coupling argument over sample paths to build equivalence between the two problems. □

3.3. Introducing the discount factor

Finally, we introduce a discount factor into our R-MAB problem. It is a typical step before applying the Whittle Index approach (e.g. Ansell et al. (2003), Aalto (2024)). The discounted objective is defined to be:

$$\begin{aligned}&\text{Cost}(\mathbf{t_0}, \alpha)\\ &= \mathbf{E}\left[\int_0^\infty \sum_{i=1}^k r_i(A_i(s)) \cdot \alpha e^{-\alpha s} ds \;\middle|\; A_i(0) = \mathbf{t_0}(i)\right],\end{aligned} \tag{3}$$

where $\mathbf{t_0}$ is the vector of the initial states and $\mathbf{t_0}(i)$ is the initial state of arm i.

Why introducing the discount factor is necessary: The Whittle index approach considers independent single-arm sub-problems with service penalty. In general, this approach is valid for time-average cost bandits. Unfortunately, it degenerates in our problem: Since any policy making the queueing system stable is busy for ρ fraction of time, the time-average service cost for any policy is the same. Thus, to minimize the time-average total cost (with service penalty), it suffices to minimize the time-average holding cost, leading to a trivial optimal policy for the single-arm bandit problem that never stays idle if any job exists. Consequently, the Whittle index for any state and any arm is 0. Thus, the Whittle index policy, which compares the Whittle indices of all arms, is degenerate.

This problem generally exists when applying Whittle index approach to queueing problems. Similar to Aalto (2024), Ansell et al. (2003), we introduce the discount factor to make the Whittle

index non-degenerated. The discount factor makes staying idle at the early stage attractive because it saves the total service cost. Therefore, with the discount factor, the optimal policy for the single-arm problem is no longer trivial, and the obtained Whittle index is no longer 0.

Why introducing the discount factor is justified: By standard results in dynamic programming (e.g., Ansell et al. (2003), Puterman (2014)), also known as the Wiener's Tauberian theorem, we have the following lemma:

LEMMA 3.2. *For any initial state* $\mathbf{s_0}$,

$$\lim_{\alpha \to 0} \text{Cost}(\mathbf{s_0}, \alpha) = \text{Cost}\,.$$

This lemma indicates that to get a good index for our R-MAB problem (and thus the TVHC problem), it suffices to find a good index policy to optimize the discounted cost $\text{Cost}(\mathbf{s_0}, \alpha)$, and then take the limit $\alpha \to 0$ on the index.

4. A Roadmap for deriving the Whittle Index

Given the discounted R-MAB problem, we now perform the second part of the road-map of Figure 5: the derivation of the Whittle Index. The section is organized as follows: First in Section 4.1, we define the single-arm bandit formulation and define the Whittle Index. Then in Section 4.2 we propose a roadmap for deriving the Whittle Index (Theorem 2). Note that Theorem 2 gives the Whittle index of the discounted R-MAB, and by taking the limit on the discount factor, we obtain the Whittle Index for our TVHC problem in Corollary 1. The roadmap will be executed in Section 5 and Section 6.

In most parts of the remaining paper, we focus on the single-arm bandit problem, and we drop the subscript i for simplicity. However, we should remember that there is an independent single-arm bandit problem for each arm in our R-MAB problem, and the Whittle Index obtained from each single-arm bandit problem serves as the index for the corresponding arm in the R-MAB problem.

4.1. Set up the Whittle Index approach

In this section, we set up the problem following the Whittle approach (see Electronic Companions, EC. 1 for a tutorial), which relaxes the problem and decomposes it into independent single-arm bandit problems to derive a heuristic. We first define the single-arm bandit for class i where pulling the arm incurs some service penalty. Based on this formulation, the Whittle Index is defined.

Formally, we drop the subscript i and define the single-arm bandit as follows.

DEFINITION 4.1 (SINGLE-ARM BANDIT). The single-arm bandit problem has only one arm, representing a certain class. Most definitions are the same as the R-MAB in Definition 3.1, except that a service penalty is incurred when being active, and also the constraint is removed. Mathematically, the arm states, transition probability, constraint, cost functions and objective are defined below:

- State and Action: The state is denoted by $A(t)$. The action set is {active, passive}, and "active" is only allowed at time t if $A(t) \geq 0$. Throughout the paper, we always use 1 to refer to the active action and 0 to refer to the passive action.
- Transition Probability:

$$\begin{aligned} &\text{passive: } dA(t) = dt, \\ &\text{active: } dA(t) = dt - I \cdot dN_\mu(t), \end{aligned} \tag{4}$$

where $I \sim \text{Exp}(\lambda)$, and N_μ is a Poisson counting process with rate μ.

- Service Penalty ℓ: We define $\ell \in (0, \infty)$ to be the service penalty rate per unit time for an active action. It is further explained in the cost function below.
- Cost Function: At state s, if the passive action is taken, the cost incurred is $\alpha \cdot r(s)$; otherwise if the active action is taken, the cost incurred is $\alpha \cdot r(s) + \ell$; Here $r(s)$ is defined as before:

$$r(s) := c(s) + \mathbf{E}\left[\sum_{j=1}^{N} c(Y_j)\right], \tag{5}$$

where $Y_j = \sum_{m=1}^{j} I_m$, $I_m \sim \text{Exp}(\lambda)$, and N is the random variable such that $Y_N < s$ and $Y_{N+1} \geq s$. For $s < 0$, define $r(s) = 0$.

- Objective: Define $V(s_0 \mid \alpha, \ell, \pi)$ to be the discounted total cost from the initial state s_0 with discount factor α, service penalty ℓ under policy π. Our objective is to minimize the discounted cost starting from initial state $A(0) = s_0$,

$$\begin{aligned} &V(s_0 \mid \alpha, \ell, \pi) := \\ &\mathbf{E}\left[\int_0^\infty (\alpha r(A(t)) + \ell \cdot \mathbf{1}_{\text{active}}(t)) e^{-\alpha t} dt\right]. \end{aligned}$$

In the above, $\mathbf{1}_{\text{active}}$ is an indicator random variable which is 1 whenever the bandit action is 'active.' Observe that while the $r(s)$ term is multiplied by α, the ℓ term is not. This is consistent with the

methodology using the vanishing discount approach in the bandit literature, see e.g., Ansell et al. (2003), Whittle (2005). Note that if we had instead multiplied ℓ by α, we would derive a modified Whittle index function, $W_i'(t) = W_i(t) \cdot \alpha$. This modified index W' yields the same index policy as W for any $\alpha > 0$. Unfortunately, when we take $\alpha \to 0$, this W' index would degenerate to 0, though the limiting behavior of the induced scheduling policies is still that of W. Whittle (2005) provides a detailed discussion of the above subtleties.

For any value of the service penalty ℓ, consider the optimal policy of the single-arm bandit problem. We define $\Pi(\ell)$ to be the set of states where the passive action is optimal.

DEFINITION 4.2 (Π). For any service penalty ℓ, define

$$\Pi(\ell) := \{t_0 \mid \text{passive action at state } t_0 \text{ is optimal}\}.$$

Throughout, we omit writing the discount factor α in Π, to simplify notation.

Note that for any service penalty ℓ, any negative state is in $\Pi(\ell)$ (because the active action is not allowed for any negative states). Intuitively, the larger the service penalty, the more likely it is that a passive action should be taken. This property is called "indexability."

DEFINITION 4.3 (INDEXABILITY). A problem is indexable if for any $\ell_1 < \ell_2$, we have $\Pi(\ell_1) \subseteq \Pi(\ell_2)$.

The Whittle Index of a state is defined to be the smallest service penalty such that the passive action is optimal at this state. It can be intuitively understood as the value of service penalty where active and passive actions are both optimal.

DEFINITION 4.4 (WHITTLE'S INDEX). Given an indexable problem, the Whittle Index of a state t with discount factor α is defined to be

$$W(t, \alpha) := \inf_{\ell} \{t \in \Pi(\ell)\}.$$

4.2. A Roadmap towards Theorem 2

Notice that both indexability and the form of the Whittle Index involves understanding $\Pi(\ell)$, which requires us to understand the optimal policy of this single-arm bandit formulation. Thus, to prove Theorem 2, we need to characterize the optimal single-arm bandit policy.

Intuitively, it is reasonable to guess that the optimal policy is a threshold policy that chooses the passive action for states smaller than some threshold and chooses active otherwise. Mathematically, we define such a policy, Thresh(x), below:

DEFINITION 4.5 (THRESH(x)). A threshold policy, Thresh(x), parameterized by state $x \geq 0$, selects the passive action if and only if the state is smaller than x.

CLAIM 4.1. *The optimal policy for the single-arm bandit problem is a threshold policy.*

With this claim (which we prove to be true in Section 6), we suppose the optimal threshold policy is Thresh(t_0). Then by the Bellman optimality equation we see that it is indifferent at state t_0 to be active or passive. To aid in our proof, we make the following definitions.

DEFINITION 4.6 (POLICY $\langle a, \delta, \pi \rangle$). For any action a and policy π, define the policy $\langle a, \delta, \pi \rangle$ to be the policy which does action a in the first δ time, and follows policy π afterwards.

Recall that $V(s_0 \mid \alpha, \ell, \pi)$ is defined to be the discounted total cost from the initial state s_0 with discount factor α, service penalty ℓ under policy π. Then, since the optimal policy Thresh(t_0) is indifferent at state $t_0 \geq 0$ between active or passive, it must satisfy the following equation: [5]

$$\lim_{\delta \to 0} \frac{1}{\delta} \Big(V(t_0 \mid \alpha, \ell, \langle 1, \delta, \text{Thresh}(t_0) \rangle) - V(t_0 \mid \alpha, \ell, \langle 0, \delta, \text{Thresh}(t_0) \rangle) \Big) = 0. \tag{6}$$

We follow this roadmap towards proving Theorem 2: First in Section 5, we solve (6) to obtain the optimal threshold policy. Next in Section 6, we verify the Bellman optimality equation for this threshold policy. Therefore, the threshold policy turns out to be the optimal policy, based on which indexability is established and the Whittle index is derived (Theorem 2).

5. The Optimal Threshold Policy

In this section, our goal is to solve (6) for the optimal threshold policy for the single-arm bandit defined in Definition 4.1. We call the policy Thresh(t_0). The goal of this section is to get the explicit expression for t_0 (Lemma 5.8).

The section is organized as follows. First we capture the Markov Dynamics of the system state under a threshold policy in Section 5.1. Then in Section 5.2, we characterize the value function and reduce (6) and reduce it to the characterization of two M/M/1 age related terms. These two terms are derived in Section 5.3. Then, in Section 5.4 we characterize the reward function r, and finally in Section 5.5, we use all the lemmas above to obtain a closed expression for the optimal threshold policy (Lemma 5.8).

[5] At initial state t_0, the policy $\langle 1, \delta, \text{Thresh}(t_0) \rangle$ is just Thresh(t_0) for $\delta \to 0$.

5.1. Markov Dynamics of Threshold Policy

In this section, we analyze the Markov Dynamics of the state A under a threshold policy Thresh(t_0). The key lemma (Lemma 5.1) shows that the dynamics of state A under a threshold policy can be fully captured by the oldest age process in an M/M/1 queue, which we define below:

DEFINITION 5.1 ($A^{M/M/1}$). Given an initial state $A^{M/M/1}(0)$, we define $A^{M/M/1}(t)$ for any $t > 0$ by the following random process: At each time t, for an infinitely small time step δ,

- If $A^{M/M/1}(t) < 0$, then $A^{M/M/1}(t+\delta) = A^{M/M/1}(t) + \delta$;
- If $A^{M/M/1}(t) \geq 0$, then with probability $\mu\delta$, $A^{M/M/1}(t+\delta) = A^{M/M/1}(t) - \text{Exp}(\lambda)$; with probability $1-\mu\delta$, $A^{M/M/1}(t+\delta) = A^{M/M/1}(t) + \delta$.

Then we give the following lemma about the dynamics under policy Thresh(t_0).

LEMMA 5.1 (**Markov Dynamics of Thresh**(t_0)). *Under policy Thresh(t_0), the state A has exactly the same dynamics as $A^{M/M/1} + t_0$.*

Proof The proof is straightforward. At any time t, if $A(t) < t_0$, by definition of the policy Thresh(t_0), the passive action is selected, and A grows with rate 1. Otherwise if $A(t) \geq t_0$, the active action is selected, and by (4), in the next infinitely small δ time step, with probability $\mu\delta$, A drops by an exponential amount with rate λ; with probability $1-\mu\delta$, A grows with rate 1. □

We close this subsection by introducing some notation related to the M/M/1 queue.

DEFINITION 5.2 (B, I, γ_1, γ_2). Define the the length of an M/M/1 queue busy period to be B, and the length of an M/M/1 queue idle period to be $I \sim \text{Exp}(\lambda)$. Define $\gamma_1 := \mathbf{E}\left[e^{-\alpha B}\right]$, $\gamma_2 := \mathbf{E}\left[e^{-\alpha I}\right]$ to be their respective Laplace transforms at value α.

We have the following equations characterizing γ_1 and γ_2:

$$\mu = \frac{(\alpha + \lambda - \lambda\gamma_1)\gamma_1}{1-\gamma_1}, \qquad \gamma_2 = \frac{\lambda}{\lambda+\alpha}. \tag{7}$$

These equations are straightforward from classic queueing theory (e.g., see Section 27.2 in Harchol-Balter (2013)), and the proof is deferred to EC. 6 in the electronic companions.

5.2. Simplification of (6)

In this subsection, we reduce (6) to two M/M/1 age related terms, which will further be characterized in Section 5.3. The key lemma is Lemma 5.3, in which we capture the value function of the single-arm bandit problem. We start from the following lemma simplifying (6).

LEMMA 5.2. *For $t_0 \geq 0$,* (6) *is equivalent to*

$$\begin{aligned}&\mu V(t_0 \mid \alpha, \ell, Thresh(t_0))\\ &= \mu \mathbf{E}\left[V(t_0 - I \mid \alpha, \ell, Thresh(t_0))\right] + \ell.\end{aligned} \tag{8}$$

Proof Starting from state $t_0 \geq 0$, if the passive action is taken in the first δ time, by (4), we have that after the first δ time, the state is $t_0 + \delta$. During this δ time, the total cost is $\delta \cdot r(t_0) + o(\delta)$. Thus we have that

$$\begin{aligned}&V(t_0 \mid \alpha, \ell, \langle 0, \delta, \text{Thresh}(t_0)\rangle)\\ &= \delta\alpha r(t_0) + o(\delta)\\ &\quad + e^{-\delta\alpha} V(t_0 + \delta \mid \alpha, \ell, \text{Thresh}(t_0)).\end{aligned} \tag{9}$$

Starting from state t_0, if the active action is taken in the first δ time, by (4), we have that with probability $\mu\delta + o(\delta)$, a state drop happens (which means a job is completed), and the resulting state is $t_0 + \delta - I$. Otherwise if no state drop happens, the state after δ time is $t_0 + \delta$. Thus we have that

$$\begin{aligned}&V(t_0 \mid \alpha, \ell, \langle 1, \delta, \text{Thresh}(t_0)\rangle)\\ &= \delta\alpha r(t_0) + \delta\ell + o(\delta) + e^{-\delta\alpha}\Big(\\ &\quad (1 - \mu\delta) V(t_0 + \delta \mid \alpha, \ell, \text{Thresh}(t_0))\\ &\quad + \mu\delta \mathbf{E}\left[V(t_0 + \delta - I \mid \alpha, \ell, \text{Thresh}(t_0))\right]\Big).\end{aligned} \tag{10}$$

Substituting (9) and (10) into (6) yields the proof. □

Now we characterize $V(t_0 \mid \alpha, \ell, \text{Thresh}(t_0))$ and $\mathbf{E}\left[V(t_0 - I \mid \alpha, \ell, \text{Thresh}(t_0))\right]$. Two key definitions are $hold_B$ and $hold_I$, which respectively represent the total holding cost during a busy period and an idle period. See Figure 6 for an illustration.

DEFINITION 5.3 (COST IN A BUSY PERIOD $hold_B$). Let $A^{M/M/1}(0) = 0$. Let B be the first time $s > 0$ such that $A^{M/M/1}(s) < 0$. Define $hold_B(t, \alpha)$ to be the expected discounted cost incurred during the busy period where the cost function at time s is $\alpha r(t + A^{M/M/1}(s))$:

$$hold_B(t, \alpha) := \mathbf{E}\left[\int_0^B r(t + A^{M/M/1}(s))\alpha e^{-\alpha s} ds\right]. \tag{11}$$

DEFINITION 5.4 (COST IN AN IDLE PERIOD $hold_I$). Let $I \sim \text{Exp}(\lambda)$ and $A^{M/M/1}(0) = -I$. We define $hold_I(t, \alpha)$ to be the expected discounted cost during an idle period where the cost function at time s is $\alpha r(t + A^{M/M/1}(s))$:

$$\begin{aligned} hold_I(t, \alpha) &:= \mathbf{E}\left[\int_0^I r(t + A^{M/M/1}(s))\alpha e^{-\alpha s} ds\right]. \\ &= \mathbf{E}\left[\int_0^I r(t - I + s)\alpha e^{-\alpha s} ds.\right] \end{aligned} \tag{12}$$

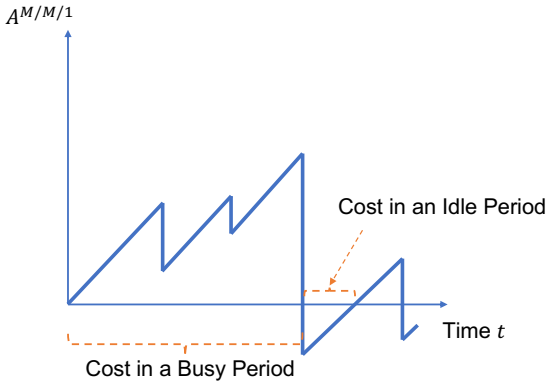


Figure 6 Illustration for Definition 5.3 and Definition 5.4.

Now we can characterize $V(t_0 \mid \alpha, \ell, \text{Thresh}(t_0))$ and $\mathbf{E}[V(t_0 - I \mid \alpha, \ell, \text{Thresh}(t_0))]$ using $hold_B$ and $hold_I$.

LEMMA 5.3. *We have the following equations: for any $t_0 \geq 0$,*

$$\begin{aligned}
&V(t_0 \mid \alpha, \ell, Thresh(t_0)) \\
&= \frac{\left(hold_B(t_0,\alpha) + \gamma_1 hold_I(t_0,\alpha) + \frac{\ell(1-\gamma_1)}{\alpha}\right)}{1-\gamma_1\gamma_2}, \\
&\mathbf{E}\left[V(t_0 - I \mid \alpha, \ell, Thresh(t_0))\right] \\
&= hold_I(t_0,\alpha) + \gamma_2 V(t_0 \mid \alpha, \ell, Thresh(t_0)).
\end{aligned}$$

Proof We start by characterizing $V(t_0 \mid \alpha, \ell, \text{Thresh}(t_0))$. We consider the state transition process of policy $\text{Thresh}(t_0)$ from the initial state t_0 until the next time the state returns to t_0. There are two phases (see Figure 7 for the illustration):

Phase 1 (the time from 0 until the first time that the state is below t_0): By Lemma 5.1, we know that the dynamics of the state is exactly the same as $A^{M/M/1} + t_0$ (with $A^{M/M/1}(0) = 0$). Since the phase ends at the first time that the state is below t_0, it is equivalent to $A^{M/M/1}(t) < 0$ when an M/M/1 busy period ends. Therefore, the length of this phase is distributed as B, and the expected total cost incurred during this time period is

$$\mathbf{E}\left[\int_0^B (\alpha r(t_0 + A^{M/M/1}(t)) + \ell) e^{-\alpha t} dt\right], \tag{13}$$

subject to $A^{M/M/1}(0) = 0$. By Definition 5.3, (13) is equal to

$$hold_B(t_0,\alpha) + \ell \cdot \mathbf{E}\left[\frac{1 - e^{-\alpha B}}{\alpha}\right] \tag{14}$$

Phase 2 (time until the state returns to t_0): At the end of phase 1, the state drops from being above t_0 to below. Since the drop is exponential with rate λ (see (4)), we know that the state at the end of phase 1 is lower than t_0 by an exponential overshoot, which is $\text{Exp}(\lambda)$. We define this state to be $t_0 - I$, where $I \sim \text{Exp}(\lambda)$.

Now since the policy $\text{Thresh}(t_0)$ stays passive when the state is smaller than t_0, phase 2 lasts for I time before the state returns to t_0 (i.e., the age grows to t_0). Thus the expected total discounted cost incurred during phase 2 is

$$\mathbf{E}\left[e^{-\alpha B} \cdot \int_0^I \alpha r(t_0 - I + t) e^{-\alpha t} dt\right]. \tag{15}$$

By Definition 5.4 and Definition 5.2, (15) is equal to

$$\gamma_1 hold_I(t,\alpha). \tag{16}$$

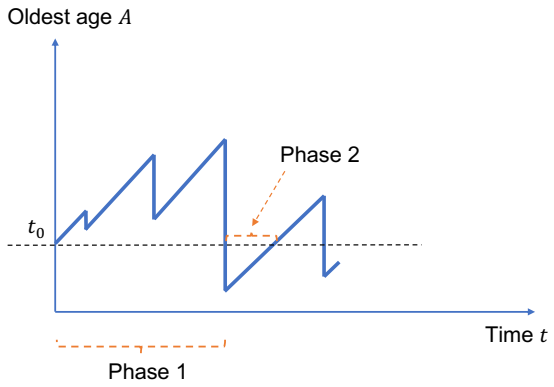


Figure 7 Illustration for Phase 1 and Phase 2.

After these two phases, the state returns to t_0, and the expected cost incurred in the future is $V(t_0 \mid \alpha, \ell, \text{Thresh}(t_0))$ times the expected decay factor after phase 1 and 2, which is $\gamma_1\gamma_2$. Thus, summing up equations (14) and (16) we have that

$$\begin{aligned}
&V(t_0 \mid \alpha, \ell, \text{Thresh}(t_0)) \\
&= hold_B(t_0,\alpha) + \gamma_1 hold_I(t_0,\alpha) + \frac{\ell(1-\gamma_1)}{\alpha} \\
&+ \gamma_1\gamma_2 V(t_0 \mid \alpha, \ell, \text{Thresh}(t_0)) \\
&= \frac{\left(hold_B(t_0,\alpha) + \gamma_1 hold_I(t_0,\alpha) + \frac{\ell(1-\gamma_1)}{\alpha}\right)}{1-\gamma_1\gamma_2}.
\end{aligned} \tag{17}$$

To characterize $\mathbf{E}\left[V(t_0 - I \mid \alpha, \ell, \text{Thresh}(t_0))\right]$, we notice that after I time (similar to phase 2), the state returns to state t_0. Thus we have that

$$\begin{aligned}
&\mathbf{E}\left[V(t_0 - I \mid \alpha, \ell, \text{Thresh}(t_0))\right] \\
&= hold_I(t_0,\alpha) + \gamma_2 V(t_0 \mid \alpha, \ell, \text{Thresh}(t_0))
\end{aligned} \tag{18}$$

Therefore we have the proof. □

5.3. Characterization of the M/M/1 age related terms

Recall that our goal in this section is to solve (6). By Lemma 5.2, we only need to characterize $\mathbf{E}\left[V(t_0 - I \mid \alpha, \ell, \text{Thresh}(t_0))\right]$ and $V(t_0 \mid \alpha, \ell, \text{Thresh}(t_0))$. Furthermore, Lemma 5.3 reduces the problem to characterizing $hold_B$ and $hold_I$. In this subsection, we obtain the closed-form expression for both terms.

We first introduce a well-known lemma for solving an ODE (e.g., see Section 2.1 in Boyce et al. (2017)).

LEMMA 5.4 (**First Order ODE**). *For the first order ODE, $y' + P(x)y = Q(x)$, the general solution has the form*

$$y = e^{-\int^x P(s)\,ds}\left(\int^x e^{\int^s P(w)\,dw} Q(s)\,ds + C\right).$$

Now we characterize $hold_B$.

LEMMA 5.5 (**Cost in a Busy Period**). *For any t and $\alpha > 0$,*

$$hold_B(t,\alpha) = (1-\gamma_1)\mathbf{E}\left[r(t+X)\right], \tag{19}$$

where $X \sim Exp\left(\frac{\alpha}{1-\gamma_1}\right)$.

Proof We consider the dynamics of $A^{M/M/1}$. During the first δ time period, a job completes with probability $\mu\delta$. If this happens, the state $A^{M/M/1}$ drops by an exponential amount, $\text{Exp}(\lambda)$. There are two cases: (1) with probability $e^{-\lambda\delta}$, the decrement is more than δ, which means $A^{M/M/1}(\delta) < 0$ and the busy period ends; (2) with probability $1 - e^{-\lambda\delta}$, the decrement amount is smaller than δ and the busy period continues. In this case, suppose $A^{M/M/1}(\delta) = \Delta < \delta$. Thus, conditioning on a drop happening in the first δ time, we have that $hold_B$ is, with a slight abuse of notation:

$$\begin{aligned}&[hold_B(t,\alpha) \mid \text{a drop happens in the first } \delta \text{ time}]\\ &= \alpha r(t)\delta + \left(1 - e^{-\lambda\delta}\right)e^{-\alpha\delta} hold_B(t+\Delta,\alpha) + o(\delta).\end{aligned} \tag{20}$$

Otherwise, with probability $1-\mu\delta$, no drop happens during the first δ time. Then $A^{M/M/1}(s)$ starts at δ, and experiences an M/M/1 busy period like process until it drops below δ. During this time, $hold_B(t+\delta,\alpha)$ is incurred. At the point that $A^{M/M/1}(s)$ drops below δ, we know that $A^{M/M/1}(s)$ is distributed as $\delta - \text{Exp}(\lambda)$ because this is an exponential overshoot. Thus with probability $e^{-\lambda\delta}$,

$A^{M/M/1}(s)$ is below 0, which means the busy period ends. Otherwise with probability $1-e^{-\lambda\delta}$, $A^{M/M/1}(s)$ is still larger than 0 (again, denoted by Δ) and the busy period continues. Thus the conditional value of $hold_B$ in this case is

$$\begin{aligned}
&[hold_B(t,\alpha) \mid \text{no drop in the first } \delta \text{ time}]\\
&=\alpha r(t)\delta + e^{-\alpha\delta} hold_B(t+\delta,\alpha)\\
&+e^{-\alpha\delta}\mathbf{E}\left[e^{-\alpha B}\right](1-e^{-\lambda\delta})hold_B(t+\Delta,\alpha)+o(\delta). \qquad (21)
\end{aligned}$$

Combining (20) and (21), we have that

$$\begin{aligned}
&hold_B(t,\alpha)\\
=&\alpha r(t)\delta+\mu\delta\left(1-e^{-\lambda\delta}\right)e^{-\alpha\delta}hold_B(t+\Delta,\alpha)\\
&+(1-\mu\delta)\Big(e^{-\alpha\delta}hold_B(t+\delta,\alpha)\\
&+e^{-\alpha\delta}\gamma_1(1-e^{-\lambda\delta})hold_B(t+\Delta,\alpha)\Big)+o(\delta).
\end{aligned}$$

Thus, multiplying both sides by -1, adding $hold_B(t+\delta,\alpha)$ to both sides, dividing by δ and taking limits, we have that

$$\begin{aligned}
&\lim_{\delta\to 0}\frac{hold_B(t+\delta,\alpha)-hold_B(t,\alpha)}{\delta}\\
=&-\alpha r(t)-\lim_{\delta\to 0}\mu\left(\left(1-e^{-\lambda\delta}\right)e^{-\alpha\delta}hold_B(t+\Delta,\alpha)\right)\\
&+\lim_{\delta\to 0}\frac{1-(1-\mu\delta)e^{-\alpha\delta}}{\delta}hold_B(t+\delta,\alpha)\\
&-\lim_{\delta\to 0}(1-\mu\delta)e^{-\alpha\delta}\gamma_1\frac{1-e^{-\lambda\delta}}{\delta}hold_B(t+\Delta,\alpha)\\
=&-\alpha r(t)+(\mu+\alpha)hold_B(t,\alpha)-\lambda\gamma_1 hold_B(t,\alpha)\\
=&-\alpha r(t)+(\mu+\alpha-\lambda\gamma_1)hold_B(t,\alpha). \qquad (22)
\end{aligned}$$

Note that by (7),

$$\alpha+\mu-\lambda\gamma_1=\alpha-\lambda\gamma_1+\frac{(\alpha-\lambda\gamma_1+\lambda)\gamma_1}{1-\gamma_1}=\frac{\alpha}{1-\gamma_1}.$$

Thus we have that (22) is equivalent to:

$$hold'_B(t,\alpha) = -\alpha r(t) + \frac{\alpha}{1-\gamma_1} hold_B(t,\alpha),$$

where $hold'_B(t,\alpha)$ is the derivative with respect to t.

Note that this is a first-order ODE of the function $hold_B(t,\alpha)$ with respect to t. By Lemma 5.4, flipping the direction of the integral, we have that

$$\begin{aligned} hold_B(t,\alpha) &= e^{\frac{\alpha}{1-\gamma_1}t}\left(\int_t^\infty e^{-\frac{\alpha}{1-\gamma_1}s}\alpha r(s)ds + C\right) \\ &= \int_0^\infty e^{-\frac{\alpha}{1-\gamma_1}s}\alpha r(t+s)ds + Ce^{\frac{\alpha}{1-\gamma_1}t} \\ &= (1-\gamma_1)\mathbf{E}\left[r(t+X)\right] + Ce^{\frac{\alpha}{1-\gamma_1}t}, \end{aligned} \tag{23}$$

where $X \sim \text{Exp}(\frac{\alpha}{1-\gamma_1})$. Next we prove that $C = 0$.

From (23) we have that for any t, $|C| \le hold_B(t,\alpha)e^{-\frac{\alpha}{1-\gamma_1}t} + (1-\gamma_1)\mathbf{E}\left[r(t+X)\right]e^{-\frac{\alpha}{1-\gamma_1}t}$.

Both terms on the right hand side go to zero as $t \to \infty$ (see Electronic Companions, section EC.4). Thus we have proven that $C = 0$. By Equation (23) we have the proof. □

We now characterize the cost in an idle period, $hold_I$.

LEMMA 5.6 (**Cost in an idle period**).

$$hold_I(t,\alpha) = (1-\gamma_2)\mathbf{E}\left[r(t-I)\right]. \tag{24}$$

Proof The proof is relatively straightforward:

$$\begin{aligned} &hold_I(t,\alpha) \\ &= \mathbf{E}\left[\int_0^I \alpha r(t-I+s)e^{-\alpha s}ds\right] \\ &= \int_0^\infty \int_0^x \alpha r(t-x+s)e^{-\alpha s}ds\ \lambda e^{-\lambda x}dx \\ &= \int_0^\infty \int_s^\infty \alpha r(t-x+s)e^{-\alpha s}\lambda e^{-\lambda x}dxds \\ &= \int_0^\infty \int_0^\infty \alpha r(t-y)e^{-\alpha s}\lambda e^{-\lambda(y+s)}dyds \\ &= \int_0^\infty r(t-y)\lambda e^{-\lambda y}dy \cdot \int_0^\infty \alpha e^{-(\alpha+\lambda)s}ds \\ &= \frac{\alpha}{\alpha+\lambda}\mathbf{E}\left[r(t-I)\right]. \end{aligned}$$

The proof can be obtained by (7). □

5.4. Characterization of the reward function r

Recall that the reward function r is defined in (5); it is one of the key ideas in the construction of our R-MAB. This "composite" reward function is introduced to capture the expected total holding cost in a class given the oldest age. The following lemma shows that this function r can be effectively captured by the holding cost function c.

LEMMA 5.7 (r'). *The right-hand derivative of r is given by*

$$r'(t) = c'(t) + \lambda c(t).$$

Proof For any $t < 0$, by definition $r'(t) = c'(t) = c(t) = 0$. Hence we only need to prove the argument for $t \geq 0$. Note that in the definition of $r(s)$ (Equation (5)), $\{Y_j\}$ can be viewed as the set of Poisson arrivals with rate λ from time 0 to time s.

A Poisson arrival process from time 0 to $t+\delta$ can be divided into two independent Poisson arrival processes: the one from time 0 to t and another one from time t to $t+\delta$. Thus we have that

$$r(t+\delta) = c(t+\delta) + \mathbf{E}\left[\sum_j c(Y_j)\right] + \mathbf{E}\left[\sum_j c(Y'_j)\right], \tag{25}$$

where $\{Y_j\}$ denotes the Poisson arrivals during $[0,t]$, and $\{Y'_j\}$ the Poisson arrivals during $[t,t+\delta]$. The expected number of arrivals during $[t,t+\delta]$ is $\lambda\delta$, and the instantaneous holding cost of those jobs is approximately $c(t)$. Thus $\mathbf{E}\left[\sum_j c(Y'_j)\right] = c(t)\cdot\lambda\delta + o(\delta)$. This together with (25) yields:

$$r'(t) = \lim_{\delta\to 0} \frac{r(t+\delta) - r(t)}{\delta} = c'(t) + \lambda c(t).$$

This finishes the proof. □

5.5. Solving the optimal threshold policy

Finally, we can solve (6) (equivalently (8)) to obtain the optimal threshold policy.

LEMMA 5.8. *Assuming the optimal policy is a threshold policy (Claim 4.1), the optimal policy $Thresh(t_0)$ must satisfy*

$$\ell = \mu\mathbf{E}\left[c(t_0+X)\right], \quad \textit{where } X \sim Exp(\frac{\alpha}{1-\gamma_1}). \tag{26}$$

Proof Substituting (17) and (18) into (8), we have that policy Thresh(t_0) must satisfy:

$$\begin{aligned}&\frac{\mu(1-\gamma_2)}{1-\gamma_1\gamma_2}\Big(hold_B(t_0,\alpha)\\&\quad+\gamma_1 hold_I(t_0,\alpha)+\frac{\ell(1-\gamma_1)}{\alpha}\Big)\\&=\mu hold_I(t_0,\alpha)+\ell.\end{aligned}\tag{27}$$

Thus, by Lemma 5.5 and Lemma 5.6 and simplification of the expressions, we have that (27) is equivalent to:

$$\begin{aligned}&\frac{\mu(1-\gamma_2)(1-\gamma_1)}{1-\gamma_1\gamma_2}\left(\mathbf{E}\left[r(t_0+X)\right]-\mathbf{E}\left[r(t_0-I)\right]\right)\\&=\ell\left(1-\frac{\mu(1-\gamma_1)(1-\gamma_2)}{\alpha(1-\gamma_1\gamma_2)}\right).\end{aligned}\tag{28}$$

The following equation can be obtained by algebraic computation, and we defer the proof to EC. 6.1 in the electronic companions: $\mathbf{E}\left[r(t_0+X)\right]-\mathbf{E}\left[r(t_0-I)\right]=\frac{\mu(1-\gamma_1)}{\gamma_1\alpha}\mathbf{E}\left[c(t_0+X)\right]$.

Using (7) to simplify the expression, we have that (28) is equivalent to $\mu\mathbf{E}\left[c(t_0+X)\right]=\ell$. □

6. Indexability and the Whittle Index

We now verify that the policy in Lemma 5.8 is not only the optimal threshold policy, but also the optimal policy. We only present the proof sketch here due to space limitations and defer the proof to the electronic companions.

LEMMA 6.1 **(Informal, Thresh(t_0) is optimal)**. *The optimal policy for the single-arm bandit under service penalty ℓ is the policy Thresh(t_0), where*

$$\mu\mathbf{E}\left[c(t_0+X)\right]=\ell,\quad X\sim Exp\left(\frac{\alpha}{1-\gamma_1}\right).$$

Proof This is proved by verifying the Hamilton–Jacobi–Bellman equation. See Electronic Companions, section EC.5. □

Based on this characterization of the optimal policy, the indexability is established and the Whittle index is derived.

Theorem 2 (Indexability) *The discounted single-arm bandit problem is indexable, and the Whittle Index is* $W(t_0, \alpha) = \mu \mathbf{E}\left[c(t_0 + X)\right]$, *where* $X \sim Exp(\frac{\alpha}{1-\gamma_1})$.

Proof See Electronic Companions, section EC.5 for the detailed proof. □

Our final result is the Whittle Index for the scheduling problem. It is a straightforward corollary of Theorem 2 by taking the limit on the discount factor and adding back the subscript i.

Corollary 1 *The Whittle Index for the scheduling problem is*

$$W_i(t_0) = \mu_i \mathbf{E}\left[c_i(t_0 + X)\right],$$

where $X \sim Exp(\mu_i - \lambda_i)$.

Proof The formula is given by $W(t_0) = \lim_{\alpha \to 0} \mu \mathbf{E}\left[c(t_0 + \mathrm{Exp}(\frac{\alpha}{1-\gamma_1}))\right]$. By the Dominated Convergence Theorem, it suffices to take the limit inside the exponential random variable: $\lim_{\alpha \to 0} \frac{\alpha}{1-\gamma_1} = \lim_{\alpha \to 0} \frac{\alpha}{1-\widetilde{B}(\alpha)} = \frac{1}{-\widetilde{B}'(0)} = \frac{1}{\mathbf{E}[B]} = \mu - \lambda$. Adding back the subscript i yields the Whittle Index. □

7. Simulations

We conduct simulations to evaluate the performance of our proposed policy (from Corollary 1), comparing it against other policies in the literature. In our evaluations, we experiment with different numbers of classes, different holding cost functions, and a range of system loads. We present the following main findings:

- Across all experiments, including cases with complex interleaving holding cost functions, our policy consistently matches the lowest time-average holding cost among all candidate policies we considered.
- In certain problem settings, our policy significantly outperforms each candidate policy.

7.1. Policies evaluated

Throughout, when we talk about "our policy," we will refer to the policy from Corollary 1 where the priority of a class i job of age t is

$$W_i(t) = \mu_i \mathbf{E}\left[c_i(t + X)\right], \text{ where } X \sim \mathrm{Exp}(\mu_i - \lambda_i).$$

We compare our policy with a broad spectrum of alternative policies:

FCFS. This policy always serves the job that arrived earliest. FCFS is a very simple policy and we compare against it as a baseline.

Strict Preemptive Priority. This policy assigns a fixed priority to each job class, where jobs from a higher-priority class have preemptive priority over those from a lower-priority class. Jobs within a class are run in FCFS order. We choose a highly optimistic version of this policy where the priority ordering can change at each value of system load. Thus we might run Prio(1;2), where class 1 has priority over class 2 when load is low, but Prio(2;1) when load is high.

Generalized cμ Van Mieghem (1995). This policy always serves the job with highest index $c_i(t) \cdot \mu_i$, where t is the age of the class i job. This policy is known to be optimal in the diffusion limit.

Aalto's Index Aalto (2024, 2026). In this policy a class i job of age t is given index $U_i(t)$, where[6]

$$U_i(t) = \mu_i \mathbf{E}\left[c_i(t+S)\right], \qquad \text{where } S \sim Exp(\mu_i).$$

Like our policy, this is also a Whittle-index based heuristic, but is motivated by the static setting (no arrivals), and hence does not incorporate the arrival rate λ_i.

Another policy which we experimented with is the Accumulated Priority policy from Stanford et al. (2014), Fajardo and Drekic (2017). However we found that this policy never improved upon our policy in our experiments and was often significantly worse than our policy, so we chose to omit it to keep the graphs cleaner.

7.2. Experimental Results

We have conducted hundreds of experiments comparing the scheduling policies from Section 7.1 under different holding cost functions, job sizes, and arrival rates. In all of these, our policy was either the best policy or matched the best policy (this excludes the pathological example in Section 8, where our policy was nearly optimal). Due to lack of space, we present only four diverse experiments which well-illustrate some important behaviors.

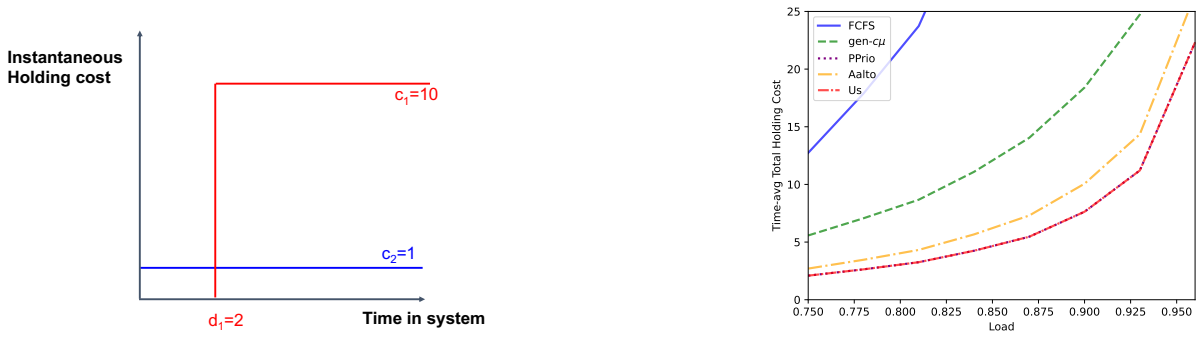
Each experiment shown is represented by (a) a set of holding cost functions (not drawn to scale) and (b) the corresponding simulation results. We experiment with small and large jobs, where holding cost functions drawn in red or orange correspond to a class with shorter jobs, while those drawn in blue correspond to a class with larger jobs. We experiment with different arrival rates, shown as a proportion of the total arrival rate, λ, across all classes, where λ is specified by the load.

Figure 8a considers an experiment where class 1 jobs (the shorter ones) only incur a holding cost after a deadline is passed, while class 2 jobs (the longer ones) incur a steady (but low) holding cost. In this figure, the arrival rate of class 1 is much higher than that of class 2. Figure 8b shows

[6] This is a simplification of Aalto's policy to the case of exponential job sizes.

the corresponding results. We see that FCFS is by far the worst policy. The generalized $c\mu$ rule improves upon FCFS, and Aalto improves upon that. Our policy (shown in red) is significantly better than all the others, except for Preemptive Priority, which is equal to our policy here.

To understand what's going on, we first observe that we should be prioritizing class 1 jobs ahead of their deadline, given that the cost of missing the deadline is so high. The $c\mu$ rule only prioritizes class 1 at the deadline point. The Aalto policy improves upon the $c\mu$ rule, by prioritizing class 1 jobs in advance of the deadline, but it doesn't do so early enough (because Aalto doesn't consider load). Our policy improves upon the Aalto policy by prioritizing class 1 jobs even earlier – in fact a better time to start prioritizing class 1 jobs is at age 0, which is what the Preemptive Priority policy does.



(a) Holding cost functions.

(b) Performance of policies.

Figure 8 Comparison of policies on holding cost functions with one deadline. We fix $\mu_1 = 3, \mu_2 = 1, \lambda_1 = 0.9\lambda$.

Figure 9a considers an experiment where both class 1 jobs (the short ones) and class 2 jobs (the long ones) only incur holding costs after their respective deadlines are passed. Here class 1 jobs incur a high penalty after a late deadline, while class 2 jobs incur a low penalty after an early deadline. The two classes have equal arrival rates. Figure 9b shows the corresponding results. We see that our policy is noticeably better than all the others, except for Aalto's policy, which we only negligibly dominate. Our policy noticeably dominates Preemptive-Priority, which noticeably dominates generalized $c\mu$ which significantly dominates FCFS.

To understand what's going on, observe that it is again useful to prioritize jobs ahead of their deadlines, which explains why we outperform FCFS and generalized $c\mu$. Strict priority is no longer effective here, because the deadlines are further out, so always prioritizing the class 1 jobs can be suboptimal. Our policy's performance is similar to Aalto for two reasons: First, because the arrival

rates are balanced, we find that when load is not too high, λ_i is small compared to μ_i, and our policy is similar to Aalto's policy. Second, while our policy looks further into the future, it does so for both classes, and hence the benefit in some sense "cancels out," leaving us with a policy similar to Aalto.

(a) Holding cost functions.

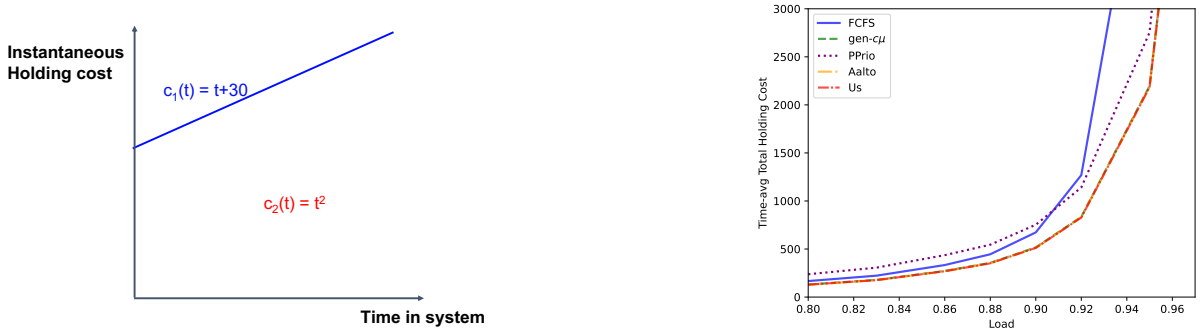
(b) Performance of policies.

Figure 9 Comparison of policies on holding cost functions with deadlines. We fix $\mu_1 = 3, \mu_2 = 1, \lambda_1 = 0.5\lambda$.

Figure 10a considers an experiment where jobs have polynomial holding cost functions. Specifically, class 1 jobs (the shorter ones) have a linear holding cost, while class 2 jobs (the longer ones) have a quadratic holding cost. We set the arrival rates such that the loads from both classes are balanced. Figure 10b shows the corresponding results. We see that our policy significantly outperforms FCFS and Preemptive-Priority; however, our policy only negligibly beats generalized $c\mu$ and Aalto.

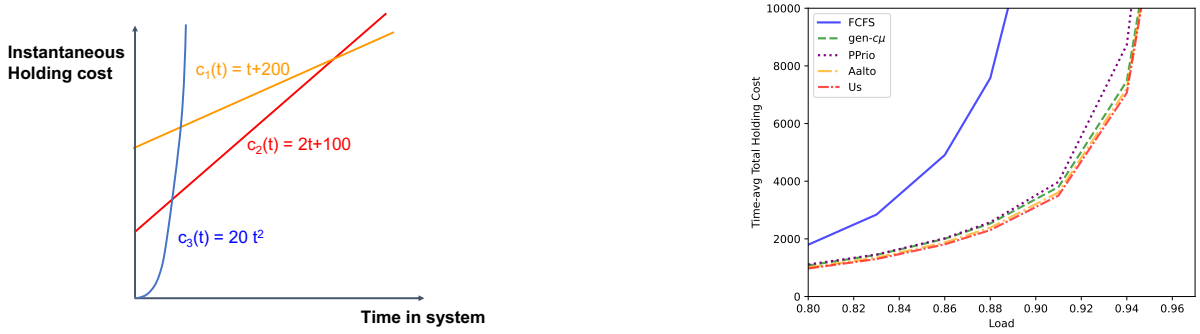
To understand why the three dominant policies (our policy, generalized $c\mu$ and Aalto) are similar, we note that, for the polynomial holding cost functions, these three policies have index functions which are polynomials with the same leading coefficient. Consequently, their behaviors are similar. We often find that these three policies perform similarly and vastly improve upon the other policies.

Lastly, Figure 11a considers the case with three job classes. Specifically, class 1 and class 2 jobs (shorter jobs) have linear interleaving holding cost functions, while class 3 jobs (longer jobs) have explosive quadratic holding cost. Figure 11b shows the corresponding results. We see that our policy visibly improves upon Aalto, though by a small margin, which visibly improves upon generalized $c\mu$, which visibly improves upon Preemptive-Priority, which significantly improves upon FCFS.



(a) Holding cost functions. (b) Performance of policies.

Figure 10 Comparison of policies on polynomial holding cost functions. We fix $\mu_1 = 3, \mu_2 = 1, \lambda_1 = 0.75\lambda$.



(a) Holding cost functions. (b) Performance of policies.

Figure 11 Comparison of policies on 3 classes. We fix $\mu_1 = \mu_2 = 3, \mu_3 = 1, \lambda_1 = \lambda_2 = \lambda_3$.

The strict ordering of policies, depicted in Figure 11b is typical of what we see in many experiments, regardless of the number of classes. This strict separation would also be more obvious in Figure 10b if we set both holding costs to be higher. When the costs are higher, the small differences between the policies are more amplified.

We also conducted experiments where the holding cost functions of different classes cross more than once. Our policy still beats all other competitors. See the Electronic Companions, EC.7, for the figure.

8. Discussion on Optimality

Our policy replicates the diffusion limit optimality of the generalized $c\mu$ rule (Van Mieghem (1995)). The diffusion limit regime in Van Mieghem (1995) is equivalent to $\lambda_i = \Theta(n), \mu_i = \Theta(n), \frac{\lambda_i}{\mu_i} =$

$1 - \Theta(\frac{1}{\sqrt{n}})$ where $n \to \infty$. Thus we have that $\mu_i - \lambda_i \to \infty$. This indicates that our policy degenerates to the generalized $c\mu$ rule, and hence is optimal as well.

We have seen that our policy performs well in simulation. However, like all the other Whittle-based heuristics we've discussed, our policy is not always optimal. We now give a counter example.

Assume that there are two classes of jobs, both with the same instantaneous holding cost function $c(t) = t$. Both classes have the same completion rate, but the arrival rates are different. Since the holding cost functions and the completion rates are the same in both classes, the optimal policy is FCFS, which is clearly not our policy. Thus, while our Whittle Index policy performs really well compared with other existing policies, there is still a need for further work on optimality.

9. Conclusion

This paper studies the classical TVHC problem: Jobs of different classes arrive over time, where each class of jobs is associated with a holding cost that increases with the job's age. The objective is to schedule the jobs so as to minimize the expected time-average total holding cost.

Various papers have provided heuristics for the TVHC problem. The generalized $c\mu$ rule (Van Mieghem (1995)) provides a simple index policy which favors jobs with currently high holding cost and high failure rate; this policy is asymptotically optimal in the diffusion limit. More recent works by Aalto (2024), consider the more tractable static setting (no arrivals).

Our work, while also heuristic, takes a principled approach to the TVHC problem with arrivals: (i) We derive the first representation of our problem as an R-MAB with a finite number of arms, and (ii) We derive a novel Whittle index policy for the resulting R-MAB. While the analysis is involved, the resulting policy is extremely simple and elegant and incorporates class load. In simulation, our policy improves upon all the other known heuristics.

This story is in no way finished. First, our policy might be generalizable beyond exponential job size distributions, to those with increasing hazard rate, using our existing R-MAB as a foundation (although this would require a more complex state space). Second, there is much more work needed to find an optimal policy.

References

Aalto S (2024) Whittle index approach to the multi-class queueing systems with convex holding costs and IHR service times. Mathematical Methods of Operations Research 100(3):603–634.

Aalto S (2026) Dynamic scheduling with convex delay costs revisited. Queueing Systems 110(1):2.

Anand A, de Veciana G (2018) A Whittle's index based approach for QoE optimization in wireless networks. Proceedings of the ACM on Measurement and Analysis of Computing Systems 2(1):1–39.

Ansell P, Glazebrook KD, Niño-Mora J, O'Keeffe M (2003) Whittle's index policy for a multi-class queueing system with convex holding costs. Mathematical Methods of Operations Research 57:21–39.

Atar R, Giat C, Shimkin N (2010) The cμ/θ rule for many-server queues with abandonment. Operations Research 58(5):1427–1439.

Atar R, Mandelbaum A, Reiman MI (2004) Scheduling a multi class queue with many exponential servers: asymptotic optimality in heavy traffic. The Annals of Applied Probability 14(3):1084–1134.

Bispo CF (2013) The single-server scheduling problem with convex costs. Queueing Systems 73:261–294.

Boyce WE, DiPrima RC, Meade DB (2017) Elementary differential equations and boundary value problems (John Wiley & Sons).

Buyukkoc C, Varaiya P, Walrand J (1985) The c-mu rule revisited. Advances in Applied Probability 17(1):237–238.

Cox DR (2020) Queues (Chapman and Hall/CRC).

Fajardo VA, Drekic S (2017) Waiting time distributions in the preemptive accumulating priority queue. Methodology and Computing in Applied Probability 19:255–284.

Gittins JC (1979) Bandit processes and dynamic allocation indices. Journal of the Royal Statistical Society Series B: Statistical Methodology 41(2):148–164.

Glazebrook KD, Lumley R, Ansell P (2003) Index heuristics for multiclass M/G/1 systems with nonpreemptive service and convex holding costs. Queueing Systems 45:81–111.

Gurvich I, Whitt W (2009) Scheduling flexible servers with convex delay costs in many-server service systems. Manufacturing & Service Operations Management 11(2):237–253.

Harchol-Balter M (2013) Performance modeling and design of computer systems: queueing theory in action (Cambridge University Press).

Larrañaga M, Ayesta U, Verloop IM (2014) Index policies for a multi-class queue with convex holding cost and abandonments. The 2014 ACM international conference on Measurement and modeling of computer systems, 125–137.

Larrañaga M, Ayesta U, Verloop IM (2016) Dynamic control of birth-and-death restless bandits: Application to resource-allocation problems. IEEE/ACM Transactions on Networking 24(6):3812–3825.

Long Z, Shimkin N, Zhang H, Zhang J (2020) Dynamic scheduling of multiclass many-server queues with abandonment: The generalized cμ/h rule. Operations Research 68(4):1218–1230.

Mandelbaum A, Stolyar AL (2004) Scheduling flexible servers with convex delay costs: Heavy-traffic optimality of the generalized cμ-rule. Operations Research 52(6):836–855.

Niño-Mora J (2007) Dynamic priority allocation via restless bandit marginal productivity indices. Transactions in Operations Research 15:161–198.

Puterman ML (2014) Markov decision processes: discrete stochastic dynamic programming (John Wiley & Sons).

Scully Z, Harchol-Balter M (2021) The Gittins policy in the M/G/1 queue. 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt 2021) (Philadelphia, PA).

Stanford DA, Taylor P, Ziedins I (2014) Waiting time distributions in the accumulating priority queue. Queueing Systems 77:297–330.

Van Mieghem JA (1995) Dynamic scheduling with convex delay costs: The generalized c-mu rule. The Annals of Applied Probability 809–833.

Whittle P (1988) Restless bandits: Activity allocation in a changing world. Journal of applied probability 25(A):287–298.

Whittle P (2005) Tax problems in the undiscounted case. Journal of Applied Probability 42(3):754–765, URL `http://dx.doi.org/10.1239/jap/1127322025`.

Zhouzi Li is a PhD student in the Computer Science Department at Carnegie Mellon University, advised by Mor Harchol-Balter. He received his B.Eng. from Yao Class (IIIS) at Tsinghua University. His research spans computer system performance and queueing theory. On the systems side, he studies scheduling in cloud GPU allocation and LLM inference serving. On the theory side, he applies bandit methods to queueing problems like parallelizable job scheduling or jobs with time varying holding costs.

Keerthana Gurushankar is a PhD student in the Computer Science Department at Carnegie Mellon University, advised by Mor Harchol-Balter. Her research focuses on stochastic scheduling, restless multi-armed bandits, and online resource allocation, with an emphasis on the design and analysis of index policies. Her prior work on optimal index policies for scheduling with convex delay costs appeared in Queueing Systems (2026). She received her B.S./M.S. in Mathematics from Carnegie Mellon University.

Mor Harchol-Balter is the Bruce J. Nelson Professor of Computer Science at Carnegie Mellon. She is the SIG Chair for ACM SIGMETRICS as well as a Fellow of both ACM and IEEE. Mor is heavily involved in the SIGMETRICS / PERFORMANCE / INFORMS research community where her papers have received over a dozen awards. She is the author of two popular textbooks, both published by Cambridge University Press: Performance Analysis and Design of Computer Systems (2013), which bridges Operations Research and Computer Science, and Introduction to Probability for Computing (2024).

Alan Scheller-Wolf is the Richard M. Cyert Professor of Operations Management at the Tepper School of Business, where he also serves as the head of the doctoral program. Alan's research

interests include inventory theory, sustainability, child-welfare operations, health care, energy, computer science, and queueing theory. He has served on the editorial boards of Management Science, Operations Research, M&SOM, and QUESTA. He has completed consulting projects with Amazon, Caterpillar, John Deere, The American Red Cross, and Allegheny County Department of Children and Families.