Probability in the Engineering and Informational Sciences, **35**, 2021, 189–209.
doi:10.1017/S0269964819000299

OPTIMAL CONTROL POLICIES FOR AN $M/M/1$ QUEUE WITH A REMOVABLE SERVER AND DYNAMIC SERVICE RATES

Pamela Badian-Pessot and Mark E. Lewis
School of Operations Research and Information Engineering, Cornell University, Ithaca, New York, United States
E-mails: plb93@cornell.edu; mark.lewis@cornell.edu

Douglas G. Down
Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada
E-mail: downd@mcmaster.ca

We consider an $M/M/1$ queue with a removable server that dynamically chooses its service rate from a set of finitely many rates. If the server is off, the system must warm up for a random, exponentially distributed amount of time, before it can begin processing jobs. We show under the average cost criterion, that work conserving policies are optimal. We then demonstrate the optimal policy can be characterized by a threshold for turning on the server and the optimal service rate increases monotonically with the number in system. Finally, we present some numerical experiments to provide insights into the practicality of having both a removable server and service rate control.

Keywords: applied probability, queueing theory, stochastic dynamic programming

1. INTRODUCTION

Much of our modern world, and in particular, large-scale computing, is built upon immense amounts of readily available energy. In fact, in 2014 data centers in the United States consumed 70 billion kWh of electricity, nearly 2% of the total US energy consumption [26]. Although the rate of energy usage relative to demand in data centers has slowed, continued improvements are needed to sustain, or ideally decrease usage levels as total demand continues to increase [9]. While large-scale computing centers boast utilization rates at 65% as compared to only 15% for small-scale centers [2], even at 65% utilization, servers spend considerable time idle. To compound this effect, servers are least energy efficient when working at low operating levels; energy-efficient servers consume as much as half their full power when idle, with less energy-efficient servers using even more power when idling [3]. Thus, identifying times when energy can be saved by turning servers off entirely is critical.

The problem of determining when to turn servers off or on is multi-layered. To begin, the savings from turning servers off must be compared to potential costs of jobs waiting longer for service. This is compounded by the fact that further delays are incurred during the time it takes to turn servers back on. Moreover, while the servers are on, managers can employ dynamic control of the service rate and thus choose to increase efficiency by using a fast service rate, albeit for a higher cost.

In this paper, we propose a Markov decision process (MDP) model that incorporates both the ability to turn the server on and off and to dynamically control the service rate in order to analyze how these controls can best be used in tandem. The analysis of this model is interesting from both a practical and theoretical point of view. While optimal policies, those attaining minimal long-run average cost, for queues utilizing just one of these controls are well understood, to our knowledge, no work has been done to understand the effects of their interaction.

A stationary policy is an N*-policy* if it calls for turning the server on when N jobs are in the system and may turn off the server only when the system is empty. Heyman [17] showed N-policies are optimal for $M/M/1$ queues with startup costs and a single service rate. For $M/M/1$ queues with dynamic service rate control, Lippman [21] showed the optimal service rate structure is monotone with the number in queue. Our main result is a combination of these results.

Define a policy that has a threshold for turning the server on, a series of thresholds increasing with the number in system for the service rate, and that only (possibly) turns the server off when the queue is empty as a *rate-threshold* N*-policy* (see Definition 3.1 below). We prove the existence of a long-run average cost optimal rate-threshold N-policy. From a managerial perspective, we are interested in such polices as they are useful in developing more sophisticated controls for implementation in systems of any size. From a methodological viewpoint, we note that inductive arguments in the spirit of Koole [19] and similar to those used in [7,20,29], that are commonly used to prove structural properties such as convexity are intractable. This is because work conserving policies, those that do not idle or turn off the server with work in the system, are not necessarily optimal under a finite horizon discounted criterion. To see this, suppose the idling cost rate is less than the service cost rate (an assumption made in the current study) then in a one period problem with zero terminal costs, idling is better than serving. Instead we use renewal theory arguments coupled with a probabilistic interpretation of the relative value functions in average cost MDP theory to prove our results.

Queues with removable servers have been studied for half a century, beginning with Yadin and Naor [28] who analyzed $M/G/1$ queues under N-policies. Heyman [17] and Bell [4] proved the optimality of N-policies for $M/G/1$ queues with warm up costs under the average and total discounted reward criteria, respectively. Baker [1] first studied $M/M/1$ queues under N-policies with both startup costs and exponentially distributed warm up times, but without service rate control. Borthakur et al. [5] extended this work, allowing for general warm up times, providing steady-state probabilities, average wait time, and number in queue. In the years since, many generalizations of these models have been studied including by Feinberg and Kella [12] who analyzed an $M/G/1$ queue where the service time becomes known upon arrival. Other variations of these models include queues with server breakdowns and warm up times [27], queues with server vacations [18], and queues with batch arrivals [11], among others.

There is also a large literature on dynamic service rate control in queues. Crabill [8] was the first to model a system with varying service rates using MDPs. Lippman [21] examined the optimal policy for $M/M/1$ queues with dynamic service rate control proving the monotonicity of the optimal service rate with the number in system. George and Harrison [16] examined a queue where service rates are chosen from a closed subset of $[0, \infty]$ and Kumar et al. [20] considered a single server system with dynamic service rate control and

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

Markov modulated arrivals. Dimitrakopoulos and Burnetas [10] studied the value of dynamic service rate control in combination with admission control.

Recently, more work has been done motivated specifically by data center applications incorporating sleep or low power modes. Research from this perspective often refers to servers as *energy aware*. However, they are modeled similarly to removable servers seen in the earlier queueing literature. To our knowledge, the first queueing-based approach from this perspective was by Cehn et al. [6]. Gebrehiwot et al. [15] studied the use of sleep-state control under various policies in $M/G/1$ queues. Gandhi et al. [13] examined an $M/G/1$ queue with warm up times and many sleep states for a specific cost function: the product of the mean power consumption and mean response time in steady state. Much of the work in this field has also considered multi-server systems, such as [14] and [22,23].

The remainder of the paper is organized as follows. Section 2 introduces the model. The main result is stated in Section 3, where the proof is divided into several parts. In Section 3.1, it is shown that the optimal policy is work conserving. In Section 3.2, the existence of an optimal policy with a series of thresholds determining the service rate is proven. In Section 3.3, we show the optimal policy has a threshold for turning the server on. Numerical considerations are presented in Section 4 and the article is concluded in Section 5.

2. MODEL DESCRIPTION

We consider a Markovian queueing system with a single removable server that, in addition to being able to be turned off, dynamically chooses its service rate. Jobs arrive according to a Poisson process with rate λ. When the server is on, the decision-maker can choose from n rates where service times for rate k are assumed to be exponentially distributed with rate μ_k and $\mu_1 < \mu_2 < \cdots < \mu_n$. We assume the server can instantaneously switch service rates and can do so without incurring any cost. Additionally we allow the server to idle at the discretion of the decision-maker whenever it is on. If the server is not on, jobs can be serviced only after a warm up period, the length of which follows an exponential distribution with rate γ. To ensure a policy that admits finite average cost exists, we assume $\lambda < \mu_n$.

Let c_w, c_u, c_k be the cost per unit time when the server is warming, idling, or serving at rate μ_k, respectively. Assume c_k is non-decreasing in k. We also assume that idling is less costly than serving at any rate or warming; $c_u < c_1, c_w$. The holding cost rate function, $h(i)$, depends only on the number in system, i, and not the state of the server. Furthermore, $h(i)$ is non-decreasing in i, such that $h(i) \to \infty$ as $i \to \infty$. Finally, assume $\sum_{i=0}^{\infty} \alpha^i h(i) < \infty$ for all $\alpha \in (0,1)$ which ensures holding costs grow subexponentially.

Define the state space $S = \mathbb{Z}_+ \times \{0,1\}$, where $(i,j) \in S$ denotes the state where i jobs are in the system, $j = 0$ represents the server warming or off and $j = 1$ represents the server being on. The set of available actions depends on the current state. Let

$$A(i,j) = \begin{cases} \{\mathit{warm},\ \mathit{off}\} & \text{if } i \geq 0,\ j = 0, \\ \{(\mathit{idle}, \mathit{on}), (\mathit{idle}, \mathit{off})\} & \text{if } i = 0,\ j = 1, \\ \{(\mathit{idle}, \mathit{on}), (\mathit{idle}, \mathit{off}), (k, \mathit{on}), (k, \mathit{off})\} & \text{if } i \geq 1,\ j = 1, \end{cases} \tag{2.1}$$

where $k = 1, \ldots, n$. When the server is off ($j = 0$), the action *off* represents leaving the server off, and *warm* represents starting a warm up period. When the server is on ($j = 1$), the first dimension of the action, k or *idle* represents the service rate used, μ_k, $k = 1, 2, \ldots, n$, or if the server idles. If the server idles, the second dimension specifies if the server should be turned off or left on following the next arrival. If the server is working, the second dimension

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

of the action specifies if the server should be turned off or left on if the next event is a service completion.

Remark 2.1: Since turning the server off occurs instantaneously and leads to a change in state ($j = 1$ to $j = 0$), allowing the action *off* in states $(i, 1)$ would lead to instantaneous transitions. Thus we define the service actions in the above non-standard, two-dimensional way so that all transition rates are finite and uniformization can be applied.

Given the exponential inter-arrival, service, and warm up times, the continuous time Markov chain induced by a policy can be uniformized in the spirit of Lippman [21] so the equivalent discrete time MDP can be analyzed. Let the uniformization rate be $\lambda + \gamma + \mu_n$ and, without loss of generality, scale time so that $\lambda + \gamma + \mu_n = 1$. The remainder of the paper discusses this discrete model.

A deterministic decision rule d is a map from S to $\mathcal{A} := \bigcup_{s \in S} A(s)$ such that action $d(s)$ is used when the system is in state s. A policy π is defined as a sequence of decision rules $\pi = \{d_1, d_2, \ldots\}$ where d_m defines the action taken in the mth decision epoch. We let Π be the set of all non-anticipatory policies. We say a policy is *stationary* if it does not depend on the time, i.e. it is of the form $\{d, d, \ldots\}$ and denote such policies d^∞.

Let $c(s, a)$ be the expected cost of taking action a when in state s. Explicitly,

$$c((i,j),a) = \begin{cases} h(i) + c_w & \text{if } a = \mathit{warm}, j = 0, \\ h(i) & \text{if } a = \mathit{off}, j = 0 \\ h(i) + c_u & \text{if } a \in \{(\mathit{idle}, \mathit{on}), (\mathit{idle}, \mathit{off})\}, j = 1, \\ h(i) + c_k & \text{if } a \in \{(k, \mathit{on}), (k, \mathit{off})\}, j = 1,\ \ k = 1, \ldots, n. \end{cases} \qquad \textbf{(2.2)}$$

The *N-step expected total cost* under policy π given initial state s is

$$J_N^\pi(s) = \mathbb{E}^\pi \left[\sum_{m=0}^{N-1} c(S_m, d_m(S_m)) \mid S_0 = s \right],$$

where S_m is the state of the system in the mth decision epoch and $d_m(S_m)$ is the decision rule in the mth decision epoch under policy π. The *long-run average cost* of policy π given initial state s is given by

$$g^\pi(s) = \limsup_{N \to \infty} \frac{J_N^\pi(s)}{N}.$$

Let $g^*(s)$ be the *optimal expected average cost*,

$$g^*(s) = \inf_{\pi \in \Pi} g^\pi(s).$$

We call a policy π^* *long-run average cost optimal* if $g^{\pi^*}(s) = g^*(s)$ for all $s \in S$.

Consider the policy that always uses rate μ_n, never turns off, and begins warming immediately, if the server is not initially in the on state. Note that the induced Markov chain has a single recurrent class and all the states where the server is off are transient. Since $\lambda < \mu_n$, the set of recurrent states for this Markov chain corresponds to a stable $M/M/1$ queue. That is to say, the set of states of the form $(i, 1)$ for $i \geq 0$ act as a positive recurrent, birth-death process. We use this policy, denoted π, to verify the three (CAV) assumptions in Corollary 7.5.9 of Sennott [25]. These are sufficient conditions to apply

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

Theorem 7.2.3 [25] which guarantees the existence of an optimal stationary policy with finite average cost which is independent of the initial state.

Assumption 2.2: *The (CAV) assumptions [adapted for the current study] are:*

1. *There exists a policy such that the expected first passage time and cost from any state (i,j) to $(0,1)$ are finite. We call the positive recurrent class of the Markov chain induced by π, R_π.*
2. *Given $U>0$, the set $D_U=\{(i,j)|c((i,j),a)\leq U$ for some $a\}$ is finite.*
3. *Given $(i,j)\in S-R_\pi$, there exists a policy $\hat{\pi}_{(i,j)}$ such that the expected first passage time and cost from $(0,1)$ to (i,j) are finite.*

Lemma 2.3: *The (CAV) assumptions hold.*

Proof: We note π induces a stable $M/M/1$ queue thus states $(i,1)$ for $i\geq 0$ form a positive recurrent class, R_π, and the stationary distribution on R_π is $\phi_i=(1-\rho)\rho^i$, where $\rho=\lambda/\mu_n$. Thus, the long-run average cost of this policy is $c_u(1-\rho)+c_n\rho+\sum_{i=0}^{\infty}h(i)(1-\rho)\rho^i$ which, following from the assumption that $h(i)$ grows sub-exponentially, is finite. It also follows from basic continuous-time Markov chain theory that the expected first passage cost between any two positive recurrent states is finite.

Finiteness of the expected transition time and cost from any transient state $(i',0)$ to the recurrent class remains to be shown. Since the server begins warming immediately if it is not initially on, the expected transition time is $1/\gamma$. The expected cost before entering the recurrent class is the sum of the expected energy and holding costs incurred while warming. The energy cost is finite since the first passage time is finite. Let H be the holding cost incurred before absorption into the recurrent class. Then

$$\begin{aligned}\mathbb{E}[H|S_0=(i',0)] &= \sum_{m=0}^{\infty}\mathbb{E}[H\mid S_0=(i',0), m \text{ arrivals before the server turns on}]\\ &\quad\times\mathbb{P}(m \text{ arrivals before the server turns on}\mid S_0=(i',0))\\ &\leq\sum_{m=0}^{\infty}(m+1)h(i'+m)\\ &\quad\times\mathbb{P}(m \text{ arrivals before the server turns on}\mid S_0=(i',0))\end{aligned}$$

Note the number of arrivals before the server turns on follows a geometric distribution with parameter $\beta=\lambda/\lambda+\gamma$. Thus,

$$\mathbb{E}[H|S_0=(i',0)]\leq\sum_{m=0}^{\infty}(m+1)h(i'+m)(1-\beta)\beta^m<\infty$$

where finiteness results from common properties of power series shown, among other places, in Theorem 8.1 of Rudin [24]. This verifies the first assumption.

To check the next two assumptions, recall that $h(i)\to\infty$ as $i\to\infty$. Hence for any $U>0$ there exists M_U such that $h(M_U)>U$. Thus, $D_U\subseteq\{(i,j)|i<M_U\}$ is finite as desired. Finally, we check that given $(i,j)\in S-R_\pi$, there exists a policy $\hat{\pi}_{(i,j)}$ such that the expected time and cost to transition from state $(0,1)$ to (i,j) is finite. Since (i,j) is transient under π, it must be of the form $(i,0)$. Consider any stationary policy, d^∞ such that $d^\infty(i',0)=$ *off* for all $i'\leq i$ and $d^\infty(0,1)=($*idle*, *off*$)$. Thus given initial state $(0,1)$, the system under d^∞

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

shuts the server off after the first arrival, transitioning into state $(1, 0)$ and stays off until there are at least $i + 1$ jobs in the system. Thus, the system must pass through state $(i, 0)$ as no jobs can be served while the server is off. Since the arrival process is Poisson, the expected hitting time and cost are finite. Thus we have constructed an appropriate policy for any $(i, j) \in S - R_\pi$. ■

Accordingly, we restrict our attention to stationary policies that induce a DTMC with a single positive recurrent class and finite average cost. We refer to such policies as *stable policies*. For any stable policy π, we redefine $g^\pi(s) = g^\pi$ and let $g^*(s) = g^*$ for all $s \in S$. Additionally, Lemma 2.3 verifies the conditions for Theorem 7.4.3 [25], which states that g^* and an optimal policy π^* can be obtained by solving the average cost optimality equations (ACOE) for g^* and $r(i, j)$, for all states (i, j) in the positive recurrent class induced by π^* where $r(i, j)$ is the relative value function. The ACOE for the current study follow. For $i \geq 0$ and $j = 0$,

$$g^* + r(i,0) = h(i) + \min \begin{cases} \lambda r(i+1,0) + (1-\lambda) r(i,0), \\ c_w + \lambda r(i+1,0) + \gamma r(i,1) + (1-\lambda-\gamma) r(i,0), \end{cases} \tag{ACOE}$$

$$g^* + r(0,1) = h(0) + \min \begin{cases} c_u + \lambda r(1,1) + (1-\lambda) r(0,1), \\ c_u + \lambda r(1,0) + (1-\lambda) r(0,1), \end{cases}$$

and for $i \geq 1$, $j = 1$, and $k = 1, \ldots, n$,

$$g^* + r(i,1) = h(i) + \min \begin{cases} c_u + \lambda r(i+1,1) + (1-\lambda) r(i,1), \\ c_u + \lambda r(i+1,0) + (1-\lambda) r(i,1), \\ c_k + \lambda r(i+1,1) + \mu_k r(i-1,1) + (1-\lambda-\mu_k) r(i,1), \\ c_k + \lambda r(i+1,1) + \mu_k r(i-1,0) + (1-\lambda-\mu_k) r(i,1). \end{cases}$$

3. MAIN RESULT

In this section we state and prove (over several steps) our main theoretical result: a complete characterization of an optimal control. As conventional inductive arguments were elusive, we have used renewal arguments through a probabilistic interpretation of the relative value function. We first state the definition of a rate-threshold N-policy.

DEFINITION 3.1: *A policy is a* ***rate-threshold*** N***-policy*** *if it has one of the two following structures:*

(1) *Never turn the server off and the service rate increases monotonically with the number in system.*
(2) *Turn the server off when the system is empty and begin warming when N or more jobs are in the queue. The service rate increases monotonically with the number in system.*

We characterize such policies with a triple $(x, y, \mathbf{z})$ where x is the action taken when the system is empty, *idle* or *off*, y is the smallest queue length where warming the server is optimal and $\mathbf{z}$ is an $n - 1 \times 1$ vector where the kth entry is the smallest queue length where rate μ_{k+1} is used. Note the threshold for turning on the system, y, is only meaningful to

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

characterize the policy when the system turns off when empty. Otherwise all of the states where the server is off are transient and actions taken in those states do not affect the long-run average cost as long as the server must turn on eventually.

Theorem 3.2: *There exists an optimal rate-threshold N-policy.*

The proof is divided into several parts which we provide over the next several sections.

3.1. Work Conserving Policies

We begin our analysis by showing the intuitive fact that turning the server on immediately after it was turned off is sub-optimal.

Definition 3.3: *A policy, $\pi = d^\infty$, has **immediate restarts** if there exists $i \geq 1$ and $k \in \{1, 2, \ldots, n\}$ such that $d(i, 1) = (k, \mathit{off})$ and $d(i-1, 0) = \mathit{warm}$.*

Lemma 3.4: *There exists an average cost optimal policy that does not have immediate restarts.*

Proof: We use a coupling argument to prove this result. Let $\pi = d^\infty$ be a policy with immediate restarts. We define an alternate policy π' that does not have immediate restarts and that has lower average cost. Call the system under π System 1 and the system under π' System 2. Let π' mimic the actions of π until System 1 turns off following a service completion and transitions into a state $(i, 0)$ such that $d(i, 0) = \mathit{warm}$. System 2 instead remains on and idles until either System 1 turns back on or an arrival occurs; if an arrival occurs first, System 2 turns off. If the server in System 1 restarts before an arrival occurs, both systems will be in state $(i-1, 1)$, and if an arrival occurs first, both systems will be in state $(i, 0)$. Thus, in either case the two systems have coupled. However, since the cost of idling is lower than that of warming, π' incurs strictly less cost. ■

Lemma 3.5: *If a policy, $\pi = d^\infty$, is stable and does not have immediate restarts, then there exists finite L_π such that for all queue lengths $i > L_\pi$,*

$$d(i, 1) \notin \{(\mathit{idle}, \mathit{on}), (\mathit{idle}, \mathit{off}), (k, \mathit{off}), k = 1, \ldots, n\}.$$

That is to say, there is a finite longest queue length above which the policy does not turn off or idle.

Proof: Let π be a stable policy that does not have immediate restarts and that shuts off or idles after a service completion in state $(L_\pi, 1)$ and let Δ^π be the DTMC it induces. All states in $D(L_\pi) := \{(i, j) \mid i < L_\pi - 1,\ j = 0, 1\}$ are transient in Δ^π. If the system begins in state (i, j), $i < L_\pi - 1$, it reaches a queue length of L_π with probability at least $\lambda^{L_\pi - i}$, the likelihood of $L_\pi - i$ consecutive arrivals. Further, once the number in the system reaches length L_π, the queue length cannot reach i again since to do so $(L_\pi, 1)$ must lead to $(L_\pi - 1, 1)$. This is not possible since the server either idles in $(L_\pi, 1)$ or is off when $L_\pi - 1$ jobs are in the system. Figure 1 illustrates this observation in the case where $d(L_\pi, 1) = (k, \mathit{off})$. Hence, if no such L_π exists, all states are transient and a stationary distribution does not exist. ■

Definition 3.6: *A **work conserving policy** is a policy that does not idle or turn the server off when customers (work) remain in the system.*

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

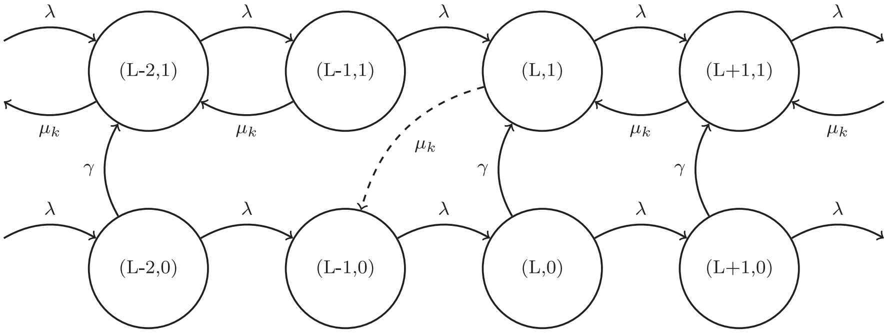

FIGURE 1. Rate transition diagram for the DTMC induced by π containing all edges that may have positive rates if π shuts off after completing service in state $(L, 1)$. Note: for simplicity, the same service rate is used in all states and we have suppressed the dependence of L on π.

PROPOSITION 3.7: *There exists an optimal work conserving policy without immediate restarts.*

PROOF: Using Lemma 3.4 consider an optimal policy π that does not have immediate restarts. Let L_π be the level described in Lemma 3.5 so above L_π the server does not idle or turn off. For simplicity, for the remainder of the proof, suppress the dependence of L on π. If π is not work conserving at least one of the following holds:

(1) the server is turned off following a service completion with work remaining;
(2) the server idles with work remaining; or
(3) the server turns off following an arrival after idling while empty.

We discuss each case separately.

Case (1) The server turns off following a service completion with work remaining. Suppose π turns the server off with $L - 1 > 0$ jobs in the system. That is, $d(L, 1) = (\cdot, \mathit{off})$. By Lemma 3.4, $d(L-1, 0) = \mathit{off}$ (otherwise π has immediate restarts) and states in T are transient, where

$$T := \{(i, j) \mid i < L, j = 1\} \cup \{(i, j) \mid i < L - 1, j = 0\}.$$

Define policy $\pi' = (d')^\infty$ such that $d'(i, j) = d(i + L - 1, j)$. That is to say, π' uses the action taken by π when there are $L - 1$ additional jobs in the system. Both policies induce renewal processes and thus their average cost can be calculated using the renewal reward theorem. We demonstrate that π' outperforms π by showing on any sample path both policies induce cycles of the same length while π' incurs less cost per cycle.

Refer to the system under π as System 1, and the system under π' as System 2. Initialize System 1 in state $(L - 1, 0)$ and System 2 in $(0, 0)$. Note each system has a single positive recurrent class so the average costs are independent of the initial state. We explain why these are recurrent states for their respective systems below. Suppose both systems experience the same arrivals, potential service completions and warming times are drawn from a common sequence. System 2 has $L - 1$ fewer jobs than System 1 and thus both systems take the same actions, incurring the same energy costs at all times. With probability

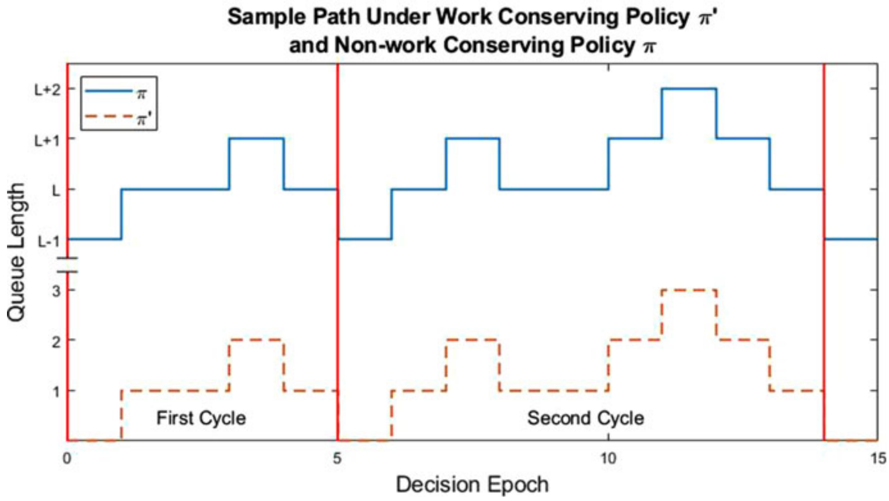

FIGURE 2. Cycles under work conserving policy π' and non-work conserving policy π which turns off with $L-1$ jobs in the queue.

one, System 1 eventually returns to state $(L-1, 0)$ and System 2 returns to state $(0, 0)$ since these states are positive recurrent. Further, by construction these events happen simultaneously. Thus on every sample path, Systems 1 and 2 have exactly the same cycle length, however System 2 always incurs less total cost since there are always fewer jobs in it. Hence, π' has lower average cost than π. Note the assumption that $d(L, 1) = (\cdot, \mathit{off})$ is arbitrary. Any $0 < i < L$ where π calls for turning the server off (and leaving it on when there are more than i customers) would suffice.

Case (2) The policy π idles with work remaining in the queue. Suppose for example, $d(L, 1) \in \{(idle, on), (idle, \mathit{off})\}$. This case follows identically to the previous case except we define policy $\pi' = (d')^\infty$ such that $d'(i, j) = d(i + L, j)$ and initialize System 1 in $(L, 1)$ and System 2 in $(0, 1)$.

Case (3) The policy idles while empty then turns off when an arrival occurs $(d(0, 1) = (idle, \mathit{off}))$. First note, that this case need only be considered if $d(1, 1) = (k, on)$, $k \in \{1, \ldots, n\}$. Otherwise, $(0, 1)$ is transient and $d(0, 1)$ does not affect the long-run average cost. Define $\pi' = (d')^\infty$ such that $d'(0, 0) = \mathit{off}$, $d'(1, 1) = (k, \mathit{off})$, and $d'(i, j) = d(i, j)$ otherwise. Arbitrarily, initializing both systems in $(1, 1)$, the systems cycle jointly. Both systems incur the same costs except when System 1 is idling in state $(0, 1)$ and System 2 is off in $(0, 0)$. Thus, π' has lower average cost.

Finally it remains to show that with probability 1 and given an arbitrary initial state, each system enters the state they are initialized in above in finite time. For example, in Case (1) System 1 must enter $(L-1, 0)$ and System 2 must enter $(0, 0)$ while in Case (2) System 1 must enter $(L, 1)$ and System 2 must enter $(0, 1)$. We verify this by showing that each system will enter the positive recurrent class in finite time. Again we begin with Case (1).

Let R_π and $R_{\pi'}$ be the positive recurrent classes of the DTMCs induced by π and π', respectively. Note $\{(i, 1) | i \geq L\} \cup \{(L-1, 0)\} \subseteq R_\pi$ and $\{(i, 1) | i \geq 1\} \cup \{(0, 0)\} \subseteq R_{\pi'}$. First consider System 1 and three cases for the initial state. We refer to these as Cases 1a,

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

1b, and 1c to avoid confusion with the cases already defined. The argument holds similarly for System 2 however we need not consider Case (1c).

Case (1a) Initial state $(i, 1)$ with $i \geq L$. When the server is on and there are more then L jobs, the system is already in R_π and thus $(L-1, 0)$ is reached.

Case (1b) Initial state $(i, 0)$ with $i \geq L$. Since π is optimal (and stable), the average cost is finite, thus the system must turn on eventually (otherwise it incurs infinite holding cost). Once this occurs, the result follows from Case (1a).

Case (1c) Initial state (i, j) where $i < L$ and $j = 0, 1$. As noted previously these states are transient and thus eventually the system reaches a state with L jobs in the queue. If at this time the server is on, it has reached $(L, 1)$ and the result follows by Case (1a), otherwise the result follows by Case (1b).

Cases 2 and 3 follow in the same manner. Thus we have constructed a work conserving policy with average cost at least as low as an arbitrary π that is not work conserving. ■

3.2. Rate Threshold Policies

The next two results lead to the main result of this section where it is shown that an optimal policy exists with monotone (in the number in the system) service rates.

Lemma 3.8: *Suppose $i+2$ is such that under a work conserving average cost optimal policy the optimal action in $(i+2, 1)$ is to work at rate μ_k, $k \in \{1, \ldots, n\}$ and leave the server on after a service. We have*

$$r(i+2, 1) - r(i+1, 1) \geq r(i+1, 1) - r(i, 1).$$

Proof: Consider two systems started on the same probability space so that they see the same arrivals and the same potential services. System 1 starts in state $(i+2, 1)$ using the optimal action (k, on), and System 2 starts in state $(i+1, 1)$. Assume that System 2 also uses (k, on) so that each system incurs the same energy costs. Note this action is potentially sub-optimal for System 2.

Consider the first event. If the first event is a service completion, which occurs with probability $\mu_k/(\lambda + \mu_k)$, both systems see the service, and enter states $(i+1, 1)$ and $(i, 1)$, respectively. The remaining difference in the costs is $r(i+1, 1) - r(i, 1)$. If on the other hand, there is an arrival, then the queue length in each system increases by 1. System 2 continues to follow the same actions as System 1 (which does not include ever turning the server off) until System 1 re-enters $(i+2, 1)$. At this time System 2 is back in state $(i+1, 1)$. Note the finite return time is guaranteed since System 1 is using an optimal policy, does not turn the server off in $(i+2, 1)$ and (by assumption) has finite average cost. Since the systems maintain their relative positions, the expected difference in cost on this cycle is non-negative. Denote this difference in cost by B. The remaining difference in cost is then $B + r(i+2, 1) - r(i+1, 1)$.

Combining these two cases yields

$$\begin{aligned} r(i+2, 1) - r(i+1, 1) \geq{} & \frac{\lambda}{\lambda + \mu_k}[B + r(i+2, 1) - r(i+1, 1)] \\ & + \frac{\mu_k}{\lambda + \mu_k}[r(i+1, 1) - r(i, 1)] \end{aligned}$$

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

$$\geq \frac{\lambda}{\lambda+\mu_k}[r(i+2,1)-r(i+1,1)] + \frac{\mu_k}{\lambda+\mu_k}[r(i+1,1)-r(i,1)].$$

This implies

$$\frac{\mu_k}{\lambda+\mu_k}[r(i+2,1)-r(i+1,1)] \geq \frac{\mu_k}{\lambda+\mu_k}[r(i+1,1)-r(i,1)],$$

which yields

$$r(i+2,1)-r(i+1,1) \geq r(i+1,1)-r(i,1),$$

as desired. ■

Lemma 3.9: *Suppose there exists an optimal policy that uses the action* (k, off) *for some* $k \in \{1,2,\ldots,n\}$ *in state* $(1,1)$*. Then*

$$[r(2,1)-r(1,1)]-[r(1,1)-r(0,0)] \geq 0.$$

Proof: Define two systems on the same probability space so that

(a) both systems see the same arrival stream,

(b) service times are selected from a common sequence when servers begin serving at the same rate at the same time, and

(c) warming times are drawn from a common sequence when servers begin warming at the same time.

System 1 starts in state $(2,1)$ and uses an optimal work conserving policy, while System 2 starts in state $(1,1)$ and uses the same service rate as System 1 with the action that turns the system off if the next event is a service completion. If the first event is an arrival, System 2 uses the same action as System 1 until it (System 2) re-enters state $(1,1)$ and again attempts to turn the server off if the next event is a service completion. This process continues until System 2 turns off. Note this means System 2 will be in state $(0,0)$ while System 1 is in state $(1,1)$. After this time, System 2 ceases mimicking the actions of System 1 and instead follows an optimal work conserving policy.

Consider the first actual event (ignoring dummy transitions due to uniformization). If it is a service completion at the rate $\mu_{k'}$ (where k' is the service rate used by System 1) the systems transition to states $(1,1)$ and $(0,0)$, respectively. If the first event is an arrival (with probability $(\lambda/\lambda+\mu_{k'})$) the queue lengths of each system increase by 1. This continues until such time that the systems re-enter states $(2,1)$ and $(1,1)$. Since System 1 remains with a queue that is one higher than System 2, the expected cost of System 1 is higher than that of System 2. Say this difference is $\bar{B} \geq 0$. Since the policy followed by System 2 may not be optimal, this all leads to the following inequality

$$\begin{aligned} r(2,1)-r(1,1) &\geq \frac{\lambda}{\lambda+\mu_{k'}}[\bar{B}+[r(2,1)-r(1,1)]] + \frac{\mu_{k'}}{\lambda+\mu_{k'}}[r(1,1)-r(0,0)] \\ &\geq \frac{\lambda}{\lambda+\mu_{k'}}[r(2,1)-r(1,1)] + \frac{\mu_{k'}}{\lambda+\mu_{k'}}[r(1,1)-r(0,0)] \end{aligned}$$

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

A little algebra yields

$$\frac{\mu_{k'}}{\lambda + \mu_{k'}}[r(2,1) - r(1,1)] \geq \frac{\mu_{k'}}{\lambda + \mu_{k'}}[r(1,1) - r(0,0)].$$

The result follows. ■

Remark 3.10: The previous proof showing $r(i,j)$ is convex does not require that $r(i,j)$ be non-decreasing (this is also the case for subsequent proofs). Typically, the two properties are connected in that the assumption that a function is non-decreasing is used to show convexity along the boundary of the state space. In the current work, we avoid this because we know the behavior of an optimal policy when we approach the boundary, can condition appropriately and we are using average cost as an optimality criterion (no discounting). Having said that, it turns out that $r(i,j)$ is indeed non-decreasing. We have included a proof of this in Appendix A for the interested reader.

PROPOSITION 3.11: *There exists an optimal policy such that the optimal service rate is monotone non-decreasing with the number in system.*

PROOF: Consider $i \geq 2$ and suppose an optimal work conserving policy uses service rate $\mu_{k(i)}$ (resp. $\mu_{k(i+1)}$) in state $(i,1)$ (resp. $(i+1,1)$). Without loss of generality assume that $\mu_{k(i)} \neq \mu_{k(i+1)}$. To prove the result it suffices to show that $\mu_{k(i)} < \mu_{k(i+1)}$. Since the optimal policy is work conserving and $i \geq 2$, the server is left on after service in $(i,1)$. The (ACOE) imply

$$\begin{aligned} g^* + r(i,1) &= h(i) + c_{k(i)} + \lambda r(i+1,1) + \mu_{k(i)} r(i-1,1) + (1 - \lambda - \mu_{k(i)}) r(i,1) \\ &\leq h(i) + c_{k(i+1)} + \lambda r(i+1,1) + \mu_{k(i+1)} r(i-1,1) + (1 - \lambda - \mu_{k(i+1)}) r(i,1). \end{aligned} \tag{3.1}$$

Similarly,

$$\begin{aligned} g^* + r(i+1,1) &= h(i+1) + c_{k(i+1)} + \lambda r(i+2,1) + \mu_{k(i+1)} r(i,1) \\ &\quad + (1 - \lambda - \mu_{k(i+1)}) r(i+1,1) \\ &\leq h(i+1) + c_{k(i)} + \lambda r(i+2,1) + \mu_{k(i)} r(i,1) + (1 - \lambda - \mu_{k(i)}) r(i+1,1). \end{aligned} \tag{3.2}$$

Assume at least one of the inequalities (3.1) or (3.2) is strict, otherwise both serving at rates $\mu_{k(i)}$ and $\mu_{k(i+1)}$ is optimal in states $(i,1)$ and $(i+1,1)$ so that a non-decreasing optimal control can be constructed. Without loss of generality, assume that the inequality in (3.1) is strict. Note that $A < B$ and $C \leq D$ implies $C - B < D - A$. Combining (3.1) and (3.2) yields,

$$\begin{aligned} &\lambda[r(i+2,1) - r(i+1,1)] + \mu_{k(i)}[r(i,1) - r(i-1,1)] + (1 - \lambda - \mu_{k(i)})[r(i+1,1) - r(i,1)] \\ &\quad > \lambda[r(i+2,1) - r(i+1,1)] + \mu_{k(i+1)}[r(i,1) - r(i-1,1)] \\ &\qquad + (1 - \lambda - \mu_{k(i+1)})[r(i+1,1) - r(i,1)] \end{aligned}$$

A little algebra yields

$$\begin{aligned} &\mu_{k(i)}[[r(i,1) - r(i-1,1)] - [r(i+1,1) - r(i,1)]] \\ &\quad > \mu_{k(i+1)}[[r(i,1) - r(i-1,1)] - [r(i+1,1) - r(i,1)]]. \end{aligned}$$

Thus,

$$[\mu_{k(i)} - \mu_{k(i+1)}][[r(i,1) - r(i-1,1)] - [r(i+1,1) - r(i,1)]] > 0. \tag{3.3}$$

Since r is convex (see Lemma 3.8), $\mu_{k(i)} < \mu_{k(i+1)}$ as desired.

Next consider the state $(1,1)$. Note that if an optimal action is to work at rate $\mu_{k(1)}$ and keep the server on if the next event is a service, the previous proof holds. Assume the decision-maker turns the server off after a service completion in $(1,1)$ and that $\mu_{k(1)} \neq \mu_{k(2)}$. The (ACOE) yield

$$\begin{aligned} g^* + r(1,1) &= h(1) + c_{k(1)} + \lambda r(2,1) + \mu_{k(1)} r(0,0) + (1 - \lambda - \mu_{k(1)}) r(1,1) \\ &\le h(1) + c_{k(2)} + \lambda r(2,1) + \mu_{k(2)} r(0,0) + (1 - \lambda - \mu_{k(2)}) r(1,1). \end{aligned} \tag{3.4}$$

and

$$\begin{aligned} g^* + r(2,1) &= h(2) + c_{k(2)} + \lambda r(3,1) + \mu_{k(2)} r(1,1) + (1 - \lambda - \mu_{k(2)}) r(2,1) \\ &\le h(2) + c_{k(1)} + \lambda r(3,1) + \mu_{k(1)} r(1,1) + (1 - \lambda - \mu_{k(1)}) r(2,1). \end{aligned} \tag{3.5}$$

Again, assume without loss of generality that (3.4) is strict. This yields

$$\mu_{k(1)}[[r(1,1) - r(2,1)] - [r(0,0) - r(1,1)]] > \mu_{k(2)}[[r(1,1) - r(2,1)] - [r(0,0) - r(1,1)]].$$

A little algebra yields

$$(\mu_{k(1)} - \mu_{k(2)})[[r(1,1) - r(2,1)] - [r(0,0) - r(1,1)]] > 0.$$

Using Lemma 3.9 yields $\mu_{k(1)} < \mu_{k(2)}$ as desired. ■

Remark 3.12: Note that Proposition 3.11 implies that there exist thresholds $1 = \ell_1 \le \ell_2 \le \cdots \le \ell_n \le \ell_{n+1} = \infty$ such that the optimal action in state $(i,1)$ uses rate μ_k if $\ell_k \le i < \ell_{k+1}$.

3.3. Thresholds to Turn the Server On

We conclude our analysis by showing there exists a threshold for turning the server on. Then we combine all the threshold results into a single result characterizing the structure of the average cost optimal policy.

DEFINITION 3.13: *We say a stationary policy is an* ***on threshold policy*** *with parameter N if it calls for the server to warm when there are more than N jobs in the system and remains off otherwise.*

Throughout this section we make the assumption that there exists a queue length, above which we would always warm the server. Note that in fact it suffices to have an infinite number of queue lengths where it is optimal to warm the server; making all states of the form $(i,0)$ recurrent for all (stationary) work-conserving policies that ever turn the server off. Recall, the action *warm* attains the minimum in ACOE, and is thus optimal, in state $(i,0)$ if

$$r(i,0) - r(i,1) \ge \frac{c_w}{\gamma}.$$

Thus, the existence of the aforementioned threshold is assured if being optimal to warm the server in $(i,0)$ implies it is optimal to warm the server in $(i+1,0)$. To get this result,

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

it suffices to show

$$r(i+1,0) - r(i+1,1) \geq r(i,0) - r(i,1). \tag{3.6}$$

Without loss of generality assume that it is optimal to turn the server off after service in state $(1,1)$ (otherwise, the states with the server off are transient for work-conserving policies). A little rearranging in (3.6) yields an equivalent inequality,

$$r(i+1,0) - r(i,0) \geq r(i+1,1) - r(i,1).$$

Lemma 3.14: *Suppose i is such that under a work conserving average cost optimal policy it is optimal to warm the server in $(i+1,0)$. Let r be the relative value function satisfying the* (ACOE)*. We have*

$$r(i+1,0) - r(i,0) \geq r(i+1,1) - r(i,1).$$

Proof: Start two systems, Systems 1 and 2, in states $(i+1,0)$ and $(i,0)$, respectively. Suppose System 2 also uses the (potentially sub-optimal) action to warm the server. The first (actual) event is either the server turns on and is ready for service with probability $\gamma/(\lambda+\gamma)$ or an arrival with probability $\lambda/\lambda+\gamma$. In the first case, the remaining difference in the costs incurred by the systems after the first event are $r(i+1,1) - r(i,1)$. In the second case, the systems enter states $(i+2,0)$ and $(i+1,0)$, respectively. Because it is optimal to warm the server for a sufficiently large queue length, eventually the server in System 1 goes from off to on. Since, the policy is work conserving, the server stays on and eventually reaches state $(i+1,1)$. Assuming System 2 chooses the same actions as System 1, it remains having one less customer in queue. When System 1 enters state $(i+1,1)$, System 2 is in state $(i,1)$. At this point, assume both policies follow the optimal policy. The difference in costs is the difference in (holding) costs accrued until the time System 1 reaches state $(i+1,1)$ (denoted $\tilde{B}$) and the remaining costs if both systems then follow the optimal policy. That is,

$$\begin{aligned} r(i+1,0) - r(i,0) &\geq \frac{\gamma}{\lambda+\gamma}[r(i+1,1) - r(i,1)] + \frac{\lambda}{\lambda+\gamma}[\tilde{B} + r(i+1,1) - r(i,1)] \\ &\geq r(i+1,1) - r(i,1) \end{aligned}$$

as desired. ■

Proposition 3.15: *There exists an optimal policy that is an on threshold policy.*

Proof: As mentioned previously, this an immediate consequence of Lemma 3.14 (recall the discussion immediately prior to (3.6)). ■

Theorem 3.2 follows directly from Propositions 3.7, 3.11, and 3.15.

4. NUMERICAL STUDY

In this section we provide insights into the value of the decision maker having both control mechanisms at their disposal in the case where $n = 2$. We refer to the service rates as low and high. First, we compare the optimal policy for systems with increasing traffic intensity to investigate when the ability to turn the server off will actually be utilized. In the second

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

Table 1. Triple description for optimal policies.

λ	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4
x	off	off	off	off	idle	idle	idle	idle	idle	idle	idle	idle
y	6	7	6	6								
z	9	10	9	9	7	6	5	5	4	4	4	3

experiment, we compare the optimal policy to simple (potentially sub-optimal) policies that only utilize one of the control mechanisms and analyze how the energy and holding cost contribute to the total cost. Then, we explore the benefit of short warming periods.

We use the following rates in the first two experiments: $\mu_{low} = 1$, $\mu_{high} = 3$, and $\gamma = 0.5$. We assume a linear holding cost function where $h(i) = 0.5i$. Finally, we let $c_u = 10$, $c_w = 20$ and $c_{low} = c_u + 0.5\mu_{low}^3$ and $c_{high} = c_u + \mu_{high}^3$. The choice of a cubic cost function is motivated by the physics of server farm energy consumption as described by Chen et al. [6]. We use a buffer size 500 as an approximation to an infinite buffer.

4.1. Value of On/Off Control

The first experiment compares the optimal policy for systems with arrival rates between 0.2 and 2.4. Table 1 contains the characterizing triple for each optimal policy. Recall x gives the optimal action taken in state $(0, 1)$ and y is the smallest queue length where the system begins warming. Note y is not provided if we reach $(0, 1)$ and the server idles since the server does not turn off making states where the server is off transient. The threshold z is the shortest queue length where μ_{high} is used. Note since $n = 2$, z is a scalar as opposed to the vector described in Section 3. The most notable observation from the experiment is how low the traffic intensity must be before it is beneficial to turn the server off. We expect the ability to turn the server off to be most useful when the intensity is low since that system could also potentially spend the most time idle. However, it is somewhat surprising that avoiding delays, even for moderate traffic intensity where holding cost contributes relatively little to the total cost, is preferred by the optimal policy even though it results in additional costs due to idling. This is illustrated in Figure 3.

4.2. Comparison to Single Control Policy

Our second experiment compares the average cost of the optimal policy to (possibly sub-optimal) policies that do not utilize both control mechanisms. The sub-optimal policies that we compare are:

(1) High rate always: This policy always uses rate μ_{high} regardless of the number in the system and never turns off. Accordingly, in the long-run, the system under this policy behaves as an $M/M/1$ queue with service rate μ_{high}.

(2) Turn off at 0: This policy uses the same rate threshold as the optimal policy but turns off whenever the system is empty. It begins warming as soon as a job arrives.

The average cost for a policy is calculated by simulating the system under the policy 100 times. Each simulation has a burn in time of 1000, then the next 10000 decision epochs are used to compute average cost. As when finding the optimal policy, we use a buffer size of 500.

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

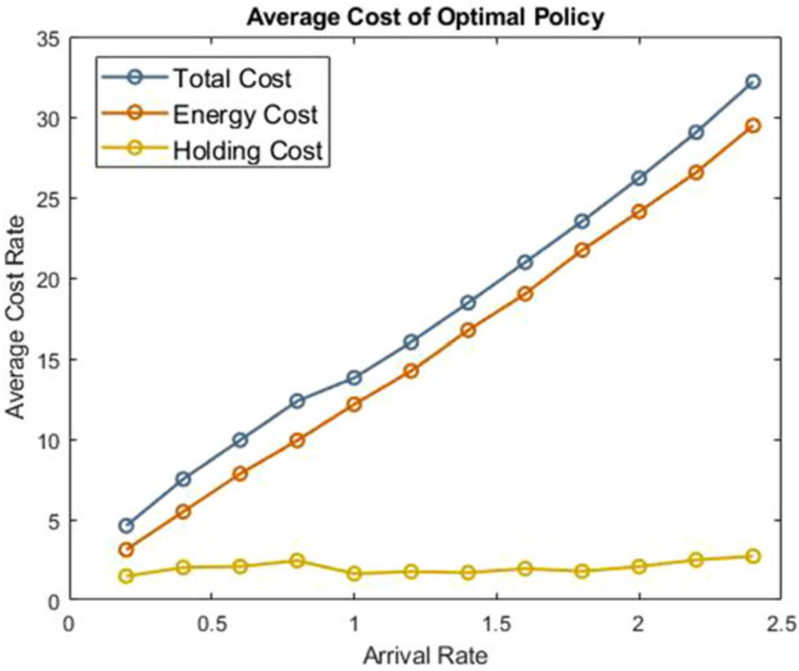

FIGURE 3. The average cost rate under the optimal policy for increasing arrival rates is split into the component costs incurred from energy costs (serving, idling, or warming) and from holding costs.

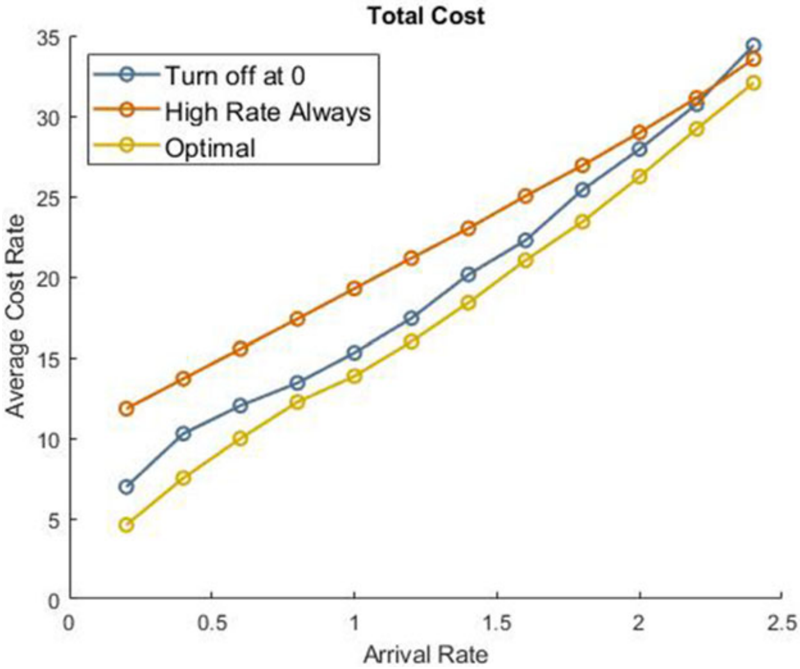

FIGURE 4. The average cost rate under the optimal policy for increasing arrival rates.

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

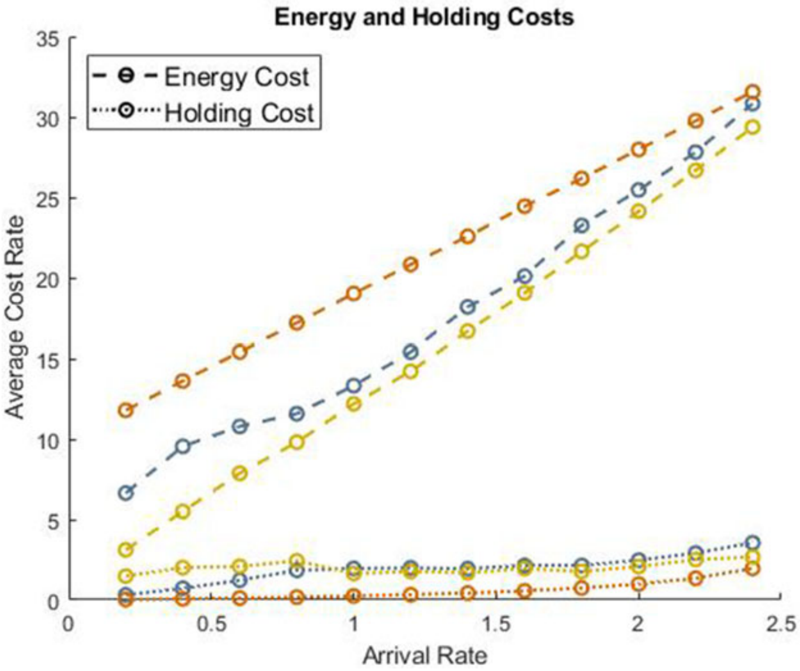

FIGURE 5. The average cost rate under the optimal policy for increasing arrival rates is split into the component costs incurred from energy cost (serving, idling, or warming) and resulting from holding costs.

TABLE 2. Average cost incurred per period

λ	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0	2.2	2.4
Optimal	4.6	7.5	10.0	12.2	13.9	16.0	18.4	21.0	23.4	26.2	29.2	32.1
High rate	11.8	13.7	15.6	17.4	19.3	21.2	23.0	25.0	26.9	29.0	31.1	33.5
Off at 0	7.0	10.3	12.0	13.4	15.3	17.4	20.1	22.3	25.4	28.0	30.7	34.4

The benefit of both controls is most pronounced when the traffic intensity is low, as shown in Figure 4. This is not surprising since when the traffic intensity is high, we expect the higher rate should be used more often and the queue is empty infrequently, thus the differences between the optimal policy and other two are less pronounced.

As observed in Figure 5, the most useful feature of this experiment is the clear trade off between holding costs and delay. For $\lambda = 1$ the reduction in energy costs no longer makes up for longer delays resulting in higher holding costs. As noted in the previous experiment, it is somewhat surprising idling is optimal even for moderate to low traffic intensity since holding costs make up less then 20% of the total cost (Table 2).

Not surprisingly the system which always uses rate μ_{high} incurs the least holding cost for all arrival rates, however utilizing the lower rate significantly decreased energy costs particularly for lower traffic intensity, for instance when $\lambda = 1$, the optimal policy incurs nearly 40% less costs.

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

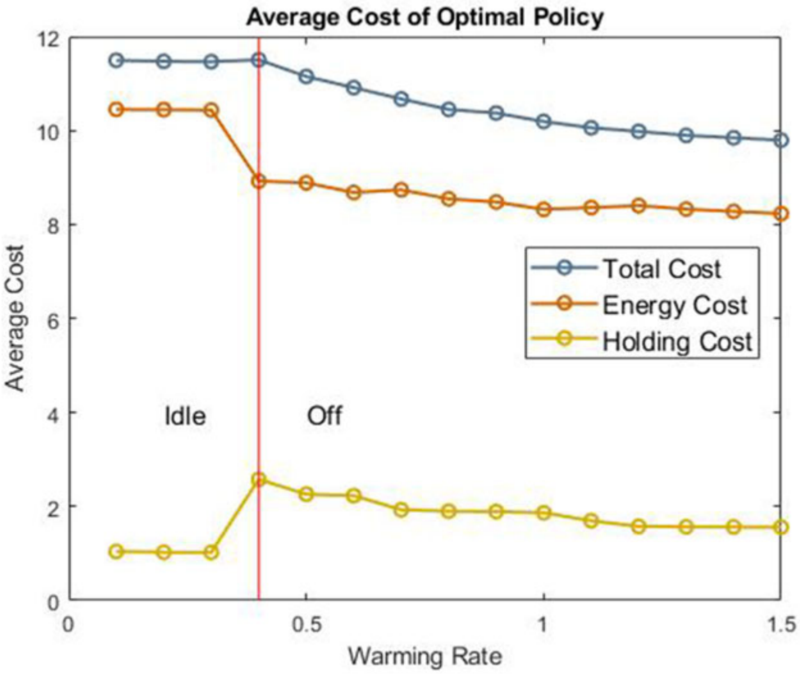

FIGURE 6. The average cost rate under the optimal policy for increasing warming rate and arrival rate $\lambda = 0.7$

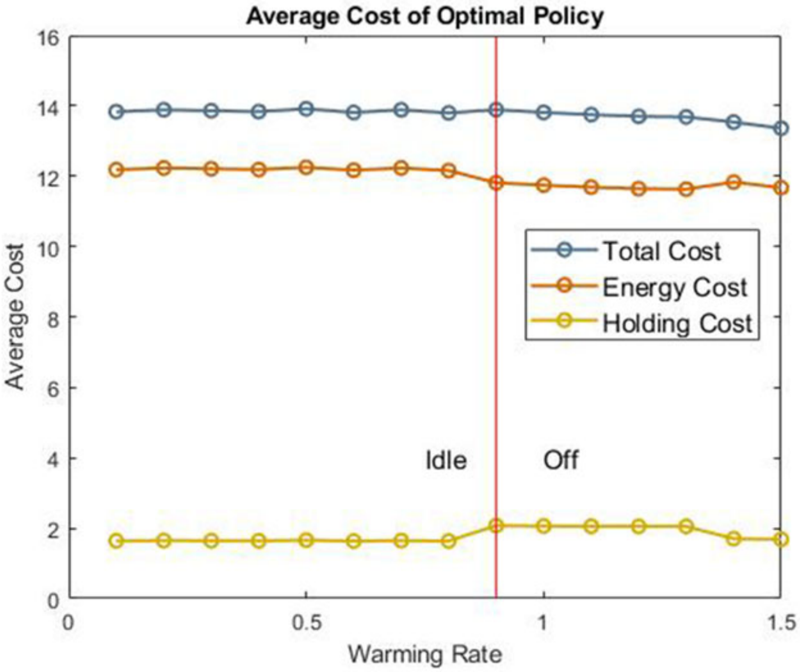

FIGURE 7. The average cost rate under the optimal policy for increasing warming rate and arrival rate $\lambda = 1$.

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

4.3. Value of Fast Warming

Finally we explored the value of faster warming rates. While not useful to a manager once the system is in use, this experiment aims to provide insights into the value of investing in faster warming times when the system is being built. This experiment uses the same inputs as the previous two except the arrival rate is $\lambda = 0.7$ and the warming rate varies from $\gamma = 0.1$ to $\gamma = 1.5$.

As expected, faster warming rates decrease total cost. More notable however, is the change in the proportion of costs incurred from energy and holding costs when the action employed by the optimal policy when the system empties switches from idling to turning off. We see this in Figure 6, where the average cost changes negligibly between when the warming rate is 0.3 (so the server idles) and when the warming rate is 0.4 (and the server turns off) but the proportion due to the holding costs increases significantly. This implies that increasing the warming rate a small amount might not be sufficient to decrease total cost but it may be beneficial if lowering costs due to energy usage is a priority. However it is worth noting that this effect diminishes as the arrival rate increases. In Figure 7, where $\lambda = 1$, we see a much less dramatic trade off when the action taken when the system empties changes.

5. CONCLUSION

In this paper we considered a system with a single removable server that dynamically chooses its service rate. We proposed an MDP model for this system and analyzed it under the average cost optimality criterion. We used a renewal reward approach and renewal arguments using a probabilistic interpretation of the relative value function to characterize an average cost optimal policy. Some interesting paths for further work include:

(1) Including switching costs in the model. In reality it may not be possible to dynamically change service rates instantly or without cost. This would also penalize switching, which in the data center application, can result in a reduction in server reliability.

(2) Multiple servers. A multiple server model of this problem is clearly of interest since data centers are not made up of a single server. Multiple server models also introduce the added complexity of routing jobs to appropriate servers.

References

1. Baker, K. (1973). A note on operating policies for the queue M/M/1 with exponential startups. *INFOR: Information Systems and Operational Research* 11(1): 71–72.
2. Barr, J. (2015). Cloud computing, server utilization, and the environment. https://aws.amazon.com/blogs/aws/cloud-computing-server-utilization-the-environment/, June 2015. Last Accessed : 11 September 2018.
3. Barroso, L.A. & Hölzle, U. (2007). The case for energy-proportional computing. *IEEE Computer* 40(12): 33–37.
4. Bell, C.E. (1971). Characterization and computation of optimal policies for operating an M/G/1 queuing system with removable server. *Operations Research* 19(1): 208–218.
5. Borthakur, A., Medhi, J., & Gohain, R. (1987). Poisson input queueing system with startup time and under control-operating policy. *Computers & Operations Research* 14(1): 33–40.
6. Chen, Y., Das, A., Qin, W., Sivasubramaniam, A., Wang, Q., & Gautam, N. (2005). Managing server energy and operational costs in hosting centers. *ACM SIGMETRICS performance evaluation review* 33(1): 303–314.

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

7. Chong, K.C., Henderson, S.G., & Lewis, M.E. (2018). Two-class routing with admission control and strict priorities. *Probability in the Engineering and Informational Sciences* 32(2): 163–178.
8. Crabill, T. B. (1974). Optimal control of a maintenance system with variable service rates. *Operations Research* 22(4): 736–745.
9. Delforge, P. & Whitney, J. (2014). Issue paper: data center efficiency assessment scaling up energy efficiency across the data center industry: evaluating key drivers and barriers. *Natural Resource Defense Council (NRDC)*, August 2014.
10. Dimitrakopoulos, Y. & Burnetas, A. (2017). The value of service rate flexibility in an M/M/1 queue with admission control. *IISE Transactions* 49(6): 603–621.
11. Federgruen, A. & So, K.C. (1991). Optimality of threshold policies in single-server queueing systems with server vacations. *Advances in Applied Probability* 23(2): 388–405.
12. Feinberg, E.A. & Kella, O. (2002). Optimality of D-policies for an M/G/1 queue with a removable server. *Queueing Systems* 42(4): 355–376.
13. Gandhi, A., Gupta, V., Harchol-Balter, M., & Kozuch, M.A. (2010). Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation* 67(11): 1155–1171.
14. Gandhi, A., Harchol-Balter, M., & Adan, I. (2010). Server farms with setup costs. *Performance Evaluation* 67(11): 1123–1138.
15. Gebrehiwot, M.E., Aalto, S.A., & Lassila, P. (2014). Optimal sleep-state control of energy-aware M/G/1 queues. *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*, pp. 82–89.
16. George, J.M. & Harrison, J.M. (2001). Dynamic control of a queue with adjustable service rate. *Operations Research* 49(5): 720–731.
17. Heyman, D.P. (1968). Optimal operating policies for M/G/1 queuing systems. *Operations Research* 16(12): 362–382.
18. Ke, J.-C. (2003). The optimal control of an M/G/1 queueing system with server startup and two vacation types. *Applied Mathematical Modelling* 27(6): 437–450.
19. Koole, G. (1998). Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems* 30(3-4): 323–339.
20. Kumar, R., Lewis, M.E., & Topaloglu, H. (2013). Dynamic service rate control for a single-server queue with markov-modulated arrivals. *Naval Research Logistics (NRL)* 60(8): 661–677.
21. Lippman, S.A. (1975). Applying a new device in the optimization of exponential queuing systems. *Operations Research* 23(4): 687–710.
22. Maccio, V.J. & Down, D.G. (2015). On optimal control for energy-aware queueing systems. In *Teletraffic Congress (ITC 27), 27th International*, pages 98–106. IEEE, 2015.
23. Maccio, V.J. & Down, D.G. (2015). On optimal policies for energy-aware servers. *Performance Evaluation* 90: 36–52.
24. Rudin, W. (1976). *Principles of mathematical analysis*, Vol. 3, New York: McGraw-Hill.
25. Sennott, L.I. (1999). *Stochastic dynamic programming and the control of queueing systems*. New York: John Wiley & Sons.
26. Shehabi, A., Smith, S.J., Sartor, D.A., Brown, R.E., Herrlin, M., Koomey, J.G., Masanet, E.R., Horner, N., Azevedo, I.L., & Lintner, W. (2016). United states data center energy usage report. *Lawrence Berkeley National Laboratory*, June 2016.
27. Wang, K.-H., Wang, T.-Y., & Pearn, W.L. (2007). Optimal control of the N-policy M/G/1 queueing system with server breakdowns and general startup times. *Applied Mathematical Modelling* 31(10): 2199–2212.
28. Yadin, M., & Naor, P. (1963). Queueing systems with a removable service station. *Journal of the Operational Research Society* 14(4): 393–405.
29. Yoon, S., & Lewis, M.E. (2004). Optimal pricing and admission control in a queueing system with periodically varying parameters. *Queueing Systems* 47(3): 177–199.

APPENDIX A

We include the following results for two reasons. First, though not necessary for the main results, interested readers may reasonably ask if indeed the relative value function is non-decreasing. Second, we think the proof technique, in which we introduce additional actions so that we may construct coupled policies, is interesting and applicable in other instances.

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press

In order to do show if $r(i,j)$ is a relative value function satisfying the (ACOE) it is non-decreasing, we will allow the actions $a = \{(k, \mathit{off}), (k, \mathit{on}),\ k = 1, \ldots, n\}$ to be taken in state $(0, 1)$. Since there is no work to be done in this state, taking one of these actions means the server idles but incurs the cost of serving. If $a = (k, \mathit{off})$, $k \in \{1, \ldots, n\}$, following a hypothetical service, the server turns off. Clearly these actions are sub-optimal as the server should either idle and never be turned off or should have turned off immediately after the system empties. We show this formally in Lemma A.1.

Lemma A.1: *There exists an optimal policy that does not use actions* $a = \{(k, \mathit{off}), (k, \mathit{on}),$ $k = 1, \ldots, n\}$ *in state* $(0, 1)$.

Proof: First note, this case need only be considered if under an optimal policy $\pi = d^\infty$, $d(1, 1) = (k, \mathit{on})$ for some $k \in \{1, \ldots, n\}$. Otherwise, state $(0, 1)$ is transient and the action taken there does not affect the long-run average cost. Thus $r(0, 0) \geq r(0, 1)$, otherwise π does not attain the minimum in the (ACOE) and $d(1, 1) = (k, \mathit{on})$ is not optimal. Suppose $a = (k, \mathit{off})$ attains the minimum in the (ACOE) for $(0, 1)$ and is strictly better than $(\mathit{idle}, \mathit{off})$. Thus,

$$\begin{aligned} c_k + \lambda r(1,1) + \mu_k r(0,0) + (1-\lambda-\mu_k) r(0,1) &< c_u + \lambda r(1,1) + (1-\lambda) r(0,1) \\ \implies c_k - c_u + \mu_k [r(0,0) - r(0,1)] &< 0 \end{aligned}$$

This is a contradiction; (k, off) need not be used in an optimal policy. Similarly, suppose $a = (k, \mathit{on})$ attains the minimum in the (ACOE) and is strictly better than $(\mathit{idle}, \mathit{off})$. This yields

$$\begin{aligned} c_k + \lambda r(1,1) + \mu_k r(0,1) + (1-\lambda-\mu_k) r(0,1) &< c_u + \lambda r(1,1) + (1-\lambda) r(0,1) \\ \implies c_k - c_u &< 0 \end{aligned}$$

Thus, again (k, on) need not be used in an optimal policy. ■

Lemma A.2: *Let* r *be a relative value function satisfying the* (ACOE). *Then* $r(i,j)$ *is non-decreasing in* i *for* $j = 0, 1$, *i.e.*

$$r(i+1, j) - r(i, j) \geq 0.$$

Proof: Recall that the difference between the relative value function evaluated at different initial states is the difference in the total cost under an average cost optimal policy. Consider two processes started on the same probability space so that they see the same arrivals and the same potential services. System 1 initializes in $(i+1, j)$ and uses an optimal policy and System 2 initializes in (i, j) and mimics the actions taken by System 1. Note by Lemma A.1 we can include the additional actions in $(0, 1)$ that allows System 2 to mimic System 1 in all states. Eventually the systems couple when a job is serviced in $(1, 1)$ in System 2 while System 1 idles (for the cost of serving). After this point the systems incur identical costs. Before coupling, System 2 incurs less cost since both systems use the same actions but System 2 incurs less holding costs as it has fewer jobs in the system. ■

https://doi.org/10.1017/S0269964819000299 Published online by Cambridge University Press