Aalto, Samuli; Lassila, Pasi

Near-optimal dispatching policy for energy-aware server clusters

Published in:
Performance Evaluation

DOI:
10.1016/j.peva.2019.102034

Published: 01/11/2019

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Aalto, S., & Lassila, P. (2019). Near-optimal dispatching policy for energy-aware server clusters. *Performance Evaluation*, *135*, Article 102034. https://doi.org/10.1016/j.peva.2019.102034

Near-optimal dispatching policy for energy-aware server clusters

Samuli Aalto, Pasi Lassila

Department of Communications and Networking, Aalto University, Finland

Abstract

A server cluster can be modeled as a set of parallel queues, and the dispatcher decides to which queue the arriving jobs are routed. We consider an energy-aware dispatching system in a Markovian setting, where each server, upon becoming empty, enters a sleep mode to save energy, and to activate the server after sleep incurs an additional setup delay cost. We seek to optimize the performance-energy trade-off by applying the so-called Whittle index approach. As our main result, we derive sufficient conditions for the system parameters under which the problem is provably indexable, and also determine the corresponding Whittle index values explicitly. Our numerical experiments demonstrate that the resulting energy-aware Whittle index policy is able to perform very close to the numerically solved optimal policy and outperforms all considered reference policies.

Keywords: Optimal dispatching, task assignment, server cluster, energy-aware server, InstantOff, Whittle index, indexability

1. Introduction

Server clusters processing huge volumes of computational jobs comprise the core of modern data centers and cloud computing systems. Hierarchical stochastic queuing models, such as multi-server systems with a central queue or distributed systems with multiple parallel servers and queues, are reasonable for the mathematical modeling of such server clusters [13], and allow fundamental insights to be obtained for optimal job processing.

In this paper, we consider a distributed system of parallel servers with their own queues where an arriving job is dispatched to one of the servers upon arrival. Dispatching (a.k.a. task assignment) problems belong to the

optimal control problems for parallel server systems [13]. However, there are very few exact optimality results available. Maybe the most well-known is the optimality of the Join-Shortest-Queue (JSQ) policy in a homogeneous setting where all the servers have the same service rate, originally proved for exponential service times in [31] and thereafter generalized to any service time distributions with a non-decreasing hazard rate in [27].

One approach presented in the literature to produce near-optimal dispatching policies is to utilize the policy iteration algorithm from the theory of Markov decision processes [25]. In policy iteration, the optimal policy is solved iteratively starting from a given initial policy. At each step, the policy is improved by optimizing in each state the action based on their immediate and future costs. In particular with Poisson arrivals, the first policy iteration (FPI) step can sometimes be made explicitly, see, e.g., [30, 3, 4, 15].

In all papers mentioned above, the servers are assumed to be ordinary ones alternating between busy and idle states. In this paper, we are, however, more interested in *energy-aware servers* that can be switched off (without any delay) to save energy but, on the other hand, incur a setup delay when switched back on. In [7], such servers were called *InstantOff* servers, while the ordinary ones were referred to as *NeverOff* servers. In [23, 10, 9, 11], it was shown, under various assumptions, that the optimal sleep state control policy for a single server is either InstantOff or NeverOff. The FPI approach has been applied also for the dispatching problem in the context of such energy-aware InstantOff servers, see, e.g., [18, 17, 16, 12, 19, 14].

In this paper, instead of the FPI approach, we apply the *Whittle index approach* to the energy-aware dispatching problem in order to generate alternative near-optimal control policies. This approach was originally developed in the context of *restless bandits* [29]. The idea is that the constrained optimization problem with exactly one activated bandit in each time epoch, which is known to be mathematically intractable, is first relaxed by allowing to activate a varying number of bandits and only requiring that one bandit is activated on average. This makes the problem much more tractable by decomposing it to separate subproblems per each bandit. The related Lagrangian relaxation parameter can be interpreted as the price of passivity. While the problem becomes separable, it is, however, not clear, a priori, whether the problem is *indexable*, or not. Briefly said, indexability means that, for each state of a single bandit, there is a unique threshold value of the Lagrangian relaxation parameter such that for lower [higher] values of this parameter the optimal decision in the relaxed problem is to be passive

[active]. In addition, such a unique threshold is called the *Whittle index* at this state of the related bandit. If the problem is indexable, the resulting Whittle index policy is known to be asymptotically optimal, at least under certain technical conditions [28, 26].

The Whittle index approach has successfully been applied, e.g., to the opportunistic scheduling problems in [5, 21, 6, 2]. To the dispatching problem, it has also been successfully applied, e.g., in [24, 4, 22], however, without the energy aspect. Niño-Mora [24] managed to apply the Whittle index approach to the dispatching problem in a successful way when the queues behave as one-dimensional birth-death processes with a finite state space. Argon et al. [4] assumed that the queues behave as ordinary M/M/1 queues with an infinite state space. They also managed to show the indexability property for a large class of cost functions (including the linear holding costs) and derive the corresponding index values. Larrañaga et al. [22] considered dispatching problems where the queues behave as one-dimensional birth-death processes with an infinite state space. They characterized the index under the conjecture that the optimal policy is of threshold type.

Our target in this paper is to derive near-optimal policies by applying the Whittle index approach for dispatching problems where the queues behave as energy-aware M/M/1 queues provided with InstantOff servers, which is an essentially *more complicated* task than for the ordinary M/M/1 queues (or more general birth-death processes) due to the two-dimensional state space. To study the performance-energy trade-off in such systems, we assume energy costs in addition to normal linear holding costs. We derive sufficient and realistic conditions for the system parameters under which the problem is provably indexable and also determine the corresponding index values explicitly.

An earlier version of the paper appeared in Proceedings of ITC 30 [1], where we proved the indexability property for the systems with sufficiently short setup delays, which, however, may not be a realistic assumption. In the present paper, we manage to find sufficient conditions for indexability that allow longer (and, thus, more realistic) setup delays. The original proof of indexability presented in [1] relied on establishing a certain *ordering of the states* in the two-dimensional state space. In the present paper, we show that a set of successively looser conditions on the mean length of the setup delay can be determined under which the problem still remains indexable. The challenge in the proof is that the ordering of the states in the two-dimensional state space changes along the system parameters. Our conditions essentially

mean that for fixed values of all other system parameters except the total arrival rate of jobs, there is always a lower bound on the total arrival rate such that indexability is guaranteed above that. In other words, indexability is guaranteed if the total load is high enough.

The performance of the Whittle index policy is illustrated by an extensive set of numerical experiments. For a small system with only two servers, we demonstrate that the resulting energy-aware Whittle index policy is indeed able to perform very close to the numerically obtained optimal policy and outperforms all other reference policies (FPI, JSQ, and the static load balancing policy). Similarly, in several scenarios involving a larger system with 10 servers, the Whittle index policy always performs best with respect to the holding costs, but typically the FPI policy can yield a lower energy consumption. However, with respect to the total costs, the Whittle index policy is clearly better than any of the reference policies in all our experiments.

The rest of the paper is organized as follows. The energy-aware dispatching problem and the Whittle index approach to tackle it by utilizing a relaxation of the original problem, are described in more detail in Sections 2 and 3, respectively. In Section 4, we describe how to get sufficient conditions for indexability, and present the main result of the paper, which includes explicit expressions for the related Whittle index values. The main result is thereafter proved piecewise in Sections 5 and 6. In Section 7, we introduce the energy-aware Whittle index policy for the original dispatching problem, and compare it numerically with the FPI dispatcing policy and the ordinary JSQ rule in Section 8. Finally, Section 9 concludes the paper.

The paper is provided with Supplemetary material available online, which includes the proofs of the main results and also some auxiliary results with their proofs, which play a central role in the proof of the main results.

2. Energy-aware dispatching problem

We consider the following energy-aware dispatching problem. New jobs arrive according to a Poisson process with rate λ. At the arrival time, the job is dispatched to one of K parallel servers, each provided with an infinite buffer. Each server i is an exponential server with rate μ_i, i.e., the service time of any job in this server is independently and exponentially distributed with mean $E[S_i] = 1/\mu_i$. A necessary stability condition for such a system is given by $\lambda < \sum_{i=1}^{K} \mu_i$.

The server is said to be *busy* when it is processing jobs. When server i has processed all the jobs and its buffer becomes empty, it is immediately switched *off*. Server i remains switched-off until a new job is dispatched to it, after which it still needs an exponential *setup* phase with mean $E[D_i] = 1/\delta_i$, before becoming busy again. In line with [7], such servers are called *InstantOff* servers in this paper. We also introduce here the following shorthand notations used later on in this paper: $\sigma_i = \lambda/\delta_i$ and $\rho_i = \lambda/\mu_i$.

The state of server i at time t is described by the pair $(N_i(t), Z_i(t))$, where $N_i(t) \in \mathcal{N} = \{0, 1, \ldots\}$ denotes the number of customers and $Z_i(t) \in \mathcal{Z} = \{\text{off}, \text{setup}, \text{busy}\}$ the energy state. Let $P_i(z) \geq 0$ denote the (constant) power consumption in energy state z. It is natural to assume that

$$0 \leq P_i(\text{off}) < P_i(\text{setup}) \leq P_i(\text{busy}). \tag{1}$$

In addition, we introduce the following differential notation for $z \in \{\text{setup}, \text{busy}\}$:

$$\hat{P}_i(z) = P_i(z) - P_i(\text{off}) > 0. \tag{2}$$

With each server i and time t, we also associate a decision variable $A_i(t) \in \mathcal{A} = \{0, 1\}$. If $A_i(t) = 1$, then the next arriving customer is dispatched to server i; otherwise not. Naturally, we require that, for any t,

$$\sum_{i=1}^{K} A_i(t) = 1. \tag{3}$$

At time t, server i incurs costs at rate

$$C_i(t) = h_i N_i(t) + \beta P_i(Z_i(t)), \tag{4}$$

where $h_i > 0$ is the holding cost rate per job and $\beta \geq 0$ is an energy weight factor. The problem is to choose the decision variables $A_i(t)$ in such a way that the expected long-run average cost,

$$\lim_{T\to\infty} E\left[\frac{1}{T}\int_0^T \left(\sum_{i=1}^{K}(h_i N_i(t) + \beta P_i(Z_i(t)))\right) dt\right], \tag{5}$$

is minimized, subject to constraint (3) for all t. The problem is considered in the context of continuous-time Markov decision processes (CTMDP) [25],

where the decision variables $A_i(t)$ can only be changed when the state of the whole system,

$$((N_i(t), Z_i(t)) \mid i \in \{1, \ldots, K\}),$$

changes.

Note that, without constraint (3), the problem is trivially solved by choosing passivity: $A_i(t) = 0$ for all i and t. Including constraint (3), however, makes the optimal dispatching problem extremely hard. In the following section, we describe how the Whittle index approach can be utilized to tackle it and to derive near-optimal dispatching policies.

3. Relaxed optimization problem

Following the ideas originally developed by Whittle in [29], we relax the dispatching problem (5) by replacing the strict constraint (3) with an averaged one,

$$\lim_{T\to\infty} E\left[\frac{1}{T}\int_0^T \left(\sum_{i=1}^K A_i(t)\right)\, dt\right] = 1, \tag{6}$$

and approach the relaxed problem by Lagrangian methods.[1] As a result, we get the following *separate* subproblems. For each server i, we try to minimize the objective function

$$\lim_{T\to\infty} E\left[\frac{1}{T}\int_0^T \left(h_i N_i(t) + \beta P_i(Z_i(t)) + \nu(1 - A_i(t))\right)\, dt\right], \tag{7}$$

where the Lagrangian multiplier ν can be interpreted as the *price of passivity* per time unit.[2] We further note that, if $\nu \leq 0$, then there is no need to be active so that $A_i(t) = 0$ is optimal for any t. Thus, from this on, we assume that $\nu > 0$.

Let us now define the continuous-time Markov decision process related to the separate subproblems with objective function (7). The state of the "system" (i.e., server i) is given by the pair $x = (n, z)$, where $n \in \mathcal{N}$ refers to the number of customers and $z \in \mathcal{Z}$ to the energy state, and the possible

[1] Note that under the relaxed policy an arriving job is permitted to be allocated to more than one server simultaneously, which is something that cannot happen in practice.

[2] Note a slight difference with [24], where the roles of activity and passivity are reversed and the Lagrangian multiplier ν is interpreted as the price of rejection per arriving job.

actions $a \in \mathcal{A}$ are "to dispatch" ($a = 1$) and "not to dispatch" ($a = 0$). Thus, the state space

$$\mathcal{X} = \{(0, \text{off})\} \cup \{(n, \text{setup}) \mid n \geq 1\} \cup \{(n, \text{busy}) \mid n \geq 1\}$$

is discrete and the action space $\mathcal{A}$ finite. The stationary deterministic policy π_i^A for server i that corresponds to the *activity set* $A \subset \mathcal{X}$ is defined on $\mathcal{X}$ as follows:

$$\pi_i^A(x) = \begin{cases} 1, & \text{if } x \in A; \\ 0, & \text{otherwise.} \end{cases}$$

Let then $c_i(x, a; \nu)$ denote the cost rate in state $x \in \mathcal{X}$ after action $a \in \mathcal{A}$ with the price of passivity fixed to ν. In our model, we have, for any $x = (n, z) \in \mathcal{X}$ and $a \in \mathcal{A}$,

$$c_i(x, a; \nu) = h_i n + \beta P_i(z) + \nu(1 - a).$$

In addition, let $q_i(y|x, a) \geq 0$ denote the transition intensity from state $x \in \mathcal{X}$ to another state $y \in \mathcal{X} \setminus \{x\}$ after action $a \in \mathcal{A}$. In our model, the following transitions are possible:

$$\begin{array}{ll} q_i((1, \text{setup})|(0, \text{off}), a) = a\lambda, & \\ q_i((n, \text{busy})|(n, \text{setup}), a) = \delta_i, & n \geq 1, \\ q_i((n+1, \text{setup})|(n, \text{setup}), a) = a\lambda, & n \geq 1, \\ q_i((0, \text{off})|(1, \text{busy}), a) = \mu_i, & \\ q_i((n-1, \text{busy})|(n, \text{busy}), a) = \mu_i, & n \geq 2, \\ q_i((n+1, \text{busy})|(n, \text{busy}), a) = a\lambda, & n \geq 1. \end{array}$$

These state transitions are illustrated in Figure 1.

Since the state space $\mathcal{X}$ is discrete, the action space $\mathcal{A}$ finite, and the cost rate linear with respect to n, there is a stationary deterministic policy π_i^* (defined by an optimal set of active states, $A_i^* \subset \mathcal{X}$) that minimizes the expected average costs (7) [25]. The optimal policy π_i^* is characterized by the *optimality equations* defined for each state $x \in \mathcal{X}$ by

$$\bar{c}_i(\nu) = \min_{a \in \mathcal{A}} \left\{ c_i(x, a; \nu) + \sum_{y \neq x} q_i(y|x, a)(v_i(y; \nu) - v_i(x; \nu)) \right\}, \qquad (8)$$

where $\bar{c}_i(\nu)$ denotes the minimum expected average cost rate (per time unit) and $v_i(x; \nu)$ refers to the value function, which gives the difference in the expected total costs when the optimal stationary policy is applied and the

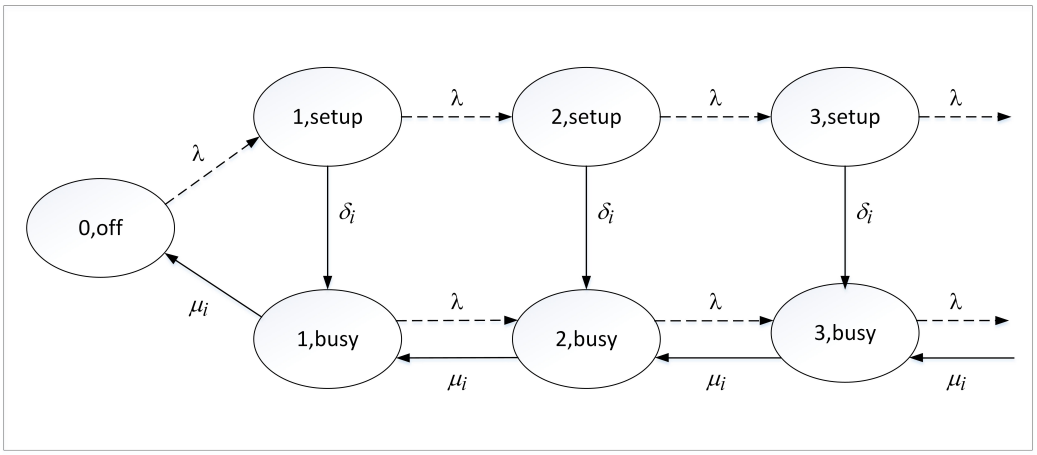

Figure 1: State transition diagram of our model. Straight lines represent transitions that are possible for both actions $a = 0$ and $a = 1$. Dashed lines represent transitions that are possible only for action $a = 1$.

system is started from state x or in equilibrium. The main task here is, for each ν, to find a real value $\bar{c}_i(\nu)$ and function $v_i(x;\nu)$ satisfying the optimality equation (8).

In our model, the optimality equations (8) read as follows. For state $x = (0, \text{off})$,

$$\bar{c}_i(\nu) = \beta P_i(\text{off}) + \min\{\nu, \lambda\Delta_i(0, \text{off}; \nu)\}, \tag{9}$$

where we have defined

$$\Delta_i(0, \text{off}; \nu) = v_i(1, \text{setup}; \nu) - v_i(0, \text{off}; \nu).$$

For states $x \in \{(n, \text{setup}) \mid n \geq 1\}$,

$$\begin{aligned}\bar{c}_i(\nu) = h_i n + \beta P_i(\text{setup}) + \delta_i(v_i(n, \text{busy}; \nu) - v_i(n, \text{setup}; \nu)) + \\ \min\{\nu, \lambda\Delta_i(n, \text{setup}; \nu)\},\end{aligned} \tag{10}$$

where we have defined

$$\Delta_i(n, \text{setup}; \nu) = v_i(n+1, \text{setup}; \nu) - v_i(n, \text{setup}; \nu).$$

For state $x = (1, \text{busy})$,

$$\begin{aligned}\bar{c}_i(\nu) = h_i + \beta P_i(\text{busy}) + \mu_i(v_i(0, \text{off}; \nu) - v_i(1, \text{busy}; \nu)) + \\ \min\{\nu, \lambda\Delta_i(1, \text{busy}; \nu)\},\end{aligned} \tag{11}$$

where we have defined

$$\Delta_i(1, \text{busy}; \nu) = v_i(2, \text{busy}; \nu) - v_i(1, \text{busy}; \nu).$$

For states $x \in \{(n, \text{busy}) \mid n \geq 2\}$,

$$\begin{aligned} \bar{c}_i(\nu) = h_i n + \beta P_i(\text{busy}) - \mu_i \Delta_i(n-1, \text{busy}; \nu) + \\ \min\{\nu, \lambda \Delta_i(n, \text{busy}; \nu)\}, \end{aligned} \tag{12}$$

where we have defined

$$\Delta_i(n, \text{busy}; \nu) = v_i(n+1, \text{busy}; \nu) - v_i(n, \text{busy}; \nu).$$

Thus, from the optimality equations (9)–(12), we deduce that, for any state $x \in \mathcal{X}$, we have the following alternatives for the optimal decision in state x:

(i) if $\nu > \lambda \Delta_i(x; \nu)$, then
it is optimal to dispatch ($a = 1$) the next job to server i;
(ii) if $\nu = \lambda \Delta_i(x; \nu)$, then
both decisions ($a = 0$ and $a = 1$) are equally good and optimal;
(iii) if $\nu < \lambda \Delta_i(x; \nu)$, then
it is optimal not to dispatch ($a = 0$) the next job to server i. (13)

We conclude this section by defining when this optimization problem is indexable.

Definition 1. *We say that the optimization problem with objective function (7) is* indexable[3] *if, for any state $x \in \mathcal{X}$, there exists $\nu_i^*(x) \in [-\infty, \infty]$ such that*

(i) it is optimal to dispatch ($a = 1$) the next job to server i in state x if and only if $\nu \geq \nu_i^*(x)$;

(ii) it is optimal not to dispatch ($a = 0$) the next job to server i in state x if and only if $\nu \leq \nu_i^*(x)$.

Such a value $\nu_i^*(x)$ is referred to as the Whittle index *of state x for the problem with objective function (7).*

Note that, according to this definition, the two actions are equally good (and, thus, optimal) if $\nu = \nu_i^*(x)$.

[3]Note that we have adapted Whittle's notation of indexability [29] to our dispatching problem in a similar way as done in [4]. As a result, the heuristic index policy for the original problem dispatches the arriving job to the server with the *lowest* index.

4. Sufficient conditions for indexability, and the main result

Typically, the first problem in the Whittle index approach is to show that the Lagrangian version of the relaxed problem is indexable [29]. In [1], we managed to prove the indexability property for the system parameter values that satisfy the following condition:

$$\delta_i > \mu_i \left(1 + \frac{\beta}{h_i}\hat{P}_i(\text{setup})\right). \tag{14}$$

Note that this condition is valid only if the mean setup delay $E[D_i] = 1/\delta_i$ is *sufficiently short*, which may be not that realistic an assumption. So, in this paper, we aim at finding sufficient conditions for indexability that allow *longer* (and, thus, more realistic) setup delays.

In this section, we describe how to get sufficient conditions for indexability, and present the main result of the paper (Theorem 1). In the forthcoming sections, we prove that those conditions are, indeed, sufficient for indexability and also determine the corresponding Whittle index values explicitly. At the end of this section, we give an illustrative example on the behavior of the resulting Whittle index values (Example 1).

Let $n \in \{0, 1, \ldots\}$ be fixed (for a while), and consider a stationary deterministic policy $\pi_i^{A_n}$ for server i defined by the activity set A_n, where

$$A_n = \{(m, \text{busy}) \in \mathcal{X} \mid m \leq n\}.$$

Note that $A_0 = \emptyset$. Denote $\pi_i^{A_n}$ here briefly by π. In addition, let $v_i^\pi(x;\nu)$ denote the value function for policy π. By the so-called Howard equations, it is a tedious but still straightforward task to derive the following value function differences (see Equation (A.8) in Appendix A, which can be found in supplementary material of this paper):

$$\begin{aligned}
&\Delta_i^\pi(0, \text{off};\nu) = v_i^\pi(1, \text{setup};\nu) - v_i^\pi(0, \text{off};\nu) = \\
&\quad h_i\left(\tfrac{1}{\delta_i} + \tfrac{1}{\mu_i}\textstyle\sum_{j=0}^{n}(j+1)\rho_i^j\right) + \beta\left(\hat{P}_i(\text{setup})\,\tfrac{1}{\delta_i} + \hat{P}_i(\text{busy})\,\tfrac{1}{\mu_i}\textstyle\sum_{j=0}^{n}\rho_i^j\right) - \\
&\quad \tfrac{\nu}{\mu_i}\textstyle\sum_{j=0}^{n-1}\rho_i^j, \\
&\Delta_i^\pi(n+1, \text{busy};\nu) = v_i^\pi(n+2, \text{busy};\nu) - v_i^\pi(n+1, \text{busy};\nu) = \\
&\quad \tfrac{1}{\mu_i}\left((n+2)h_i + \beta\hat{P}_i(\text{busy})\right).
\end{aligned}$$

Let us now require that

$$\lambda\Delta_i^\pi(0, \text{off};\nu) = \lambda\Delta_i^\pi(n+1, \text{busy};\nu) = \nu. \tag{15}$$

The idea of condition (15) is as follows: From equation (13), we see that if policy $\pi = \pi_i^{A_n}$ is optimal, then the requirement $\lambda\Delta_i^\pi(n+1, \text{busy}; \nu) = \nu$ results in such a combination of parameter values, where both decisions ($a = 0$ and $a = 1$) are equally qood in state $(n+1, \text{busy})$. It follows that, with these parameter values, policies $\pi_i^{A_n}$ and $\pi_i^{A_{n+1}}$ are both optimal. Similarly, from equation (13), we see that if policy π is optimal, then the requirement $\lambda\Delta_i^\pi(0, \text{off}; \nu) = \nu$ results in such a combination of parameter values, where both decisions ($a = 0$ and $a = 1$) are equally qood in state $(0, \text{off})$. It follows that, with these parameter values, policies $\pi_i^{A_n}$ and $\pi_i^{B_n}$ are both optimal, where we have defined $B_n = A_n \cup \{(0, \text{off})\}$. Thus, if policy $\pi = \pi_i^{A_n}$ is optimal and condition (15) is satisfied with some parameter combination, then all three policies $\pi_i^{A_n}$, $\pi_i^{A_{n+1}}$, and $\pi_i^{B_n}$ are optimal for such a parameter combination. From condition (15) we are able to solve (any) two of the parameters as a function of the other parameters. For δ_i, we get the following solution:

$$\delta_i = \frac{\mu_i \left(1 + \frac{\beta}{h_i}\hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{n}(n+1-j)\rho_i^j}.$$

Let us denote this solution by $a_{i,n}$ so that

$$a_{i,n} = \frac{\mu_i \left(1 + \frac{\beta}{h_i}\hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{n}(n+1-j)\rho_i^j}, \quad n \in \{0, 1, \ldots\}. \tag{16}$$

It is easy to see that the sequence $(a_{i,n})$ is strictly decreasing and converging to 0:

$$a_{i,0} > a_{i,1} > a_{i,2} > \ldots > 0. \tag{17}$$

Let then $n \in \{1, 2, \ldots\}$ be fixed (for a while), and consider now another stationary deterministic policy $\pi_i^{B_n}$ for server i defined by the activity set B_n, where

$$B_n = \{(0, \text{off})\} \cup \{(m, \text{busy}) \in \mathcal{X} \mid m \leq n\}.$$

Denote $\pi_i^{B_n}$ here briefly by π. Again by the Howard equations, it is possible to derive the following value function differences (see Equation (A.11) in

Appendix A, which can be found in supplementary material of this paper):

$$\Delta_i^\pi(1,\text{setup};\nu) = v_i^\pi(2,\text{setup};\nu) - v_i^\pi(1,\text{setup};\nu) =$$
$$h_i\left(\tfrac{1}{\delta_i} + \tfrac{1}{\mu_i}\sum_{j=0}^{n-1}(j+2)\rho_i^j\right) + \beta\left(\tfrac{P_i(\text{busy})}{\mu_i}\sum_{j=0}^{n-1}\rho_i^j\right) + \tfrac{\nu}{\mu_i}\rho_i^{n-1} - \tfrac{\bar{c}_i^\pi(\nu)}{\mu_i}\sum_{j=0}^{n-1}\rho_i^j,$$
$$\Delta_i^\pi(n+1,\text{busy};\nu) = v_i^\pi(n+2,\text{busy};\nu) - v_i^\pi(n+1,\text{busy};\nu) =$$
$$\tfrac{1}{\mu_i}\left((n+2)h_i + \beta P_i(\text{busy}) + \nu - \bar{c}_i^\pi(\nu)\right),$$

where $\bar{c}_i^\pi(\nu)$ denotes the expected average cost rate (per time unit) for policy π.

Let us now require that

$$\lambda\Delta_i^\pi(1,\text{setup};\nu) = \lambda\Delta_i^\pi(n+1,\text{busy};\nu) = \nu. \tag{18}$$

The idea of condition (18) is very similar to that of (15): From equation (13), we see that if policy $\pi = \pi_i^{B_n}$ is optimal, then the requirement $\lambda\Delta_i^\pi(n+1,\text{busy};\nu) = \nu$ results in such a combination of parameter values, where both decisions ($a = 0$ and $a = 1$) are equally qood in state $(n+1,\text{busy})$. It follows that, with these parameter values, policies $\pi_i^{B_n}$ and $\pi_i^{B_{n+1}}$ are both optimal. Similarly, from equation (13), we see that if policy π is optimal, then the requirement $\lambda\Delta_i^\pi(1,\text{setup};\nu) = \nu$ results in such a combination of parameter values, where both decisions ($a = 0$ and $a = 1$) are equally qood in state $(1,\text{off})$. It follows that, with these parameter values, policies $\pi_i^{B_n}$ and $\pi_i^{C_n}$ are both optimal, where we have defined $C_n = B_n \cup \{(1,\text{setup})\}$. Thus, if policy $\pi = \pi_i^{B_n}$ is optimal and condition (18) is satisfied with some parameter combination, then all three policies $\pi_i^{B_n}$, $\pi_i^{B_{n+1}}$, and $\pi_i^{C_n}$ are optimal for such a parameter combination. From condition (18) we are able to solve (any) two of the parameters as a function of the other parameters. For δ_i, we get the following solution:

$$\delta_i = \frac{\mu_i}{\sum_{j=0}^{n-1}(n-j)\rho_i^j}.$$

Let us denote this solution by $b_{i,n}$ so that

$$b_{i,n} = \frac{\mu_i}{\sum_{j=0}^{n-1}(n-j)\rho_i^j}, \quad n \in \{1,2,\ldots\}. \tag{19}$$

It is again easy to see that the sequence $(b_{i,n})$ is strictly decreasing and converging to 0:

$$b_{i,1} > b_{i,2} > b_{i,3} > \ldots > 0. \tag{20}$$

In addition, by comparing (16) and (19), we see that, for any $n \in \{0, 1, \ldots\}$,

$$a_{i,n} = b_{i,n+1} \left(1 + \tfrac{\beta}{h_i} \hat{P}_i(\text{setup})\right) > b_{i,n+1}. \tag{21}$$

Now we are ready to formulate our main result related to the sufficient conditions for indexability. This result will be proved piecewise in the forthcoming sections (see Theorem 2 in Section 5 and Theorems 3 and 4 in Section 6).

Theorem 1. *If there is $L \in \{0, 1, \ldots\}$ such that, for all $k \in \{0, 1, \ldots, L\}$,*

$$b_{i,k} > a_{i,k}, \tag{22}$$

where $a_{i,k}$ and $b_{i,k}$ are defined in (16) and (19), respectively, and $b_{i,0} = \infty$, then the relaxed optimization problem with objective function (7) for an InstantOff server i is indexable for the system parameter values that additionally satisfy condition

$$\delta_i > a_{i,L} = \frac{\mu_i \left(1 + \tfrac{\beta}{h_i} \hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{L} (L + 1 - j) \rho_i^j}. \tag{23}$$

In this case, the corresponding Whittle index for any state $x \in \mathcal{X}$ is given by

$$\nu_i^*(x) = h_i H_{i,\ell,s}(x) + \beta B_{i,\ell,s}(x), \tag{24}$$

where ℓ is defined by

$$\ell = \min\{k \in \{0, 1, \ldots, L\} : \delta_i > a_{i,k}\}, \tag{25}$$

s is defined by

$$s = \begin{cases} 0, & \text{if } \delta_i > b_{i,\ell}, \\ 1, & \text{otherwise,} \end{cases} \tag{26}$$

and factors $H_{i,\ell,s}(x)$ and $B_{i,\ell,s}(x)$ are defined as follows:
For $x = (0, \text{off})$,

$$H_{i,\ell,s}(0, \text{off}) = \tfrac{1}{\sum_{j=0}^{\ell} \rho_i^j} \left(\rho_i \textstyle\sum_{j=0}^{\ell} (j+1) \rho_i^j + \sigma_i\right),$$
$$B_{i,\ell,s}(0, \text{off}) = \hat{P}_i(\text{busy}) \rho_i + \hat{P}_i(\text{setup}) \tfrac{\sigma_i}{\sum_{j=0}^{\ell} \rho_i^j}.$$

For any $x = (n, \text{busy})$, *where* $1 \le n \le \ell$,

$$H_{i,\ell,s}(n, \text{busy}) = (n+1)\rho_i,$$
$$B_{i,\ell,s}(n, \text{busy}) = \hat{P}_i(\text{busy})\rho_i.$$

For any $x = (n, \text{busy})$, *where* $n \ge \ell + 1$,

$$H_{i,\ell,0}(n, \text{busy}) = H_{i,\ell-1,1}(n, \text{busy}),$$
$$H_{i,\ell,1}(n, \text{busy}) =$$
$$\rho_i\Big(\tfrac{1}{1+\sigma_i}\sum_{k=0}^{n-\ell-1}\big(\tfrac{\sigma_i}{1+\sigma_i}\big)^k \sum_{j=0}^{n+1-k}(n+1-k-j)\rho_i^j + (\ell+1)\big(\tfrac{\sigma_i}{1+\sigma_i}\big)^{n-\ell}\Big),$$
$$B_{i,\ell,s}(n, \text{busy}) = \rho_i\left(\hat{P}_i(\text{busy}) - \hat{P}_i(\text{setup})\tfrac{\sigma_i}{1+\sigma_i}\right).$$

For any $x = (n, \text{setup})$, *where* $n \ge 1$,

$$H_{i,\ell,0}(n, \text{setup}) = H_{i,\ell-1,1}(n, \text{setup}),$$
$$H_{i,\ell,1}(n, \text{setup}) = H_{i,\ell,1}(n+\ell, \text{busy}) +$$
$$\tfrac{\sigma_i - \rho_i\sum_{j=0}^{\ell-1}(\ell-j)\rho_i^j}{\sum_{j=0}^{\ell}\rho_i^j}\Big(\tfrac{1}{1+\sigma_i}\sum_{k=0}^{n-1}\big(\tfrac{\sigma_i}{1+\sigma_i}\big)^k \sum_{j=0}^{n+\ell+1-k}\rho_i^j + \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^n\Big),$$
$$B_{i,\ell,s}(n, \text{setup}) = \rho_i\left(\hat{P}_i(\text{busy}) - \hat{P}_i(\text{setup})\tfrac{\sigma_i}{1+\sigma_i}\right).$$

Remark 1. *In [1], we proved the indexability property for the special case* $L = 0$ *of Theorem 1. Note that condition (22) is trivially true in this case, and (23) is equivalent with condition (14).*

Remark 2. *It is easy to give examples of the system parameter values that satisfy condition (22) even for any* $k \in \{0, 1, \ldots\}$. *This is the case, for example, when*

$$\rho_i = \frac{\lambda}{\mu_i} \ge 2, \quad \frac{\beta}{h_i}\hat{P}_i(\text{setup}) \le 1,$$

as we see from the following proposition. In general, we can say that the results of Theorem 1 can be applied whenever the arrival rate λ *is high enough. Note that, in such a case, the mean setup delay* $E[D_i] = 1/\delta_i$ *can take any positive value and still the problem is indexable.*

Proposition 1. $b_{i,n} > a_{i,n}$ *for all* $n \in \{0, 1, \ldots\}$ *if and only if*

$$\rho_i \ge 1 + \frac{\beta}{h_i}\hat{P}_i(\text{setup}). \tag{27}$$

Proof. For the proof, we define function

$$f_n(z) = \frac{\sum_{j=0}^{n}(n+1-j)z^j}{\sum_{j=0}^{n-1}(n-j)z^j}, \quad z \geq 0.$$

Note that

$$f_n(z) = \begin{cases} \frac{n+2}{n}, & \text{if } z = 1; \\ \frac{n+1-(n+2)z+z^{n+2}}{n-(n+1)z+z^{n+1}}, & \text{otherwise.} \end{cases}$$

It is also easy to see that, for any $z \geq 0$,

$$f_n(z) > \max\{1, z\} \quad \text{and} \quad \lim_{n\to\infty} f_n(z) = \max\{1, z\}. \tag{28}$$

1° Assume first that $b_{i,n} > a_{i,n}$ for all $n \in \{0, 1, \ldots\}$. By (16) and (19), inequality $b_{i,n} > a_{i,n}$ is equivalent with inequality

$$f_n(\rho_i) > 1 + \tfrac{\beta}{h_i}\hat{P}_i(\text{setup}). \tag{29}$$

Now if $\rho_i < 1 + \frac{\beta}{h_i}\hat{P}_i(\text{setup})$, it follows from (28) that there is n such that

$$f_n(\rho_i) \leq 1 + \tfrac{\beta}{h_i}\hat{P}_i(\text{setup}),$$

which contradicts (29). Thus, it must be so that $\rho_i \geq 1 + \frac{\beta}{h_i}\hat{P}_i(\text{setup})$.

2° Assume now that $\rho_i \geq 1 + \frac{\beta}{h_i}\hat{P}_i(\text{setup})$. By (28), we have, for any n,

$$f_n(\rho_i) > \max\{1, \rho_i\} = \rho_i \geq 1 + \tfrac{\beta}{h_i}\hat{P}_i(\text{setup}),$$

which completes the proof due to (29). □

Example 1. *As an illustrative example of Whittle index values resulting from (24), consider the following parameter values:*

$$\lambda = 2, \quad h_i = 1, \quad \mu_i = 1, \quad \beta = 1, \quad \hat{P}_i(\text{setup}) = 1,$$

which implies that condition (27) is satisfied so that we can apply Theorem 1. The resulting Whittle index values $\nu_i^*(x)$ for states

$$x \in \{(0, \text{off}), (1, \text{busy}), (2, \text{busy}), (3, \text{busy}), (4, \text{busy}), (1, \text{setup}), (2, \text{setup})\}$$

as a function of the mean setup delay $E[D_i] = 1/\delta_i$ are shown in Figure 2.

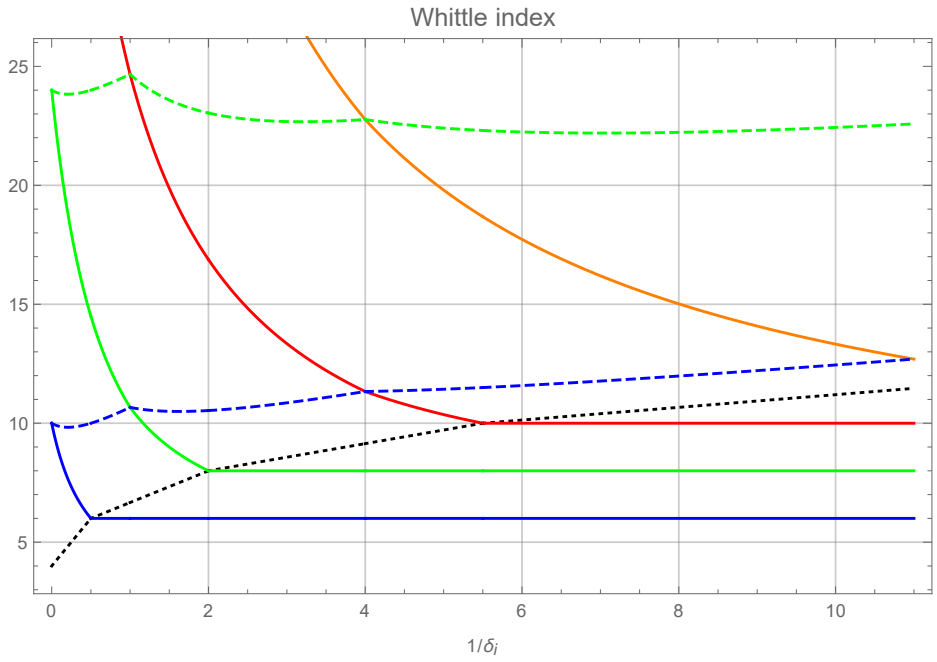

Figure 2: Whittle index values $\nu_i^*(x)$ in Example 1 for different states x ($(0, \text{off})$: dotted black; $(1, \text{busy})$: blue; $(2, \text{busy})$: green; $(3, \text{busy})$: red; $(4, \text{busy})$: orange; $(1, \text{setup})$: dashed blue; $(2, \text{setup})$: dashed green) as a function of the mean setup delay $E[D_i] = 1/\delta_i$. Condition (14) for a short mean setup delay is satisfied in the interval $E[D_i] \in (0, 1/a_{i,0})$, where $1/a_{i,0} = 0.5$. The other thresholds for the mean setup delay $E[D_i]$ are as follows: $1/b_{i,1} = 1.0$, $1/a_{i,1} = 2.0$, $1/b_{i,2} = 4.0$, $1/a_{i,2} = 5.5$, $1/b_{i,3} = 11.0$. Note that, in between these thresholds, the order of Whittle index values for different states remains unchanged, as we will prove later on.

5. Whittle index for short setup delays

In this section, we prove (for completeness) the indexability property for the special case $L = 0$ of Theorem 1. In addition, we derive an explicit expression for the corresponding Whittle index. These results are given in Theorem 2 below. As already mentioned above, condition (22) is trivially true in this case, and (23) is equivalent with condition (14). So we are considering the case of short setup delays.

Let us first introduce the following *total order* among all states $x \in \mathcal{X}$:

$$(0, \text{off}) \prec (1, \text{busy}) \prec (1, \text{setup}) \prec (2, \text{busy}) \prec (2, \text{setup}) \prec \dots \tag{30}$$

In addition, let $T(x)$, where $x \in \mathcal{X}$, denote the following set of states:

$$T(x) = \{y \in \mathcal{X} \mid y \preceq x\}.$$

Thus, we have

$$\begin{aligned}
&T(0, \text{off}) = \{(0, \text{off})\},\\
&T(1, \text{busy}) = \{(0, \text{off}), (1, \text{busy})\},\\
&T(1, \text{setup}) = \{(0, \text{off}), (1, \text{busy}), (1, \text{setup})\},\\
&T(2, \text{busy}) = \{(0, \text{off}), (1, \text{busy}), (1, \text{setup}), (2, \text{busy})\}, \quad \ldots
\end{aligned}$$

The corresponding policies $\pi_i^{T(x)}$ with activity sets $T(x)$ are called *threshold policies* with respect to the total order (30). In addition, we include the rudimentary threshold policy $\pi_i^{\emptyset}$, where all states are passive, in the family of threshold policies, denoted by Π_i^{T}. Note that each of these policies generate a Markov process with a single and finite positive recurrent class. Its steady-state distribution can be determined based on the global balance equations and the normalization condition, and its relative value function by solving the so-called Howard equations.

Theorem 2. *Under assumption (14), the optimization problem with objective function (7) for an InstantOff server i is indexable, and the corresponding Whittle index for any state $x \in \mathcal{X}$ is given by*

$$\nu_i^*(x) = h_i H_i(x) + \beta B_i(x), \tag{31}$$

where factors $H_i(x)$ and $B_i(x)$ are defined as follows:
For $x = (0, \text{off})$,

$$\begin{aligned}
H_i(0, \text{off}) &= \rho_i + \sigma_i,\\
B_i(0, \text{off}) &= \hat{P}_i(\text{busy})\rho_i + \hat{P}_i(\text{setup})\sigma_i.
\end{aligned}$$

For any $x = (n, \text{busy})$, where $n \geq 1$,

$$\begin{aligned}
&H_i(n, \text{busy}) \;=\\
&\begin{cases}
\frac{\rho_i}{\rho_i(1+\sigma_i)-\sigma_i}\Big(\frac{\rho_i\left((n+1)-(n+2)\rho_i+\rho_i^{n+2}\right)}{(1-\rho_i)^2} + \\
\quad \sigma_i^2 - (n+1)\sigma_i - \sigma_i(\sigma_i-\rho_i)(\frac{\sigma_i}{1+\sigma_i})^n\Big), & \lambda \neq \mu_i,\\
\frac{1}{2}(n+1)(n+2) + \\
\quad \sigma_i^2 - (n+1)\sigma_i - \sigma_i(\sigma_i-1)(\frac{\sigma_i}{1+\sigma_i})^n, & \lambda = \mu_i,
\end{cases}\\
&B_i(n, \text{busy}) = \rho_i\left(\hat{P}_i(\text{busy}) - \hat{P}_i(\text{setup})\tfrac{\sigma_i}{1+\sigma_i}\right).
\end{aligned}$$

For any $x = (n, \text{setup})$, where $n \geq 1$,

$$H_i(n, \text{setup}) = \begin{cases} \frac{\rho_i\left((n+1)-(n+2)\rho_i+\rho_i^{n+2}\right)}{(1-\rho_i)^2} + \sigma_i, & \lambda \neq \mu_i, \\ \frac{1}{2}(n+1)(n+2) + \sigma_i, & \lambda = \mu_i, \end{cases}$$

$$B_i(n, \text{setup}) = \rho_i \left(\hat{P}_i(\text{busy}) - \hat{P}_i(\text{setup}) \tfrac{\sigma_i}{1+\sigma_i} \right).$$

Proof. Because of page restrictions, the proof can be found in Appendix B of the supplementary material of this paper. □

Remark 3. *As can be seen from the proof, the holding cost factors $H_i(n, \text{busy})$ and $H_i(n, \text{setup})$ in (31) can also be given as follows:*

$$H_i(n, \text{busy}) = \rho_i \Big(\tfrac{1}{1+\sigma_i} \textstyle\sum_{k=0}^{n-1} \left(\tfrac{\sigma_i}{1+\sigma_i}\right)^k \sum_{j=0}^{n+1-k} (n+1-k-j)\rho_i^j + \left(\tfrac{\sigma_i}{1+\sigma_i}\right)^n \Big),$$

$$H_i(n, \text{setup}) = H_i(n, \text{busy}) + \sigma_i \Big(\tfrac{1}{1+\sigma_i} \textstyle\sum_{k=0}^{n-1} \left(\tfrac{\sigma_i}{1+\sigma_i}\right)^k \sum_{j=0}^{n+1-k} \rho_i^j + \left(\tfrac{\sigma_i}{1+\sigma_i}\right)^n \Big).$$

Note also that, in [1, Theorem 1], there is an unfortunate misprint in the formula of $H_i(n, \text{setup})$ for the case $\lambda \neq \mu_i$, which can be identified when comparing it with the correct formula given above in (31).

Remark 4. *The Whittle index for an ordinary NeverOff server with idle and busy states can be determined from (31) by taking the limit $\delta \to \infty$ (i.e., $\sigma \to 0$) and interpreting the state $(0, \text{off})$ as the idle state of the NeverOff server. Note that by taking the limit $\delta \to \infty$, assumption (14) is satisfied for any combination of the other parameters. In the limit, we get the following formulas:*

For the idle state with $n = 0$,

$$H_i(0) = \rho_i,$$
$$B_i(0) = \rho_i(P_i(\text{busy}) - P_i(\text{idle})).$$

For any busy state with $n \geq 1$,

$$H_i(n) = \begin{cases} \frac{\rho_i\left((n+1)-(n+2)\rho_i+\rho_i^{n+2}\right)}{(1-\rho_i)^2}, & \lambda \neq \mu_i, \\ \frac{1}{2}(n+1)(n+2), & \lambda = \mu_i, \end{cases}$$

$$B_i(n) = \rho_i(P_i(\text{busy}) - P_i(\text{idle})).$$

We also note that without any energy costs (i.e., assuming $\beta = 0$) the resulting Whittle index $\nu_i^*(n) = h_i H_i(n)$, $n \geq 0$, is equal to the corresponding Whittle index given in [24, Eqn. (7.3)] for a finite-state M/M/1/n queue when multiplied by the arrival rate λ. The slight difference in the index is due to the difference in the problem setting itself: In our formulation the Lagrangian multiplier ν is interpreted as the price of passivity per time unit, while in [24] it is the price of rejection per arriving job.

Remark 5. *Note that condition (14) does not include parameter λ. Thus, under (14), indexability is guaranteed for any load of the system.*

6. Whittle index for general setup delays

In this section, we prove the indexability property for the general case $L \geq 1$ of Theorem 1. In addition, we derive an explicit expression for the corresponding Whittle index.

Note first that, for $\ell = 0$, the proof is exactly same as the proof of Theorem 2. Let then $\ell \in \{1, 2, \ldots, L\}$ be such that $a_{i,\ell-1} \geq \delta_i > a_{i,\ell}$, i.e.,

$$\frac{\mu_i \left(1 + \frac{\beta}{h_i} \hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{\ell-1} (\ell - j) \rho_i^j} \geq \delta_i > \frac{\mu_i \left(1 + \frac{\beta}{h_i} \hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{\ell} (\ell + 1 - j) \rho_i^j}. \tag{32}$$

We split this case into two separate parts depending on whether $s = 0$ or $s = 1$, and study them in their own subsections below.

6.1. Case $s = 0$

Let us first assume that $b_{i,\ell-1} > a_{i,\ell-1} > \delta_i > b_{i,\ell} > a_{i,\ell}$, i.e.,

$$\begin{aligned} &\frac{\mu_i}{\sum_{j=0}^{\ell-2} (\ell - 1 - j) \rho_i^j} > \frac{\mu_i \left(1 + \frac{\beta}{h_i} \hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{\ell-1} (\ell - j) \rho_i^j} > \delta_i \; > \\ &\qquad \frac{\mu_i}{\sum_{j=0}^{\ell-1} (\ell - j) \rho_i^j} > \frac{\mu_i \left(1 + \frac{\beta}{h_i} \hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{\ell} (\ell + 1 - j) \rho_i^j}, \end{aligned} \tag{33}$$

which corresponds to the case $s = 0$ of Theorem 1. For this case, the indexability property and the corresponding Whittle index are formulated below in Theorem 3. The remaining special case $\delta_i = a_{i,\ell-1}$ is thereafter commented on in Remark 6.

We start by introducing a new *total order* among all states $x \in \mathcal{X}$, which replaces the earlier one, given in (30), from this on:

$$\begin{aligned}&(1,\text{busy}) \prec \ldots \prec (\ell,\text{busy}) \prec (0,\text{off}) \prec \\ &\quad (1,\text{setup}) \prec (\ell+1,\text{busy}) \prec (2,\text{setup}) \prec (\ell+2,\text{busy}) \prec \ldots\end{aligned} \tag{34}$$

Naturally, this new order also affects the definition of sets $T(x) = \{y \in \mathcal{X} \mid y \preceq x\}$ so that now we have

$$\begin{aligned}&T(1,\text{busy}) = \{(1,\text{busy})\},\\ &\ldots\\ &T(\ell,\text{busy}) = \{(1,\text{busy}),\ldots,(\ell,\text{busy})\},\\ &T(0,\text{off}) = \{(1,\text{busy}),\ldots,(\ell,\text{busy}),(0,\text{off})\},\\ &T(1,\text{setup}) = \{(1,\text{busy}),\ldots,(\ell,\text{busy}),(0,\text{off}),(1,\text{setup})\},\\ &T(\ell+1,\text{busy}) = \{(1,\text{busy}),\ldots,(\ell,\text{busy}),(0,\text{off}),(1,\text{setup}),(\ell+1,\text{busy})\},\\ &\ldots\end{aligned}$$

The corresponding policies $\pi_i^{T(x)}$ with activity sets $T(x)$ (together with rudimentary policy $\pi_i^{\emptyset}$) are called *threshold policies* with respect to the total order (34), and collectively denoted by Π_i^{T}.

Theorem 3. *Under assumption (33), the optimization problem with objective function (7) for an InstantOff server i is indexable, and the corresponding Whittle index for any state $x \in \mathcal{X}$ is given by*

$$\nu_i^*(x) = h_i H_i(x) + \beta B_i(x), \tag{35}$$

where factors $H_i(x)$ and $B_i(x)$ are defined as follows:
For $x = (n,\text{busy})$, where $1 \le n \le \ell$,

$$\begin{aligned}H_i(n,\text{busy}) &= (n+1)\rho_i,\\ B_i(n,\text{busy}) &= \hat{P}_i(\text{busy})\rho_i.\end{aligned}$$

For $x = (0,\text{off})$,

$$\begin{aligned}H_i(0,\text{off}) &= \tfrac{1}{\sum_{j=0}^{\ell}\rho_i^j}\Big(\rho_i \textstyle\sum_{j=0}^{\ell}(j+1)\rho_i^j + \sigma_i\Big),\\ B_i(0,\text{off}) &= \hat{P}_i(\text{busy})\rho_i + \hat{P}_i(\text{setup})\tfrac{\sigma_i}{\sum_{j=0}^{\ell}\rho_i^j}.\end{aligned}$$

For any $x = (n, \text{setup})$, where $n \geq 1$,

$$\begin{aligned}
&H_i(n, \text{setup}) \;= \\
&\quad \rho_i \Big(\tfrac{1}{1+\sigma_i} \textstyle\sum_{k=0}^{n-1} \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^k \sum_{j=0}^{n+\ell-k} (n+\ell-k-j)\rho_i^j + \ell \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^n \Big) + \\
&\quad \tfrac{\sigma_i - \rho_i \sum_{j=0}^{\ell-2} (\ell-1-j)\rho_i^j}{\sum_{j=0}^{\ell-1} \rho_i^j} \Big(\tfrac{1}{1+\sigma_i} \textstyle\sum_{k=0}^{n-1} \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^k \sum_{j=0}^{n+\ell-k} \rho_i^j + \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^n \Big), \\
&B_i(n, \text{setup}) = \rho_i \left(\hat{P}_i(\text{busy}) - \hat{P}_i(\text{setup}) \tfrac{\sigma_i}{1+\sigma_i} \right).
\end{aligned}$$

For any $x = (n, \text{busy})$, where $n \geq \ell + 1$,

$$\begin{aligned}
&H_i(n, \text{busy}) \;= \\
&\quad \rho_i \Big(\tfrac{1}{1+\sigma_i} \textstyle\sum_{k=0}^{n-\ell} \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^k \sum_{j=0}^{n+1-k} (n+1-k-j)\rho_i^j + \ell \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^{n-\ell+1} \Big), \\
&B_i(n, \text{busy}) = \rho_i \left(\hat{P}_i(\text{busy}) - \hat{P}_i(\text{setup}) \tfrac{\sigma_i}{1+\sigma_i} \right).
\end{aligned}$$

Proof. Because of page restrictions, the proof can be found in Appendix C of the supplementary material of this paper. □

Remark 6. *Consider now the special case $\delta_i = a_{i,\ell-1}$.*

Assume first that $\ell = 1$ so that

$$a_{i,0} = \delta_i > b_{i,1} > a_{i,1}. \tag{36}$$

Now, by (35), it follows that

$$\nu_i^*(1, \text{busy}) = \nu_i^*(0, \text{off}).$$

On the other hand, we see from part 1° of the previous proof that the policies $\pi_i^{\emptyset}$, $\pi_i^{T(1,\text{busy})}$, and $\pi_i^{T(0,\text{off})}$ are equally good and optimal at this point $\nu = \nu_i^*(1, \text{busy}) = \nu_i^*(0, \text{off})$. So, for a proper handling of this special case, we need to modify the total order (34) in such a way that the states $(1, \text{busy})$ and $(0, \text{off})$ have the same "rank":

$$\{(1, \text{busy}), (0, \text{off})\} \prec (1, \text{setup}) \prec (2, \text{busy}) \prec (2, \text{setup}) \prec (3, \text{busy}) \prec \ldots \tag{37}$$

This also affects the threshold policies $T(1, \text{busy})$ and $T(0, \text{off})$, which need to redefined as follows:

$$T(1, \text{busy}) = T(0, \text{off}) = \{(1, \text{busy}), (0, \text{off})\}.$$

Assume now that $\ell \geq 2$ so that

$$b_{i,\ell-1} > a_{i,\ell-1} = \delta_i > b_{i,\ell} > a_{i,\ell}. \tag{38}$$

Again by (35), it follows that

$$\nu_i^*(\ell, \text{busy}) = \nu_i^*(0, \text{off}).$$

On the other hand, we see from part 2.2° of the previous proof that the policies $\pi_i^{T(\ell-1,\text{busy})}$, $\pi_i^{T(\ell,\text{busy})}$, and $\pi_i^{T(0,\text{off})}$ are equally good and optimal at this point $\nu = \nu_i^*(\ell, \text{busy}) = \nu_i^*(0, \text{off})$. So, for a proper handling of this special case, we need to modify the total order (34) in such a way that the states (ℓ, busy) and $(0, \text{off})$ have the same "rank":

$$\begin{aligned} &(1, \text{busy}) \prec \ldots \prec (\ell-1, \text{busy}) \prec \{(\ell, \text{busy}), (0, \text{off})\} \prec \\ &\quad (1, \text{setup}) \prec (\ell+1, \text{busy}) \prec (2, \text{setup}) \prec (\ell+2, \text{busy}) \prec \ldots \end{aligned} \tag{39}$$

This also affects the threshold policies $T(\ell, \text{busy})$ and $T(0, \text{off})$, which need to redefined as follows:

$$T(\ell, \text{busy}) = T(0, \text{off}) = \{(1, \text{busy}), \ldots, (\ell-1, \text{busy}), (\ell, \text{busy}), (0, \text{off})\}.$$

With these modifications, the optimization problem with objective function (7) is indexable even under assumption (36) or (38), and the corresponding Whittle index for any state $x \in \mathcal{X}$ is still given by (35), which can be proved similarly as Theorem 3.

6.2. Case $s = 1$

Let us now assume that $b_{i,\ell-1} > a_{i,\ell-1} > b_{i,\ell} > \delta_i > a_{i,\ell}$, i.e.,

$$\begin{aligned} &\frac{\mu_i}{\sum_{j=0}^{\ell-2}(\ell-1-j)\rho_i^j} > \frac{\mu_i\left(1+\frac{\beta}{h_i}\hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{\ell-1}(\ell-j)\rho_i^j} > \\ &\qquad \frac{\mu_i}{\sum_{j=0}^{\ell-1}(\ell-j)\rho_i^j} > \delta_i > \frac{\mu_i\left(1+\frac{\beta}{h_i}\hat{P}_i(\text{setup})\right)}{\sum_{j=0}^{\ell}(\ell+1-j)\rho_i^j}, \end{aligned} \tag{40}$$

which corresponds to the case $s = 1$ of Theorem 1. For this case, the indexability property and the corresponding Whittle index are formulated below

in Theorem 4. The remaining special case $\delta_i = b_{i,\ell}$ is thereafter commented on in Remark 7.

We start again by introducing a new *total order* among all states $x \in \mathcal{X}$, which replaces the earlier ones, given in (30) and (34), from this on:

$$\begin{aligned}(1,\text{busy}) \prec \ldots \prec (\ell,\text{busy}) \prec (0,\text{off}) \prec \\ (\ell+1,\text{busy}) \prec (1,\text{setup}) \prec (\ell+2,\text{busy}) \prec (2,\text{setup}) \prec \ldots\end{aligned} \tag{41}$$

Note that the difference between (41) and (34) is related to the order of states (n,setup) and $(n+\ell,\text{busy})$ for all $n \geq 1$. Naturally, this new order again affects the definition of sets $T(x) = \{y \in \mathcal{X} \mid y \preceq x\}$ so that now we have

$$\begin{aligned}
&T(1,\text{busy}) = \{(1,\text{busy})\},\\
&\ldots\\
&T(\ell,\text{busy}) = \{(1,\text{busy}),\ldots,(\ell,\text{busy})\},\\
&T(0,\text{off}) = \{(1,\text{busy}),\ldots,(\ell,\text{busy}),(0,\text{off})\},\\
&T(\ell+1,\text{busy}) = \{(1,\text{busy}),\ldots,(\ell,\text{busy}),(0,\text{off}),(\ell+1,\text{busy})\},\\
&T(1,\text{setup}) = \{(1,\text{busy}),\ldots,(\ell,\text{busy}),(0,\text{off}),(\ell+1,\text{busy}),(1,\text{setup})\},\\
&\ldots
\end{aligned}$$

The corresponding policies $\pi_i^{T(x)}$ with activity sets $T(x)$ (together with rudimentary policy $\pi_i^{\emptyset}$) are called *threshold policies* with respect to the total order (41), and collectively denoted by Π_i^{T}.

Theorem 4. *Under assumption (40), the optimization problem with objective function (7) for an InstantOff server i is indexable, and the corresponding Whittle index for any state $x \in \mathcal{X}$ is given by*

$$\nu_i^*(x) = h_i H_i(x) + \beta B_i(x), \tag{42}$$

where factors $H_i(x)$ and $B_i(x)$ are defined as follows:
For $x = (n,\text{busy})$, where $1 \leq n \leq \ell$,

$$\begin{aligned}H_i(n,\text{busy}) &= (n+1)\rho_i,\\ B_i(n,\text{busy}) &= \hat{P}_i(\text{busy})\rho_i.\end{aligned}$$

For $x = (0,\text{off})$,

$$\begin{aligned}H_i(0,\text{off}) &= \tfrac{1}{\sum_{j=0}^{\ell}\rho_i^j}\Big(\rho_i \textstyle\sum_{j=0}^{\ell}(j+1)\rho_i^j + \sigma_i\Big),\\ B_i(0,\text{off}) &= \hat{P}_i(\text{busy})\rho_i + \hat{P}_i(\text{setup})\tfrac{\sigma_i}{\sum_{j=0}^{\ell}\rho_i^j}.\end{aligned}$$

For any $x = (n, \text{busy})$, where $n \geq \ell + 1$,

$$\begin{aligned}
&H_i(n, \text{busy}) \;= \\
&\quad \rho_i \Big(\tfrac{1}{1+\sigma_i} \textstyle\sum_{k=0}^{n-\ell-1} \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^k \sum_{j=0}^{n+1-k} (n+1-k-j)\rho_i^j + (\ell+1)\big(\tfrac{\sigma_i}{1+\sigma_i}\big)^{n-\ell} \Big), \\
&B_i(n, \text{busy}) = \rho_i \left(\hat{P}_i(\text{busy}) - \hat{P}_i(\text{setup}) \tfrac{\sigma_i}{1+\sigma_i} \right).
\end{aligned}$$

For any $x = (n, \text{setup})$, where $n \geq 1$,

$$\begin{aligned}
&H_i(n, \text{setup}) \;= \\
&\quad \rho_i \Big(\tfrac{1}{1+\sigma_i} \textstyle\sum_{k=0}^{n-1} \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^k \sum_{j=0}^{n+\ell+1-k} (n+\ell+1-k-j)\rho_i^j + (\ell+1)\big(\tfrac{\sigma_i}{1+\sigma_i}\big)^{n} \Big) + \\
&\quad \tfrac{\sigma_i - \rho_i \sum_{j=0}^{\ell-1} (\ell-j)\rho_i^j}{\sum_{j=0}^{\ell} \rho_i^j} \Big(\tfrac{1}{1+\sigma_i} \textstyle\sum_{k=0}^{n-1} \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^k \sum_{j=0}^{n+\ell+1-k} \rho_i^j + \big(\tfrac{\sigma_i}{1+\sigma_i}\big)^{n} \Big), \\
&B_i(n, \text{setup}) = \rho_i \left(\hat{P}_i(\text{busy}) - \hat{P}_i(\text{setup}) \tfrac{\sigma_i}{1+\sigma_i} \right).
\end{aligned}$$

Proof. Because of page restrictions, the proof can be found in Appendix D of the supplementary material of this paper. □

Remark 7. *Consider now the special case $\delta_i = b_{i,\ell}$. More precisely said, assume that*

$$b_{i,\ell-1} > a_{i,\ell-1} > b_{i,\ell} = \delta_i > a_{i,\ell}. \tag{43}$$

By (42), it follows that, for any $n \geq 1$,

$$\nu_i^*(n + \ell, \text{busy}) = \nu_i^*(n, \text{setup}).$$

On the other hand, we see from part 4.2° of the previous proof that the policies $\pi_i^{T(0,\text{off})}$, $\pi_i^{T(\ell+1,\text{busy})}$, and $\pi_i^{T(1,\text{setup})}$ are equally good and optimal at point $\nu = \nu_i^*(\ell + 1, \text{busy}) = \nu_i^*(1, \text{setup})$, and from 6.2° that the policies $\pi_i^{T(n,\text{setup})}$, $\pi_i^{T(n+\ell+1,\text{busy})}$, and $\pi_i^{T(n+1,\text{setup})}$ are equally good and optimal at point $\nu = \nu_i^*(n + \ell + 1, \text{busy}) = \nu_i^*(n + 1, \text{setup})$ for any $n \geq 1$. So, for a proper handling of this special case, we need to modify the total order (41) in such a way that the states $(n + \ell, \text{busy})$ and (n, setup) have the same "rank" for any $n \geq 1$:

$$\begin{aligned}
&(1, \text{busy}) \prec \ldots \prec (\ell, \text{busy}) \prec (0, \text{off}) \prec \\
&\quad \{(\ell + 1, \text{busy}), (1, \text{setup})\} \prec \{(\ell + 2, \text{busy}), (2, \text{setup})\} \prec \ldots
\end{aligned} \tag{44}$$

This also affects the threshold policies $T(n+\ell, \text{busy})$ *and* $T(n, \text{setup})$*, which need to redefined, for any* $n \geq 1$*, as follows:*

$$\begin{aligned} &T(n+\ell, \text{busy}) = T(n, \text{setup}) \ = \\ &\quad \{(1, \text{busy}), \ldots, (\ell, \text{busy}), (0, \text{off}), (\ell+1, \text{busy}), (1, \text{setup}), \ldots, \\ &\qquad (n+\ell, \text{busy}), (n, \text{setup})\}. \end{aligned}$$

With these modifications, the optimization problem with objective function (7) is indexable even under assumption (43), and the corresponding Whittle index for any state $x \in \mathcal{X}$ *is still given by (42), which can be proved similarly as Theorem 4.*

7. Energy-aware Whittle index policy

Now we return to the original dispatching problem described in Section 2, where we have K servers and a strict dispatching condition (3). Based on the results given in Section 4, we introduce the following energy-aware index policy.

Definition 2. *For any InstantOff server* i *with state* $x_i = (n_i, z_i)$*, we define index*

$$\nu_i^{\text{EW}}(x_i) = \nu_i^*(x_i), \tag{45}$$

where $\nu_i^*(x_i)$ *is defined by (24) in Theorem 1. The dispatching rule that at every time* t *chooses the server with the lowest index* $\nu_i^{\text{EW}}(x_i)$ *is called the* Energy-aware Whittle index policy *(EW) for the original dispatching problem. All possible ties are broken randomly.*

In the following section, we will evaluate, by numerical simulations, the performance of the proposed energy-aware Whittle index policy (EW), and compare it to the dynamic policies Join-Shortest-Queue (JSQ) and First-Policy-Iteration (FPI) mentioned in Section 1. Both of these policies are also index-based. For JSQ, the index of an InstantOff server i with state $x_i = (n_i, z_i)$ is clearly given by

$$\nu_i^{\text{JSQ}}(x_i) = n_i. \tag{46}$$

All possible ties are broken randomly.

For FPI, the corresponding index was derived in [12]. This approach requires to fix a basic dispatching policy, which is then improved by the policy iteration method. The first iteration is mathematically tractable if the basic policy is chosen to be a static policy, in which the decisions are state-independent. A natural static policy in this context is the Load Balancing (LB) policy that makes probabilistic dispatching decisions, where the dispatching probabilities p_i are proportional to the service rates μ_i. From [12], we get the following index for an InstantOff server i with state $x_i = (n_i, z_i)$, for which the dedicated arrival rate for server i, according to the static basic policy, is given by $\lambda_i = \lambda p_i$:

$$\nu_i^{\mathrm{FPI}}(x_i) = h_i H_i^{\mathrm{FPI}}(x_i) + \beta B_i^{\mathrm{FPI}}(x_i), \tag{47}$$

where we have used the following notations:
For $x_i = (0, \mathrm{off})$,

$$\begin{aligned} H_i^{\mathrm{FPI}}(0, \mathrm{off}) &= \tfrac{1}{\mu_i - \lambda_i} + \tfrac{1}{\delta_i}, \\ B_i^{\mathrm{FPI}}(0, \mathrm{off}) &= \tfrac{1}{\mu_i} \left(\hat{P}_i(\mathrm{busy}) + \hat{P}_i(\mathrm{setup}) \tfrac{\mu_i - \lambda_i}{\lambda_i + \delta_i} \right). \end{aligned}$$

For any $x_i = (n_i, \mathrm{busy})$, where $n \geq 1$,

$$\begin{aligned} H_i^{\mathrm{FPI}}(n_i, \mathrm{busy}) &= \tfrac{n_i + 1}{\mu_i - \lambda_i} - \tfrac{1}{\delta_i} \tfrac{\lambda_i}{\mu_i - \lambda_i}, \\ B_i^{\mathrm{FPI}}(n_i, \mathrm{busy}) &= \tfrac{1}{\mu_i} \left(\hat{P}_i(\mathrm{busy}) - \hat{P}_i(\mathrm{setup}) \tfrac{\lambda_i}{\lambda_i + \delta_i} \right). \end{aligned}$$

For any $x_i = (n_i, \mathrm{setup})$, where $n \geq 1$,

$$\begin{aligned} H_i^{\mathrm{FPI}}(n_i, \mathrm{setup}) &= \tfrac{n_i + 1}{\mu_i - \lambda_i} + \tfrac{1}{\delta_i}, \\ B_i^{\mathrm{FPI}}(n_i, \mathrm{setup}) &= \tfrac{1}{\mu_i} \left(\hat{P}_i(\mathrm{busy}) - \hat{P}_i(\mathrm{setup}) \tfrac{\lambda_i}{\lambda_i + \delta_i} \right). \end{aligned}$$

All possible ties are again broken randomly.

8. Numerical results

In this section we illustrate the performance of the Whittle-index based EW policy and compare it against several reference policies to gain insight. In addition to the dynamic policies JSQ and FPI already mentioned in the

previous section, we use the static LB policy (also mentioned in the previous section) as a reference policy. Moreover, we apply the policy iteration algorithm to numerically solve the optimal policy, whenever possible.

In our examples, the results for the policies JSQ, FPI and EW have been produced by using discrete-event simulations. For each combination of the parameters, the results are based on simulation runs with $4 \cdot 10^6$ arrivals. Thus, there are enough statistics to estimate the performance measures so accurately that the 95% confidence intervals are practically indistinguishable and they have been omitted from the figures. The LB policy can be easily evaluated numerically, since under that policy each queue behaves independently from each other as an M/G/1 queue with setup delays, see, e.g., [12]. Finally, the results are shown as a function of the total load of the system, given by $\rho = \lambda / \sum_i \mu_i$. To vary the load ρ, we fix the μ_i and vary λ. Also, in our examples for the power settings, we assume $P_i(\text{busy}) = P_i(\text{setup})$, for all i, and the holding costs $h_i = 1$, for all i.

Note that in the Whittle index policy, for a given value of λ with all other parameters fixed, the indexability is not necessarily guaranteed and it must be verified. To do this, one first determines the parameter ℓ through (25) and then verifies that $b_{i,k} > a_{i,k}$ for all $k \leq \ell$ to have an indexable problem. The parameter s is thereafter determined from (26). If the problem is indexable, the parameters ℓ and s define the precise form of the index values, as also illustrated earlier in Example 1. Basically, as the load increases with λ in our numerical examples, there is always a lower bound for λ such that the problem is indexable, unless $\ell = 0$ in which case indexability is guaranteed independent of λ, see (14).

8.1. Small system with 2 servers

In our first example, we consider a small system with only 2 servers. The parameters are the following: $\{\mu_1, \mu_2\} = \{1, 2\}$ 1/s, $\{\delta_1, \delta_2\} = \{4, 17\}$ 1/s, $\{P_1(\text{busy}), P_2(\text{busy})\} = \{0.2, 0.3\}$ kW, $P_1(\text{off}) = P_2(\text{off}) = 0$ kW and $\beta = 1$. The parameters satisfy the condition (14) for both servers, and thus the indexability is guaranteed for any value of the load. For this small system, the state space of the four-dimensional process remains moderate and we are able to apply the policy iteration algorithm to numerically solve the optimal policy minimizing the mean total costs (5) in a truncated state space. Here the truncation has been done at 35 jobs in each queue, which is sufficiently high to allow a reasonably accurate estimation of the optimal policy even at relatively high values of the load ρ. For each load, the policy iteration

has been performed for 10 iterations, which yielded the optimal costs with at least 5 digit accuracy.

The results are shown in Figure 3, which depicts the ratio of the mean number of jobs (top left panel), the mean power (top right panel) and the mean total cost (bottom panel) to the corresponding quantities of the optimal policy as a function of the load ρ for different policies (LB, JSQ, FPI and EW).

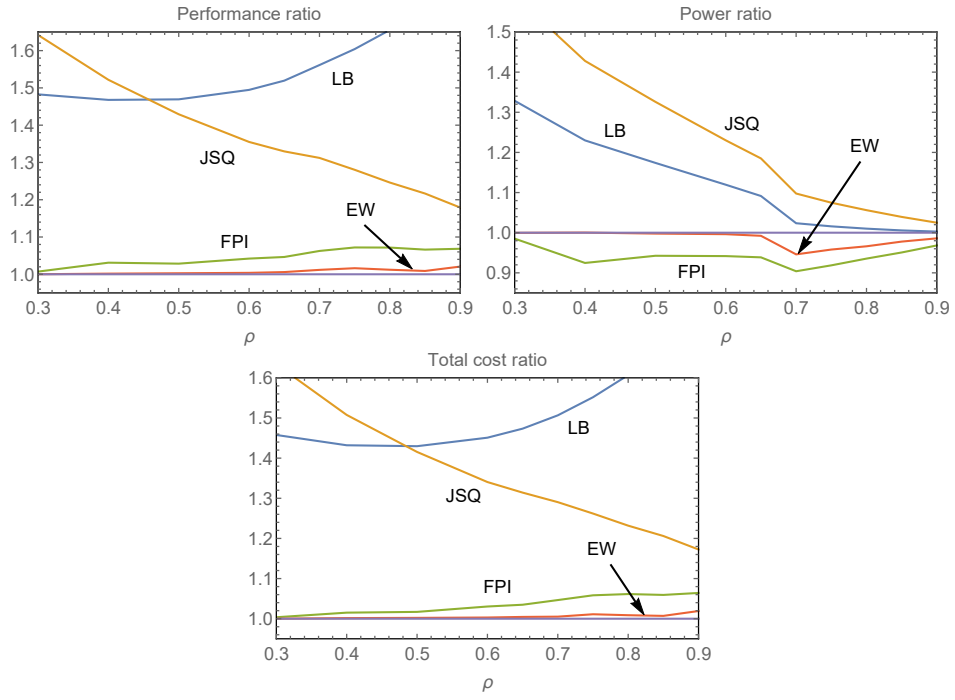

Figure 3: Ratio of mean number of jobs (top left panel), mean power (top right panel) and mean total cost (bottom panel) to the corresponding quantities of the optimal policy as a function of the load ρ for different policies (LB, JSQ, FPI and EW) with 2 servers.

Consider first the performance ratio (top left panel). The static LB policy clearly does not perform very well, and gets worse as load increases relative to the optimal policy. On the other hand, the JSQ policy gets better and better the higher the load, which can be explained by the fact that at very high load all queues are active all the time and the energy-aware features are not affecting the behavior that much. The servers are heterogeneous, but JSQ also indirectly takes care of this as the queue in the faster server is typically shorter than in the slower one. The FPI and EW policies are both

clearly near-optimal, i.e., the ratio is close to 1. However, the FPI policy is less optimal than the EW policy. Then considering the results for the power ratio (top right panel), it can be observed that the non-energy-aware JSQ policy is the worst, being even worse than the static LB policy. Both LB and JSQ are performing poorly at low loads but become better as load increases, since at higher loads both servers are on all the time anyway. However, the FPI policy is here even better with respect power consumption than the optimal policy, and it is also better than EW, which remains very close to optimal until $\rho = 0.6$ but then becomes marginally better than the optimal policy. Finally, by looking at the total costs (bottom panel) we see that they are close to the ones for the performance part as the weight $\beta = 1$ is quite small relative to the total mean power. In summary, our proposed EW policy is performing systematically better than FPI and it is overall very close to the optimal policy; in fact it is indistinguishable from the optimal until load $\rho = 0.6$, while the performance of the FPI policy starts deviating from the optimal already at low load, reaching a deviation of approximately 10% at higher load.

8.2. Larger system with 10 servers

In the previous example, the setup delays in particular were unrealistically short for real servers. Thus, next we study a somewhat larger system consisting of 10 servers with realistic values for the power and setup delay parameters. All servers are homogeneous in this scenario with the following power parameters: $P_i(\text{busy}) = 0.2$ kW and $P_i(\text{off}) = 0.01$ kW, for $i = 1, \ldots, 10$. The mean setup delay is 10 s, i.e., $\delta_i = 0.1$ 1/s, for all $i = 1, \ldots, 10$. These power values and the setup times are similar to those used in more experimental studies, see, e.g., [8, 20]. Finally, the service rate $\mu_i = 1.0$ 1/s, and $\beta = 1$. In this case, the indexability is guaranteed approximately for $\rho > 0.05$. Due to the size of the state space, the optimal policy cannot be anymore numerically evaluated. Thus, in the following figures the results are no longer normalized, but they represent the absolute values of the corresponding quantities.

The results are shown in Figure 4, which shows the mean number of jobs (top left panel), the mean power in kW units (top right panel) and the mean total cost (bottom panel) as a function of the load ρ for different policies (LB, JSQ, FPI and EW). Qualitatively the behavior of the policies relative to each other is similar to what we observed earlier with 2 heterogeneous servers, except for the properties of the FPI policy. In terms of the mean

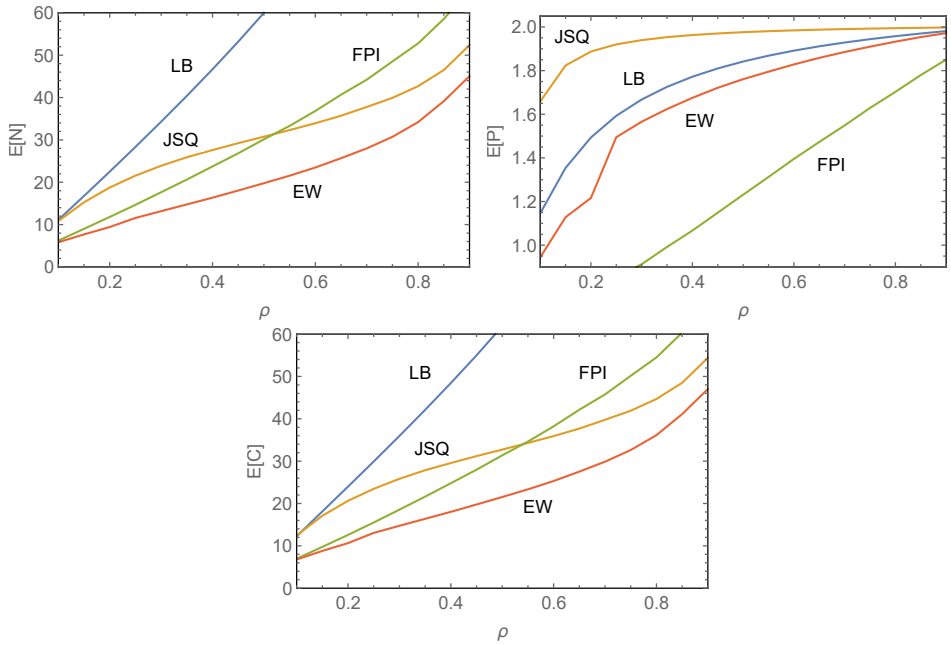

Figure 4: The mean number of jobs (top left panel), mean power (top right panel) and mean total cost (bottom panel) with 10 homogeneous servers as a function of the load ρ for different policies (LB, JSQ, FPI and EW).

number of jobs (top left panel), we again see that the static LB policy behaves significantly worse than all dynamic policies. However, in this larger system example a clear difference between JSQ and FPI emerges: at higher loads JSQ is performing better than the FPI policy, while at lower loads FPI is better than JSQ. Notably, our EW policy is still performing significantly better than either FPI or JSQ. On the other hand, with respect to the power consumption (top right panel), FPI performs best, i.e., it is packing jobs on a fewer number of servers allowing them to sleep and save energy. Our EW policy is still somewhat better than the static LB policy, while JSQ is even worse than LB. With respect to the total cost (bottom panel), our EW policy is overall giving the lowest mean cost, being smaller than that achieved by FPI or JSQ. All dynamic policies are much better than LB.

Next we modify the system by having heterogeneous service rates and busy powers such that $\mu_i = 1.0$ 1/s and $P_i(\text{busy}) = 0.2$ kW, for all $i = 1, \ldots, 5$, and $\mu_i = 2.0$ 1/s and $P_i(\text{busy}) = 0.3$ kW for all $i = 6, \ldots, 10$. Now indexability holds if $\rho > 0.13$. The results are shown in Figure 5, which

shows the mean number of jobs (top left panel), the mean power in kW units (top right panel) and the mean total cost (bottom panel) as a function of the load ρ for different policies (LB, JSQ, FPI and EW). Otherwise we can observe qualitatively very similar to previous case, but our EW policy is now somewhat better than FPI or JSQ in optimizing the performance when servers have heterogeneous service rates. Correspondingly, the gap between EW and FPI or JSQ in the total cost (bottom panel) is also slightly larger.

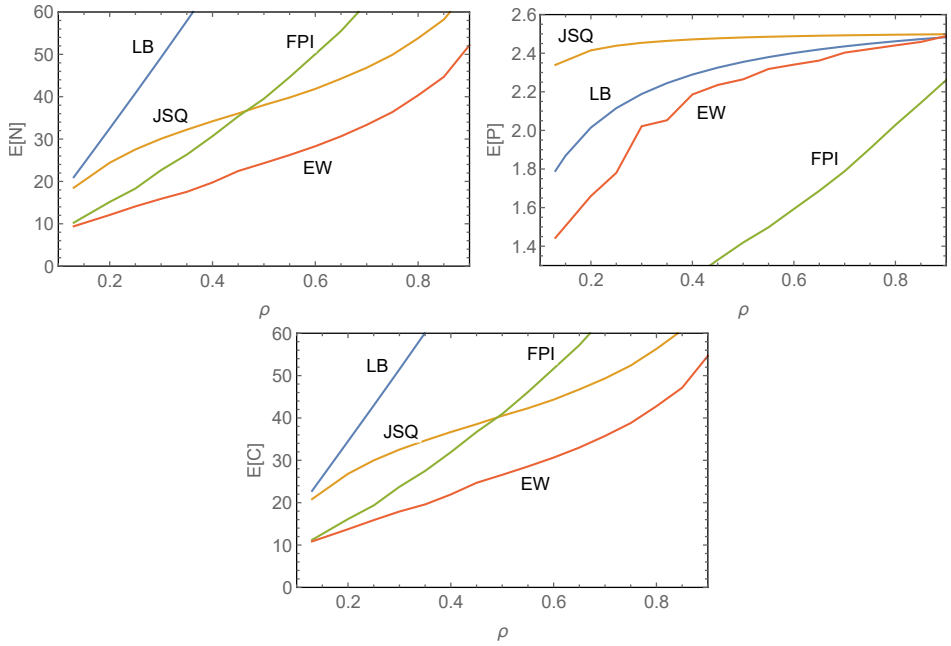

Figure 5: The mean number of jobs (top left panel), mean power (top right panel) and mean total cost (bottom panel) with 10 heterogeneous servers as a function of the load ρ for different policies (LB, JSQ, FPI and EW).

Observe also the sawtooth-like behavior, in particular, in the energy graphs for the EW policy, see Figure 4 (top right panel) and Figure 5 (top right panel), but also to a lesser degree in the corresponding graphs on the mean number of jobs. Also, by inspecting closely the mean number of jobs graph for the FPI policy in Figure 5 (top left panel), the same phenomenon can be observed at lower loads. We believe, that this is a consequence of the dependence of the policies on λ, which is varying. The policies LB and JSQ do not depend explicitly on λ and do not have this behavior.

8.3. System with 10 NeverOff or InstantOff servers

Finally, we consider a scenario where the system with 10 servers consists of a mixture of NeverOff and InstantOff servers. A NeverOff server in our model corresponds to a server with $\delta \to \infty$ and the power consumption in the off state is the idle state power of the server. We assume that servers have all an identical service rate $\mu_i = 1.0$ 1/s and busy power consumption $P_i(\text{busy}) = 0.2$ kW, for all $i = 1, \ldots, 10$. The InstantOff servers have all identical setup delays of 10 s, i.e., $\delta_i = 0.1$ 1/s and the off state power is $P_i(\text{off}) = 0.01$ kW. Similarly, the NeverOff servers have off state power (corresponding to idle state power), which is 60% of the busy state power, i.e., $P_i(\text{off}) = 0.12$ kW. The NeverOff servers satisfy the indexability condition independent of ρ and the InstantOff servers satisfy indexability when $\rho > 0.05$, i.e., they correspond to the earlier homogeneous scenario with 10 servers.

In this scenario, the number of NeverOff servers also needs to be determined. To this end, for a given value of λ, we choose the number of NeverOff servers to be such that with them alone the system is slightly unstable and the rest of the servers are then InstantOff servers. More precisely, the number of NeverOff servers is set to $\lfloor \lambda/\mu_i \rfloor$.

The results as a function of the load with the different policies (LB, JSQ, FPI, EW) for the performance (top left panel), power consumption (top right panel) and total cost (bottom panel) are depicted in Figure 6. As earlier, LB performs worst (top left panel). Also, FPI is performing better than JSQ at lower loads, as earlier, but at higher loads although JSQ is better, the difference to FPI is not that large. For the energy part (top right panel), interestingly our EW policy is now slightly worse than LB at higher loads. Overall in terms of the total cost (bottom panel), our EW policy yields the lowest costs among all policies.

Finally, in order to have a more accurate idea of the behavior of each policy across the different load values, in Figure 6 on the x-axis the load values have been discretized with a spacing of 0.01, while in the earlier figures the spacing was 0.05. The sawtooth-like behavior in the graphs is now evident in all policies, which is here due to the changing configuration as the load varies, i.e., the number of NeverOff servers increases by 1 each time load increases by 0.1. In particular, for the LB, JSQ and EW policies the mean number of jobs drops down (see upper left panel) each time a new NeverOff server is added because this always reduces delays compared with a system, where the corresponding server is an InstantOff server. However, for the FPI policy, in addition to the effects of the changing configuration, the policy

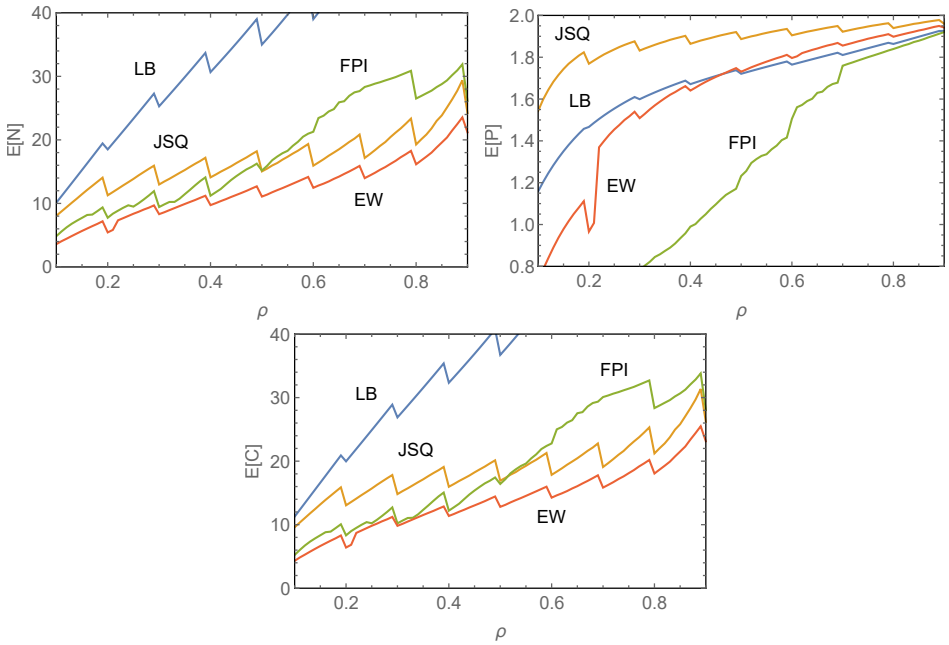

Figure 6: The mean number of jobs (top left panel), mean power (top right panel) and mean total cost (bottom panel) with 10 NeverOff or InstantOff servers as a function of the load ρ for different policies (LB, JSQ, FPI and EW).

itself changes gradually with the load causing additional non-smoothness in the performance.

9. Conclusions

We have considered the energy-aware dispatching problem in a system consisting of parallel M/M/1 queues with InstantOff servers. Such servers go to sleep after becoming empty to save energy, but activation of the server after sleep incurs an additional setup delay penalty. The costs in our system model consist of linear holding costs and power consumption costs. The performance-energy trade-off is characterized as a weighted sum of these.

To optimize the trade-off we have applied the Whittle index approach, which is based on a certain relaxation of the original intractable dispatching problem, and results in a separable problem, where each queue is considered in isolation. In the earlier version of the paper [1], we proved the indexability property for the systems with sufficiently short setup delays, which, however, may not be a realistic assumption. In the present paper, we have

found sufficient conditions for indexability that allow longer (and, thus, more realistic) setup delays. In addition, we have also derived the explicit form of the Whittle index under these conditions. Our conditions essentially mean that for fixed values of all other system parameters except the total arrival rate of jobs, there is always a lower bound on the total arrival rate such that indexability is guaranteed above that. In other words, indexability is guaranteed if the total load is high enough.

The proof is technically challenging, as each queue is described by a two-dimensional Markov process, representing the M/M/1 queue with setup. The original proof of indexability relied on establishing a certain ordering of the states in the two-dimensional state space when the mean setup delay is sufficiently short. Here we show that a set of successively looser conditions on the mean length of the setup delay can be determined under which the problem still remains indexable. The challenge in the proof is that the ordering of the states changes along the system parameters.

As demonstrated by our numerical experiments, the resulting energy-aware Whittle index policy is able to perform very close to the numerically solved optimal policy in a small system. In larger systems, where numerically solving the optimal policy becomes intractable, the Whittle index policy still outperforms all considered reference policies.

While we have managed to prove the indexability property and to derive the Whittle index values explicitly whenever the system load is high enough, it is still open whether this is possible for all parameter combinations with lower load. Future research includes attacking this problem.

It would also be worth studying a more general system where the server is woken up only when there is a sufficient number (say, $k \geq 1$) of jobs waiting, since, for a system with a single energy-aware server, it is known to perform better than just having $k = 1$, see, e.g., [7, 23, 10]. Another direction of future research is to consider whether the resulting Whittle index policy is asymptotically optimal, which is the case under certain technical assumptions [28, 26].

References

[1] S. Aalto and P. Lassila. Whittle index approach to energy-aware dispatching. In *Proc. of ITC 30*, pages 19–27, September 2018. Available from: https://itc-conference.org/en/itc-library/itc30.html.

[2] S. Aalto, P. Lassila, and P. Osti. Opportunistic scheduling with flow size information for Markovian time-varying channels. *Performance Evaluation*, 112:27–52, 2017.

[3] P.S. Ansell, K.D. Glazebrook, and C. Kirkbride. Generalised 'join the shortest queue' policies for the dynamic routing of jobs to multiclass queues. *Journal of the Operational Research Society*, 54:379–389, 2003.

[4] N.T. Argon, L. Ding, K.D. Glazebrook, and S. Ziya. Dynamic routing of customers with general delay costs in a multiserver queuing system. *Probability in the Engineering and Informational Sciences*, 23:175–203, 2009.

[5] U. Ayesta, M. Erasquin, and P. Jacko. A modeling framework for optimizing the flow-level scheduling with time-varying channels. *Performance Evaluation*, 67:1014–1029, 2010.

[6] F. Cecchi and P. Jacko. Nearly-optimal scheduling of users with Markovian time-varying transmission rates. *Performance Evaluation*, 99-100:16–36, 2016.

[7] A. Gandhi, V. Gupta, M. Harchol-Balter, and M.A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation*, 67:1155–1171, 2010.

[8] A. Gandhi, M. Harchol-Balter, and M.A. Kozuch. Are sleep states effective in data centers? In *Proc. of IGCC*, June 2012.

[9] M. Gebrehiwot, S. Aalto, and P. Lassila. Energy-performance trade-off for processor sharing queues with setup delay. *Operations Research Letters*, 44:101–106, 2016.

[10] M. Gebrehiwot, S. Aalto, and P. Lassila. Optimal energy-aware control policies for FIFO servers. *Performance Evaluation*, 103:41–59, 2016.

[11] M. Gebrehiwot, S. Aalto, and P. Lassila. Energy-aware SRPT server with batch arrivals: Analysis and optimization. *Performance Evaluation*, 115:92–107, 2017.

[12] M. Gebrehiwot, S. Aalto, and P. Lassila. Near-optimal policies for energy-aware task assignment in server farms. In *Proc. of CCGrid*, pages 1017–1026, May 2017.

[13] M. Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[14] E. Hyytiä, D. Down, P. Lassila, and S. Aalto. Dynamic control of running servers. In *Proc. of MMB 2018*, pages 127–141, February 2018.

[15] E. Hyytiä, A. Penttinen, and S. Aalto. Size- and state-aware dispatching problem with queue-specific job sizes. *European Journal of Operational Research*, 217:357–370, 2012.

[16] E. Hyytiä and R. Righter. Fairness through linearly increasing holding costs in systems of parallel servers with setup delays. In *Proc. of ITC 27*, pages 143–151, September 2015.

[17] E. Hyytiä, R. Righter, and S. Aalto. Energy-aware job assignment in server farms with setup delays under LCFS and PS. In *Proc. of ITC 26*, September 2014.

[18] E. Hyytiä, R. Righter, and S. Aalto. Task assignment in a heterogeneous server farm with switching delays and general energy-aware cost structure. *Performance Evaluation*, 75-76:17–35, 2014.

[19] E. Hyytiä, R. Righter, J. Virtamo, and L. Viitasaari. Value (generating) functions for the $M^X/G/1$ queue. In *Proc. of ITC 29*, pages 232–240, September 2017.

[20] C. Isci, S. McIntosh, K. Jeffrey, R. Das, J. Hanson, S. Piper, R. Wolford, T. Brey, R. Kantner, A. Ng, J. Norris, A. Traore, and M. Frissora. Agile, efficient virtualization power management with low-latency server power states. *ACM SIGARCH Computer Architecture News*, 41(3):96–107, 2013.

[21] P. Jacko. Value of information in optimal flow-level scheduling of users with Markovian time-varying channels. *Performance Evaluation*, 68:1022–1036, 2011.

[22] M. Larrañaga, U. Ayesta, and I.M. Verloop. Dynamic control of birth-and-death restless bandits: Application to resource-allocation problems. *IEEE/ACM Transactions on Networking*, 24:3812–3825, 2016.

[23] V.J. Maccio and D.G. Down. On optimal policies for energy-aware servers. *Performance Evaluation*, 90:36–52, 2015.

[24] J. Niño-Mora. Dynamic allocation indices for restless projects and queueing admission control: A polyhedral approach. *Mathematical Programming*, 93:361–413, 2002.

[25] M.L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 2005.

[26] I.M. Verloop. Asymptotically optimal priority policies for indexable and nonindexable restless bandits. *The Annals of Applied Probability*, 26:1947–1995, 2016.

[27] R. Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, 15:406–413, 1978.

[28] R. Weber and G. Weiss. On an index policy for restless bandits. *Journal of Applied Probability*, 27:637–648, 1990.

[29] P. Whittle. Restless bandits: activity allocation in a changing world. *Journal of Applied Probability*, 25A:287–298, 1988.

[30] P. Whittle. *Optimal Control: Basics and Beyond*. Wiley, 1996.

[31] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14:181–189, 1977.