Energy-Efficient Heuristics for Insensitive Job Assignment in Processor-Sharing Server Farms

Jing Fu, *Graduate Student Member, IEEE,* Jun Guo, *Member, IEEE,* Eric W. M. Wong, *Senior Member, IEEE,* and Moshe Zukerman, *Fellow, IEEE*

***Abstract*—Energy efficiency of server farms is an important design consideration of the green datacenter initiative. One effective approach is to optimize power consumption of server farms by controlling the carried load on the networked servers. In this paper, we propose a robust heuristic policy called E* for stochastic job assignment in a server farm, aiming to improve the energy efficiency by maximizing the ratio of job throughput to power consumption. Our model of the server farm considers a parallel system of finite-buffer processor-sharing queues with heterogeneous server speeds and energy consumption rates. We devise E* as an insensitive policy so that the stationary distribution of the number of jobs in the system depends on the job size distribution only through its mean. We provide a rigorous analysis of E* and compare it with a baseline approach, known as most energy-efficient server first (MEESF), that greedily chooses the most energy-efficient servers for job assignment. We show that E* has always a higher job throughput than that of MEESF, and derive realistic conditions under which E* is guaranteed to outperform MEESF in energy efficiency. Extensive numerical results are presented and demonstrate that E* can improve the energy efficiency by up to 100%.**

***Index Terms*—Energy efficiency, insensitivity, job assignment, processor sharing, server farm.**

I. Introduction

DATA centers have become essential to the functioning of virtually every sector of a modern economy [1]. Server farms in data centers are known for their massive power consumption [2]. Energy efficiency of server farms is important considering greenhouse gas emissions concerns. Driven by the green datacenter initiative to facilitate a low-carbon economy in the information age [3], there is a strong incentive for managing power consumption of server farms while maintaining acceptable levels of performance [4].

Various approaches have been proposed in the literature aiming at energy conservation in server farms. These include speed scaling methods for optimizing power consumption by controlling server speeds based on their carried load [5]–[10], right sizing techniques for dynamically activating/deactivating servers [11], [12], and resource allocation solutions for performance optimization in server farms [13], [14]. The approach that we take in this paper is in line with [15] and investigates the problem of stochastic job assignment in a server farm. The objective is to maximize its *energy efficiency*, defined as the ratio of job throughput to power consumption. This approach provides a way for optimizing power consumption of server farms by controlling the carried load on the networked servers as a function of the (fixed) server speeds and energy consumption rates. It can also be combined with speed scaling at each server for local fine-tuning.

Our model of the server farm considers a parallel system of finite-buffer processor-sharing (PS) queues with heterogeneous server speeds and energy consumption rates. PS queues are suitable for modeling web-server systems [16]–[19]. Under PS, all existing jobs at each server share the processing capacity and are served at equal rates. PS enables fair processing of jobs, which is desirable in web-server systems where the file size distribution is known to have high variability [20]. Note that broader applications of PS queues can also be found in communication systems (see e.g. [21], [22]).

In this paper, we focus on job assignment policies that allow *jockeying*. When jockeying is permitted, jobs can be reassigned to any server with buffer vacancies at any time before they are completed. Assignment with jockeying has more freedom and suits a server farm where the servers are collocated in a single physical center and can use e.g. a shared DRAM-storage [23] or flash-storage [24]. It may also suit a data center with more advanced virtualization technologies that enable high-speed live migration of jobs [25]. Jockeying policies in general can significantly improve the system performance. In addition, they are scalable in computation and hence make resource optimization more tractable.

For such a problem, a straightforward heuristic approach is by greedily choosing the most energy-efficient servers for job assignment. It can be shown that, under certain conditions, this approach, which we call most energy-efficient server first (MEESF) in this paper, maximizes the ratio of instantaneous job departure rate to instantaneous energy consumption rate. In general, however, this is not the case, and we shall see that MEESF does not necessarily maximize the ratio of long-run average job departure rate (i.e., job throughput) to long-run average energy consumption rate (i.e., power consumption). This observation motivates us to design a more robust heuristic policy for improving the energy efficiency of the system and yet achieving a higher job throughput than what can be achieved with MEESF.

A preliminary conference version of this paper was presented in [26]. Here, we extend [26] by providing a more thorough understanding of the heuristic policy and a more rigorous analysis of its properties, taking into consideration servers with heterogeneous buffer sizes. Our main contribu-

This work was supported by City University of Hong Kong under Project No. 9380044. A preliminary version of this paper was presented at IEEE INFOCOM'15, Hong Kong, April 2015.

The authors are with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong (e-mail: jingfu6-c@my.cityu.edu.hk; j.guo@cityu.edu.hk; eeewong@cityu.edu.hk; m.zu@cityu.edu.hk).

tions in this paper are summarized as follows:

- We propose a robust heuristic policy for stochastic job assignment in a finite-buffer PS server farm. Note that in this paper we shall name this heuristic policy as E* to reflect our goal of designing a "star" policy that can maximize the energy efficiency of the system. We demonstrate the effectiveness of E* by comparing it to the baseline MEESF policy. Unlike MEESF that greedily chooses the most energy-efficient servers for job assignment, E* aggregates an optimal number of $\widehat{K} \geq 2$ most energy-efficient servers to form a *virtual* server. In our design, E* always gives preference to this virtual server and utilizes its service capacity in such a way that guarantees a higher job throughput than what is achievable with MEESF and yet can improve the energy efficiency of the system. The decision variable $\widehat{K}$ provides a degree of freedom for E* to fine-tune its performance.
- We discuss the insight gained from our design of E*, based on which we propose a simple rule of thumb for determining the optimal $\widehat{K}$ value that maximizes the energy efficiency of the system under E*. The resulting policy, referred to as *rate matching* (RM), simply chooses the value of $\widehat{K}$ such that the aggregate service rate of the virtual server matches the job arrival rate. We provide extensive numerical results to demonstrate the effectiveness of RM relative to E*.
- We devise E* as an insensitive job assignment policy. That is, the stationary (steady-state) distribution of the number of jobs in the system depends on the job size distribution only through its mean. This insensitivity property is useful for assuring robustness and predictability of the performance of the server farm under a wide range of job size distributions.
- We perform a rigorous analysis of E*. In particular, we prove that E* has always a higher job throughput than that of MEESF. We also prove that, under the realistic scenario where at least two servers in a heterogeneous server farm are equally most energy efficient, E* is guaranteed to outperform MEESF in terms of the energy efficiency of the system. Having at least two servers that are equally energy efficient can be justified as a server farm is likely to comprise multiple servers of the same type purchased at a time.

The rest of this paper is organized as follows. In Section II, we discuss the related work. In Section III, we describe the server farm model. In Section IV, we provide details of the job assignment policies. Insensitive conditions of the job assignment policies are derived in Section V. In Section VI, we provide a rigorous analysis of the E* policy. Numerical results are presented in Section VII. Finally, we draw conclusions in Section VIII.

II. Related Work

Since the work of Haight [27] in 1958, various queueing models for job assignment with or without jockeying among multiple servers have been considered in the literature. Most of the existing work is focused on job assignment policies that aim to improve the system performance under the first-come-first-served (FCFS) service discipline. Among them, join the shortest queue (JSQ) is a classic policy.

The JSQ policy in the non-jockeying case has been studied in [28]–[30] for FCFS queues and in [31]–[33] for PS queues. Bonomi [31] proved the optimality of JSQ in a PS model with homogeneous servers under general arrival process and Markov departure process. Whitt [32] provided a counter-example for showing the non-optimality of JSQ in the case of non-exponential job size distribution. Gupta [33] presented an approximate analysis of the performance of JSQ in a PS model with general job size distribution, and intuitively explained its optimality in terms of the average delay in a system with heterogeneous servers.

Server farm applications of the JSQ policy in the jockeying case have been studied in [34]–[36] for FCFS queues. The jockeying action is triggered when the difference between the queue length of the shortest queue and that of the longest queue reaches a threshold value. The focus of this existing work is on the expression of the equilibrium distribution of the queue length, and it has been shown that jockeying policies can significantly improve the system performance.

Several energy-aware non-jockeying policies have been studied in [14], [37] for a multi-queue heterogeneous system with infinite buffer and setup delay, where exact expressions of the value function for the Markov decision process (MDP) are given. Hyytiä *et al.* [37] showed that M/G/1-LCFS is only sensitive to the mean of the set-up delay, while M/G/1-PS loses its insensitivity with consideration of the set-up delay. Li *et al.* [38] proposed a heuristic approach which aims to maximize the weighted probability of the combined execution time (delay) and the energy consumption metric with constraints on both the delay deadline and the energy budget in a heterogeneous computing system. To the best of our knowledge, PS multi-queue systems with jockeying have been studied before only in [15], where the optimization problem is characterized by the semi-Markov decision process (SMDP) [39] and attempts to maximize the ratio of the job throughput to the power consumption.

In the general context of MDP or SMDP, significant work has been done to optimize the expected average reward (cost). In [40], Lippman optimized the finite horizon discounted reward in three models, namely, M/M/c finite capacity queue, M/M/1 service rate control problem and M/M/c finite capacity queue with policy-dependent arrival rates, by using the concept of "uniformization" device which defines a virtual exponential clock independent from both the policy and the state of the stochastic process. The concept of *g-revised* reward [41] provides a bridge between optimization problems of the expected total reward and the expected average reward. In addition, an optimal solution that maximizes the expected average reward can be achieved by using a procedure proposed in [41]. In particular, Stidham and Weber [42] considered a service rate control problem of a single queue with left-skip-free transition structure in both exponential and non-exponential service time cases, where the decisions of the policy depend on the number of customers (jobs) in the queue and no discounting is considered. They provided a method to prove the monotonicity of the optimal service rates, in which the optimal service rates are increasing in queue length, and some of their resulted optimal policies are shown to be insensitive to the shape of the service time distribution. George and Harrison [43] studied the service rate control problem of a single queue evolving a birth-and-death Markov process, and proved the existence of monotonic optimal service rates with weaker assumption than those of [42].

The objective function, defined as the long-run average reward per unit cost (e.g. time consumption, energy consumption, etc.), in [15] is a generalized version of the long-run average service quality per unit time that has been studied previously. Rosberg *et al.* [15], using similar techniques of [40]–[43], proposed an algorithm for optimal stationary job assignment that maximizes the energy efficiency of the system by exploring server heterogeneity. However, the optimal policy is not scalable in computation. Rosberg *et al.* [15] further proposed the scalable slowest server first (SSF) policy which aims to approximate this optimality. SSF was numerically demonstrated to be near-optimal under certain relationship between the energy consumption rate and the service rate of a server. Our approach in this paper gives rise to a class of scalable job assignment policies that are provably more robust than SSF and can significantly improve the energy efficiency of the system under a variety of cases.

III. System Model

For the reader's convenience, Table I provides a list of major symbols that we shall define and use in this paper. Our model of a server farm considers K independent servers, each having a finite buffer for queuing jobs. For $j = 1, 2, \ldots, K$, we denote by B_j the buffer size of server j where $B_j > 1$ for all j. For notational convenience, we denote by $\widetilde{B}_i$ the aggregate buffer size of the first i servers in the system, given by

$$\widetilde{B}_i = \sum_{j=1}^{i} B_j,\ i = 0, 1, \ldots, K \qquad (1)$$

where $\widetilde{B}_0 = 0$ by definition.

We denote by μ_j the service rate of server j, defined as the units of jobs that can be processed per time unit, and by ε_j the energy consumption rate of server j. Note that, in the literature, the energy consumption rate of a server is usually related to the server speed by a convex function of the form

$$\varepsilon(\mu) \propto \mu^{\beta},\ \beta > 0 \qquad (2)$$

with $\beta = 3$ being the most commonly used value [5], [6], [44]. However, some researchers suggest that $\varepsilon(\mu)$ is not necessarily convex [7], [8]. The job assignment policies that we propose in this paper do not require such an assumption.

We refer to the ratio μ_j/ε_j as the energy efficiency of server j. Accordingly, server i is defined to be more energy-efficient than server j if and only if $\mu_i/\varepsilon_i > \mu_j/\varepsilon_j$. Since our focus in this paper is on developing energy-efficient job assignment policies, for convenience of description, we label the servers according to their energy efficiency. That is, for any pair of servers i and j, if $i < j$, we have $\mu_i/\varepsilon_i \geq \mu_j/\varepsilon_j$.

TABLE I
Summary of major symbols

Symbol	Definition
K	Number of servers in the system
B_j	Buffer size of server j
$\widetilde{B}_i$	Aggregate buffer size of the first i servers in the system
μ_j	Service rate of server j
ε_j	Energy consumption rate of server j
μ_j/ε_j	Energy efficiency of server j
λ	Job arrival rate
$\widehat{K}$	Number of energy-efficient servers forming a virtual server
$\widetilde{\mu}_{\widehat{K}}$	Aggregate service rate of the virtual server
$\widetilde{\varepsilon}_{\widehat{K}}$	Aggregate energy consumption rate of the virtual server
$\widetilde{\mu}_{\widehat{K}}/\widetilde{\varepsilon}_{\widehat{K}}$	Energy efficiency of the virtual server
$\widehat{K}^*$	Optimal value of $\widehat{K}$ chosen by the E* policy
$\widehat{K}^{\mathrm{RM}}$	Empirical value of $\widehat{K}$ chosen by the RM policy
$\mathcal{L}^{\phi}$	Job throughput of the system under policy ϕ
$\mathcal{E}^{\phi}$	Power consumption of the system under policy ϕ
$\mathcal{L}^{\phi}/\mathcal{E}^{\phi}$	Energy efficiency of the system under policy ϕ

Remark 1. *In our context, for $i < j$, we have $\mu_i\varepsilon_j - \varepsilon_i\mu_j \geq 0$ since $\mu_i/\varepsilon_i \geq \mu_j/\varepsilon_j$.*

We consider that jobs arrive at the system according to a Poisson process with mean rate λ. An arriving job is assigned to one of the servers with at least one vacant slot in its buffer, subject to the control of an assignment policy. If all buffers are full, the arriving job is lost.

We assume that job sizes (in units) are independent and identically distributed with an absolutely continuous cumulative distribution function (CDF) $F(x)$, $x \geq 0$. Without loss of generality, we normalize the average size of jobs to one. Each server j serves its jobs at a total rate of μ_j using the PS service discipline.

IV. Energy-Efficient Job Assignment

In this section, we provide details of the job assignment policies. Section IV-A briefly describes the MEESF policy. Section IV-B presents the E* policy. Section IV-C discusses and describes the RM policy.

A. MEESF

MEESF is a straightforward heuristic approach that works by greedily choosing the most energy-efficient servers for job assignment. With jockeying, at any point in time, MEESF is required to satisfy the condition that no server in the system is busy if a more energy-efficient server has at least one vacant slot in its buffer. A rearrangement of the existing jobs is always required whenever it becomes necessary to satisfy the condition of the policy.

The SSF policy proposed in [15] is a special case of MEESF, assuming for each server j that its energy consumption rate satisfies $\varepsilon_j = \mu_j^3$. It can be shown that, under the condition where all servers in the system have the same buffer size,

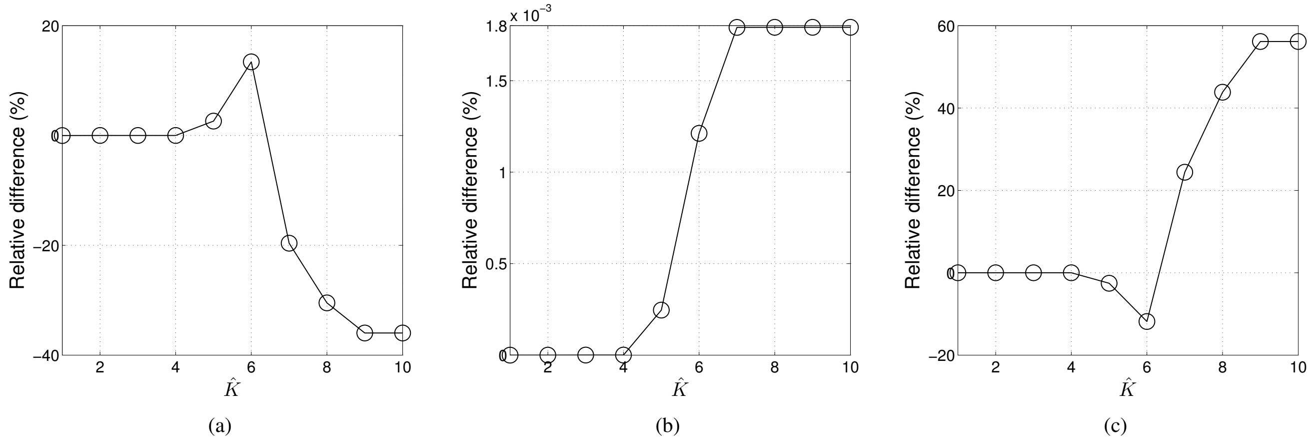


Fig. 1. Illustration of optimizing $\widehat{K}$ for the E* policy. (a) Relative difference of E* to MEESF in the energy efficiency of the system. (b) Relative difference of E* to MEESF in the job throughput. (c) Relative difference of E* to MEESF in the power consumption.

i.e., $B_1 = B_2 = \ldots = B_K$, SSF maximizes the ratio of instantaneous job departure rate to instantaneous energy consumption rate [45]. In general, however, this is not the case. We shall see that MEESF does not necessarily maximize the ratio of long-run average job departure rate to long-run average energy consumption rate. This observation motivates us to design the more robust E* policy.

B. E*

The E* policy is devised in such a way that it aggregates $\widehat{K} \geq 2$ most energy-efficient servers to form a virtual server. Let $\widetilde{\mu}_{\widehat{K}}$ denote the aggregate service rate of this virtual server, given by

$$\widetilde{\mu}_{\widehat{K}} = \sum_{j=1}^{\widehat{K}} \mu_j. \tag{3}$$

Let $\widetilde{\varepsilon}_{\widehat{K}}$ denote the aggregate energy consumption rate of the virtual server, given by

$$\widetilde{\varepsilon}_{\widehat{K}} = \sum_{j=1}^{\widehat{K}} \varepsilon_j. \tag{4}$$

Similar to the way that we specify the energy efficiency of a server, we refer to the ratio $\widetilde{\mu}_{\widehat{K}}/\widetilde{\varepsilon}_{\widehat{K}}$ as the energy efficiency of the virtual server.

Proposition 1. *The virtual server formed by aggregating $\widehat{K}$ most energy-efficient servers in the system is not worse than any server k, $k \geq \widehat{K}+1$, in terms of energy efficiency. That is,*

$$\frac{\widetilde{\mu}_{\widehat{K}}}{\widetilde{\varepsilon}_{\widehat{K}}} \geq \frac{\mu_k}{\varepsilon_k}, \quad k = \widehat{K}+1, \widehat{K}+2, \ldots, K. \tag{5}$$

Proof: We have

$$\frac{\widetilde{\mu}_{\widehat{K}}}{\widetilde{\varepsilon}_{\widehat{K}}} - \frac{\mu_k}{\varepsilon_k} = \frac{\sum_{j=1}^{\widehat{K}} (\mu_j \varepsilon_k - \varepsilon_j \mu_k)}{\sum_{j=1}^{\widehat{K}} \varepsilon_j \varepsilon_k} \geq 0 \tag{6}$$

where the final inequality follows from Remark 1. ■

Accordingly, in our design, E* always gives preference to the virtual server and utilizes its service capacity in such a way that guarantees a higher job throughput than what is achievable with MEESF and yet can improve the energy efficiency of the system. In particular, with jockeying, E* is required to satisfy the following two conditions at any point in time:

1) Among the $\widehat{K}$ servers that form the virtual server, no server is idle if a less energy-efficient server is busy, and no server has more than one job if another server is idle.
2) None of the other $K - \widehat{K}$ servers in the system is busy if a more energy-efficient server has at least one vacant slot in its buffer.

One property of the E* policy as a result of this design is that it maximizes the instantaneous job departure rate of the virtual server at any point in time under the condition that it greedily chooses the most energy-efficient servers for job assignment among the $\widehat{K}$ servers that form the virtual server.

Note that the decision variable $\widehat{K}$ provides a degree of freedom for E* to fine-tune its performance. Thus, an important objective in the context of the E* policy is to determine within the range $[2, K]$ an optimal $\widehat{K}$ value, denoted by $\widehat{K}^*$, such that the energy efficiency of the system can be maximized.

C. RM

Our proposed rule of thumb, RM, for determining the $\widehat{K}^*$ value of the E* policy simply chooses a number $\widehat{K}^{\mathrm{RM}}$ so that the aggregate service rate of the virtual server matches the job arrival rate. More specifically, $\widehat{K}^{\mathrm{RM}}$ is chosen to be the largest $\widehat{K}$ satisfying $\widetilde{\mu}_{\widehat{K}} \leq \lambda$. Intuitively, this is to consider the fact that the maximum job throughput of the system is no more than the job arrival rate. As a result, one may expect that the servers chosen by the RM policy are (roughly) sufficient to support the maximum job throughput.

We provide below an intuitive explanation of why this simple rule of thumb may work, using the numerical results presented in Fig. 1. In this example, we have a system of ten servers. The server speeds and the energy consumption rates are randomly generated. The job arrival rate is set to be the sum of the service rates of the six most energy-efficient servers. The results in Fig. 1 are presented in the form of the *relative difference* of E* to MEESF in terms of each corresponding performance measure. Note that in this paper, given two numerical quantities x and y, we define the relative difference of x to y as $(x-y)/y$.

Based on the results of Fig. 1, we argue that choosing a different value of $\widehat{K}$ other than $\widehat{K}^{\mathrm{RM}}$ is likely to decrease the energy efficiency of the system. This is because:

- On one hand, if we choose a value of $\widehat{K}$ larger than $\widehat{K}^{\mathrm{RM}}$, the excessive service capacity made available with the virtual server can increase the job throughput only marginally. However, it can substantially increase the power consumption since it now uses more of those less energy-efficient servers. As a result, the energy efficiency of the system (defined in our context as the ratio of job throughput to power consumption) is likely to decrease with an increasing value of $\widehat{K}$ within the range $[\widehat{K}^{\mathrm{RM}}, K]$.
- On the other hand, if we choose a value of $\widehat{K}$ smaller than $\widehat{K}^{\mathrm{RM}}$, the aggregate service rate of the virtual server is not sufficient to support the input traffic in a long run. As a result, incoming jobs will be queued up and we are forced to use more of those less energy-efficient servers for serving the backlog. This will again increase the power consumption. Since it can be shown that the job throughput decreases with decreasing $\widehat{K}$, the energy efficiency is also likely to decrease with a decreasing value of $\widehat{K}$ within the range $[\widehat{K}^{\mathrm{RM}}, 2]$.

V. Insensitive Conditions With Jockeying

Zvi *et al.* [15] developed a class of stationary policies for insensitive job assignment with jockeying based on the concept of *symmetric queue* introduced in [46]. As in [15], we adapt here the symmetric queue model with state-dependent service rates defined in [46] to our system that allows jockeying among multiple queues. We begin by providing in Section V-A an explanation based on [15] of this adaptation to make the paper self-contained. Then, we derive the insensitive MEESF policy in Section V-B and the insensitive E* policy in Section V-C, respectively.

A. Adaptation of symmetric queue

A symmetric queue is defined in [46] as a queue that has a symmetry between the service rate allocated to each position in the queue and the probability that a newly arrived job will join the queue in the corresponding position. In particular, when there are n jobs in the queue, the ordered jobs are contained at positions $1, 2, \ldots, n$, and the total service rate is $\mu(n)$. A proportion $\gamma(l, n)$ of $\mu(n)$ is directed to the job at position l, $l = 1, 2, \ldots, n$.

- When the job at position l completes its service and leaves the queue, the jobs at positions $l+1, l+2, \ldots, n$ move to positions $l, l+1, \ldots, n-1$, respectively.
- When a job arrives at the queue, it moves into position l with probability $\gamma(l, n+1)$. The jobs previously at positions $l, l+1, \ldots, n$ move to positions $l+1, l+2, \ldots, n+1$, respectively.

Kelly [46] showed that a stationary symmetric queue is insensitive to the service time distribution, given that it can be represented as a mixture of Erlang distributions. This result was extended by Barbour [47] to arbitrarily distributed service times. Taylor [48] showed that the canonical insensitive queueing model, the Erlang loss system, can be described as a symmetric queue.

Now consider the system of multiple queues in our context. Let the number of existing jobs at server j be denoted by n_j. Let the total number of existing jobs in the system be denoted by n; by definition, $n = \sum_{j=1}^{K} n_j$. As in [15], we define a feasible group of server sets $\{\mathcal{T}^{\phi}(n) : 1 \le n \le \widetilde{B}_K\}$ for each policy ϕ, where $\mathcal{T}^{\phi}(n)$ is the set of servers designated for serving the existing jobs in the system at state n. For stationary policies, the assignment decisions are made only upon arrivals and departures.

Let the buffer position in the multi-queue system be defined using a 2-tuple notation (j, k) for position k of server j, where $1 \le j \le K$ and $1 \le k \le B_j$. For each policy ϕ, a one-to-one mapping Θ^{ϕ} from the buffer positions in the multi-queue system, defined as $\mathcal{Q}_{\mathrm{m}} = \{(j,k) : 1 \le j \le K,\ 1 \le k \le B_j\}$, to the buffer positions in the *logically-combined queue*, defined as $\mathcal{Q}_{\mathrm{s}} = \{l : 1 \le l \le \widetilde{B}_K\}$, matches the server sets included in $\{\mathcal{T}^{\phi}(n)\}$. Note that there could be more than one such mapping Θ^{ϕ} that matches $\{\mathcal{T}^{\phi}(n)\}$. Since the service discipline at each server is PS, all the relevant mappings associated with the positions of a given server are equivalent. Moreover, because of the insensitivity property of the logically-combined queue, following the symmetric queue construction of [46] that we shall show below, all these mappings give the same stationary distribution of the underlying stochastic process $\{n^{\phi}(t),\ t \ge 0\}$.

For any mapping Θ^{ϕ} of policy ϕ, the multi-queue system can be implemented as a symmetric queue on the $\mathcal{Q}_{\mathrm{s}}$ domain. To show this, we begin by deriving the state-dependent service rates of $\mathcal{Q}_{\mathrm{s}}$. With the logically-combined queue, each server j serves only the jobs located at its associated positions using the PS discipline. For policy ϕ at state n, the total service rate $\mu^{\phi}(n)$ is given by

$$\mu^{\phi}(n) \stackrel{\mathrm{def}}{=} \sum_{j \in \mathcal{T}^{\phi}(n)} \mu_j. \tag{7}$$

Under the PS discipline, the proportion of $\mu^{\phi}(n)$ allocated to the job at position $l \stackrel{\mathrm{def}}{=} \Theta^{\phi}(j,k)$ on the $\mathcal{Q}_{\mathrm{s}}$ domain is equivalent to that allocated to the job at its corresponding position k of server j on the $\mathcal{Q}_{\mathrm{m}}$ domain, and is given by

$$\gamma^{\phi}(l, n) = \frac{\mu_j}{n_j \mu^{\phi}(n)}. \tag{8}$$

To complete the construction of the logically-combined queue as a symmetric queue, it remains to enforce a symmetry in the same manner as [46] between the service rate allocated to each position in the logically-combined queue and the probability that a newly arrived job will join the queue in the corresponding position. That is:

- When the job at position $l = \Theta^{\phi}(j,k)$ with (j,k) being its corresponding position in the multi-queue system completes its service and leaves the logically-combined queue, the jobs at positions $l+1, l+2, \ldots, n$ move to positions $l, l+1, \ldots, n-1$, respectively.

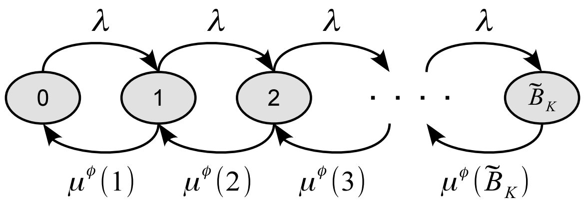


Fig. 2. State transition diagram of the logically-combined queue under any insensitive jockeying policy ϕ.

- When a job arrives at the logically-combined queue, it is assigned to position l, which corresponds to position (j,k) in the multi-queue system where $l = \Theta^\phi(j,k)$, with probability $\gamma^\phi(l, n+1)$. The jobs previously at positions $l, l+1, \ldots, n$ move to positions $l+1, l+2, \ldots, n+1$, respectively.

In general, based on the one-to-one position mapping Θ^ϕ from $\mathcal{Q}_{\mathrm{m}}$ to $\mathcal{Q}_{\mathrm{s}}$, the movements of jobs in the multi-queue system match the movements of jobs in the logically-combined queue.

Due to the insensitivity property of the symmetric queue, the state transition process of the logically-combined queue in this context can be modeled as a birth-death process with birth rate λ and death rate $\mu^\phi(n)$, $n = 1, 2, \ldots, \widetilde{B}_K$, as shown in Fig. 2. Accordingly, the stationary distribution $\pi^\phi(n)$ of the process $\{n^\phi(t),\ t \geq 0\}$, under any insensitive jockeying policy ϕ, can be obtained by solving the steady-state equations:

$$\lambda \pi^\phi(n) = \mu^\phi(n+1)\pi^\phi(n+1),\ n = 0, 1, \ldots, \widetilde{B}_K - 1. \quad (9)$$

Then, the job throughput of the system under policy ϕ, which is equivalent to the long-run average job departure rate, can be obtained as

$$\mathcal{L}^\phi = \sum_{n=1}^{\widetilde{B}_K} \mu^\phi(n)\pi^\phi(n). \quad (10)$$

Alternatively, we can obtain $\mathcal{L}^\phi$ by

$$\mathcal{L}^\phi = \lambda \left[1 - \pi^\phi(\widetilde{B}_K)\right]. \quad (11)$$

The power consumption of the system under policy ϕ, which is equivalent to the long-run average energy consumption rate, can be obtained as

$$\mathcal{E}^\phi = \sum_{n=1}^{\widetilde{B}_K} \varepsilon^\phi(n)\pi^\phi(n) \quad (12)$$

where $\varepsilon^\phi(n)$ is the total energy consumption rate at state n, given by

$$\varepsilon^\phi(n) \stackrel{\text{def}}{=} \sum_{j \in \mathcal{T}^\phi(n)} \varepsilon_j. \quad (13)$$

By definition, $\mathcal{L}^\phi / \mathcal{E}^\phi$ is the energy efficiency of the system under policy ϕ.

B. Insensitive MEESF

The insensitive MEESF policy is derived by specifying the set of servers $\mathcal{T}^{\mathrm{MEESF}}(n)$ designated for serving the existing jobs in the system at state n as

$$\mathcal{T}^{\mathrm{MEESF}}(n) = \{1, \ldots, i\},\ 1 \leq n \leq \widetilde{B}_K \quad (14)$$

where $i \leq K$ is the smallest integer satisfying $\sum_{j=1}^{i} B_j \geq n$. The server sets specified in (14) indeed define the MEESF policy, in which the available servers that are most energy efficient are designated for service at each state n.

The position mapping Θ^{MEESF} of MEESF is defined iteratively as follows. The B_1 positions of server 1 (the most energy-efficient server) are mapped to the first B_1 positions of the logically-combined queue that are associated with server 1. The B_2 positions of server 2 (the second most energy-efficient server) are mapped to the following B_2 positions of the logically-combined queue that are associated with server 2. The procedure continues until the B_K positions of server K have been mapped.

C. Insensitive E*

The insensitive E* policy is derived by specifying the set of servers $\mathcal{T}^{\mathrm{E}^*}(n)$ designated for serving the existing jobs in the system at state n as

$$\mathcal{T}^{\mathrm{E}^*}(n) = \{1, \ldots, \min(n, \widehat{K})\},\ 1 \leq n \leq \widetilde{B}_{\widehat{K}} - 1 \quad (15)$$

and

$$\mathcal{T}^{\mathrm{E}^*}(n) = \{1, \ldots, i\},\ \widetilde{B}_{\widehat{K}} \leq n \leq \widetilde{B}_K \quad (16)$$

where $i \leq K$ is the smallest integer satisfying $\sum_{j=1}^{i} B_j \geq n$. The server sets specified in (15) and (16) indeed define the E* policy, in which preference is always given to the virtual server at any state n.

The position mapping Θ^{E^*} of E* is defined iteratively as follows. In the first iteration, the first buffer positions of servers $1, 2, \ldots, \widehat{K}$ are mapped to the first $\widehat{K}$ positions of the logically-combined queue in the order of the server labels $1, 2, \ldots, \widehat{K}$, inheriting their original server speeds and energy consumption rates. In every subsequent iteration until all the positions of the first $\widehat{K}$ servers have been mapped, the next remaining position levels from the remaining buffers, say $m \leq \widehat{K}$ positions, are mapped to the next m positions of the logically-combined queue in the order of the server labels. Then, the $B_{\widehat{K}+1}$ positions of server $\widehat{K}+1$ are mapped to the following $B_{\widehat{K}+1}$ positions of the logically-combined queue that are associated with server $\widehat{K}+1$. The $B_{\widehat{K}+2}$ positions of server $\widehat{K}+2$ are mapped to the following $B_{\widehat{K}+2}$ positions of the logically-combined queue that are associated with server $\widehat{K}+2$. The iterations terminate when all the positions of all server buffers have been mapped.

VI. Analysis

Here, we provide a rigorous analysis of the E* policy. First, we show that E* has always a higher job throughput than that of MEESF. Then, we derive conditions under which E* is guaranteed to outperform MEESF in terms of the energy efficiency of the system.

For convenience, let

$$P(n) = \frac{\pi^{\mathrm{E}^*}(n)}{\pi^{\mathrm{E}^*}(\widetilde{B}_K)},\quad 0 \leq n \leq \widetilde{B}_K. \quad (17)$$

Then, from (9), we derive for the E* policy that

$$P(n) = \begin{cases} \prod_{i=n+1}^{\widetilde{B}_K} \frac{\mu^{\mathrm{E}^*}(i)}{\lambda}, & 0 \le n \le \widetilde{B}_K - 1 \\ 1, & n = \widetilde{B}_K. \end{cases} \tag{18}$$

Similarly, let

$$P'(n) = \frac{\pi^{\mathrm{MEESF}}(n)}{\pi^{\mathrm{MEESF}}(\widetilde{B}_K)}, \quad 0 \le n \le \widetilde{B}_K \tag{19}$$

and we obtain from (9) for the MEESF policy that

$$P'(n) = \begin{cases} \prod_{i=n+1}^{\widetilde{B}_K} \frac{\mu^{\mathrm{MEESF}}(i)}{\lambda}, & 0 \le n \le \widetilde{B}_K - 1 \\ 1, & n = \widetilde{B}_K. \end{cases} \tag{20}$$

Remark 2. *Because of the nature of the two policies, we have*

$$\begin{cases} \mu^{\mathrm{E}^*}(n) > \mu^{\mathrm{MEESF}}(n), & 2 \le n \le \widetilde{B}_{\widehat{K}-1} \\ \mu^{\mathrm{E}^*}(n) = \mu^{\mathrm{MEESF}}(n), & \textit{elsewhere.} \end{cases} \tag{21}$$

Lemma 1. *For $P(n)$ defined in the form of (18) and $P'(n)$ defined in the form of (20), we have*

$$\begin{cases} P(n) > P'(n), & 0 \le n \le \widetilde{B}_{\widehat{K}-1} - 1 \\ P(n) = P'(n), & \widetilde{B}_{\widehat{K}-1} \le n \le \widetilde{B}_K. \end{cases} \tag{22}$$

Proof: The result follows from Remark 2. ■

Proposition 2. *For the stochastic job assignment problem studied in this paper, we have*

$$\mathcal{L}^{\mathrm{E}^*} > \mathcal{L}^{\mathrm{MEESF}}. \tag{23}$$

Proof: Using (11), we obtain the job throughput of the system under the E* policy and that under the MEESF policy, respectively, as

$$\mathcal{L}^{\mathrm{E}^*} = \lambda \left[1 - \pi^{\mathrm{E}^*}(\widetilde{B}_K)\right] \tag{24}$$

and

$$\mathcal{L}^{\mathrm{MEESF}} = \lambda \left[1 - \pi^{\mathrm{MEESF}}(\widetilde{B}_K)\right]. \tag{25}$$

From (17), we have for the E* policy that

$$\pi^{\mathrm{E}^*}(n) = P(n)\pi^{\mathrm{E}^*}(\widetilde{B}_K), \quad 0 \le n \le \widetilde{B}_K - 1. \tag{26}$$

By normalization, we have

$$\sum_{n=0}^{\widetilde{B}_K - 1} P(n)\pi^{\mathrm{E}^*}(\widetilde{B}_K) + \pi^{\mathrm{E}^*}(\widetilde{B}_K) = 1 \tag{27}$$

and hence

$$\pi^{\mathrm{E}^*}(\widetilde{B}_K) = \frac{1}{\sum_{n=0}^{\widetilde{B}_K - 1} P(n) + 1}. \tag{28}$$

Likewise, from (19), we have for the MEESF policy that

$$\pi^{\mathrm{MEESF}}(n) = P'(n)\pi^{\mathrm{MEESF}}(\widetilde{B}_K), \quad 0 \le n \le \widetilde{B}_K - 1 \tag{29}$$

and hence obtain

$$\pi^{\mathrm{MEESF}}(\widetilde{B}_K) = \frac{1}{\sum_{n=0}^{\widetilde{B}_K - 1} P'(n) + 1}. \tag{30}$$

It follows from Lemma 1 that

$$\sum_{n=0}^{\widetilde{B}_K - 1} P(n) > \sum_{n=0}^{\widetilde{B}_K - 1} P'(n). \tag{31}$$

Therefore, we have

$$\pi^{\mathrm{E}^*}(\widetilde{B}_K) < \pi^{\mathrm{MEESF}}(\widetilde{B}_K) \tag{32}$$

and hence (23). ■

Lemma 2. *Given $P(n)$ defined in the form of (18) and $P'(n)$ defined in the form of (20), for any two integers x and y where $x < y \le K$, we have*

$$\sum_{n=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K} P(n) \sum_{n'=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K} P'(n') - \sum_{n=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K} P(n) \sum_{n'=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K} P'(n') \begin{cases} > 0, & 1 \le x \le \widehat{K} - 1 \\ = 0, & x \ge \widehat{K}. \end{cases} \tag{33}$$

Proof: Note that

$$\sum_{n=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K} P(n) = \sum_{n=\widetilde{B}_{x-1}+1}^{\widetilde{B}_{y-1}} P(n) + \sum_{n=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K} P(n) \tag{34}$$

and

$$\sum_{n'=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K} P'(n') = \sum_{n'=\widetilde{B}_{x-1}+1}^{\widetilde{B}_{y-1}} P'(n') + \sum_{n'=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K} P'(n'). \tag{35}$$

Thus, proving (33) is equivalent to proving

$$\sum_{n=\widetilde{B}_{x-1}+1}^{\widetilde{B}_{y-1}} P(n) \sum_{n'=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K} P'(n') - \sum_{n=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K} P(n) \sum_{n'=\widetilde{B}_{x-1}+1}^{\widetilde{B}_{y-1}} P'(n') \begin{cases} > 0, & 1 \le x \le \widehat{K} - 1 \\ = 0, & x \ge \widehat{K}. \end{cases} \tag{36}$$

It suffices to show that, for any two integers l and m where $\widetilde{B}_{x-1} + 1 \le l \le \widetilde{B}_{y-1}$ and $\widetilde{B}_{y-1} + 1 \le m \le \widetilde{B}_K$, we have

$$P(l)P'(m) - P(m)P'(l) > 0 \tag{37}$$

or, equivalently,

$$\frac{P(l)}{P(m)} = \prod_{i=l+1}^{m} \frac{\mu^{\mathrm{E}^*}(i)}{\lambda} > \frac{P'(l)}{P'(m)} = \prod_{i=l+1}^{m} \frac{\mu^{\mathrm{MEESF}}(i)}{\lambda} \tag{38}$$

if $l \le \widetilde{B}_{\widehat{K}-1} - 1$, and we have

$$P(l)P'(m) - P(m)P'(l) = 0 \tag{39}$$

or, equivalently,

$$\frac{P(l)}{P(m)} = \prod_{i=l+1}^{m} \frac{\mu^{\mathrm{E}^*}(i)}{\lambda} = \frac{P'(l)}{P'(m)} = \prod_{i=l+1}^{m} \frac{\mu^{\mathrm{MEESF}}(i)}{\lambda} \tag{40}$$

if $l \ge \widetilde{B}_{\widehat{K}-1}$. The inequality in (38) and the equality in (40) for all l in the defined range follow from Remark 2. ■

Proposition 3. *A sufficient condition for*

$$\frac{\mathcal{L}^{\mathrm{E}^*}}{\mathcal{E}^{\mathrm{E}^*}} > \frac{\mathcal{L}^{\mathrm{MEESF}}}{\mathcal{E}^{\mathrm{MEESF}}} \tag{41}$$

to hold is that

$$\frac{\mu_j}{\varepsilon_j} = \frac{\mu_1}{\varepsilon_1}, \quad j = 2, 3, \ldots, \widehat{K} \tag{42}$$

and there exists at least one pair of servers x and y, where $1 \le x \le \widehat{K}$ and $x < y \le K$, such that $\mu_x/\varepsilon_x > \mu_y/\varepsilon_y$.

Proof: From (12), we derive for E* that

$$\begin{aligned}\mathcal{E}^{\mathrm{E}^*} &= \sum_{n=1}^{\widetilde{B}_K} \varepsilon^{\mathrm{E}^*}(n)\pi^{\mathrm{E}^*}(n) \\ &= \sum_{n=1}^{\widehat{K}} \pi^{\mathrm{E}^*}(n) \sum_{j=1}^{n} \varepsilon_j + \sum_{n=\widehat{K}+1}^{\widetilde{B}_{\widehat{K}}} \pi^{\mathrm{E}^*}(n) \sum_{j=1}^{\widehat{K}} \varepsilon_j \\ &\quad + \sum_{i=\widehat{K}+1}^{K} \sum_{n=\widetilde{B}_{i-1}+1}^{\widetilde{B}_i} \pi^{\mathrm{E}^*}(n) \sum_{j=1}^{i} \varepsilon_j.\end{aligned} \tag{43}$$

Interchanging the summations in (43), we obtain

$$\begin{aligned}\mathcal{E}^{\mathrm{E}^*} &= \sum_{j=1}^{\widehat{K}} \varepsilon_j \sum_{n=j}^{\widehat{K}} \pi^{\mathrm{E}^*}(n) + \sum_{j=1}^{\widehat{K}} \varepsilon_j \sum_{n=\widehat{K}+1}^{\widetilde{B}_{\widehat{K}}} \pi^{\mathrm{E}^*}(n) \\ &\quad + \left(\sum_{j=1}^{\widehat{K}} \varepsilon_j \sum_{i=\widehat{K}+1}^{K} + \sum_{j=\widehat{K}+1}^{K} \varepsilon_j \sum_{i=j}^{K} \right) \sum_{n=\widetilde{B}_{i-1}+1}^{\widetilde{B}_i} \pi^{\mathrm{E}^*}(n) \\ &= \sum_{j=1}^{\widehat{K}} \varepsilon_j \sum_{n=j}^{\widetilde{B}_K} \pi^{\mathrm{E}^*}(n) + \sum_{j=\widehat{K}+1}^{K} \varepsilon_j \sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K} \pi^{\mathrm{E}^*}(n).\end{aligned} \tag{44}$$

Note that, since $B_j > 1$ for all j, we have $\widetilde{B}_{j-1} + 1 \ge j$ for $1 \le j \le \widehat{K}$. Thus, we can rewrite the elements of $\mathcal{E}^{\mathrm{E}^*}$ in (44) as

$$\begin{aligned}\mathcal{E}^{\mathrm{E}^*} &= \sum_{j=1}^{\widehat{K}} \varepsilon_j \sum_{n=j}^{\widetilde{B}_{j-1}} \pi^{\mathrm{E}^*}(n) + \sum_{j=1}^{\widehat{K}} \varepsilon_j \sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K} \pi^{\mathrm{E}^*}(n) \\ &\quad + \sum_{j=\widehat{K}+1}^{K} \varepsilon_j \sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K} \pi^{\mathrm{E}^*}(n) \\ &= \sum_{j=1}^{\widehat{K}} \varepsilon_j \sum_{n=j}^{\widetilde{B}_{j-1}} \pi^{\mathrm{E}^*}(n) + \sum_{j=1}^{K} \varepsilon_j \sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K} \pi^{\mathrm{E}^*}(n).\end{aligned} \tag{45}$$

Similar to the way we derive the expression of $\mathcal{E}^{\mathrm{E}^*}$ in (45), we obtain $\mathcal{L}^{\mathrm{E}^*}$ as

$$\begin{aligned}\mathcal{L}^{\mathrm{E}^*} &= \sum_{n=1}^{\widetilde{B}_K} \mu^{\mathrm{E}^*}(n)\pi^{\mathrm{E}^*}(n) \\ &= \sum_{j=1}^{\widehat{K}} \mu_j \sum_{n=j}^{\widetilde{B}_{j-1}} \pi^{\mathrm{E}^*}(n) + \sum_{j=1}^{K} \mu_j \sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K} \pi^{\mathrm{E}^*}(n).\end{aligned} \tag{46}$$

Then, using (17), we have

$$\begin{aligned}\frac{\mathcal{L}^{\mathrm{E}^*}}{\mathcal{E}^{\mathrm{E}^*}} &= \frac{\mathcal{L}^{\mathrm{E}^*}/\pi^{\mathrm{E}^*}(\widetilde{B}_K)}{\mathcal{E}^{\mathrm{E}^*}/\pi^{\mathrm{E}^*}(\widetilde{B}_K)} \\ &= \frac{\sum_{j=1}^{\widehat{K}} \mu_j \sum_{n=j}^{\widetilde{B}_{j-1}} P(n) + \sum_{j=1}^{K} \mu_j \sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K} P(n)}{\sum_{j=1}^{\widehat{K}} \varepsilon_j \sum_{n=j}^{\widetilde{B}_{j-1}} P(n) + \sum_{j=1}^{K} \varepsilon_j \sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K} P(n)}.\end{aligned} \tag{47}$$

On the other hand, from (12), we derive for MEESF that

$$\begin{aligned}\mathcal{E}^{\mathrm{MEESF}} &= \sum_{n'=1}^{\widetilde{B}_K} \varepsilon^{\mathrm{MEESF}}(n')\pi^{\mathrm{MEESF}}(n') \\ &= \sum_{i'=1}^{K} \sum_{n'=\widetilde{B}_{i'-1}+1}^{\widetilde{B}_{i'}} \pi^{\mathrm{MEESF}}(n') \sum_{j'=1}^{i'} \varepsilon_{j'}.\end{aligned} \tag{48}$$

Interchanging the summations in (48), we obtain

$$\begin{aligned}\mathcal{E}^{\mathrm{MEESF}} &= \sum_{j'=1}^{K} \varepsilon_{j'} \sum_{i'=j'}^{K} \sum_{n'=\widetilde{B}_{i'-1}+1}^{\widetilde{B}_{i'}} \pi^{\mathrm{MEESF}}(n') \\ &= \sum_{j'=1}^{K} \varepsilon_{j'} \sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K} \pi^{\mathrm{MEESF}}(n').\end{aligned} \tag{49}$$

Similar to the way we derive the expression of $\mathcal{E}^{\mathrm{MEESF}}$ in (49), we obtain $\mathcal{L}^{\mathrm{MEESF}}$ as

$$\begin{aligned}\mathcal{L}^{\mathrm{MEESF}} &= \sum_{n'=1}^{\widetilde{B}_K} \mu^{\mathrm{MEESF}}(n')\pi^{\mathrm{MEESF}}(n') \\ &= \sum_{j'=1}^{K} \mu_{j'} \sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K} \pi^{\mathrm{MEESF}}(n').\end{aligned} \tag{50}$$

Then, using (19), we have

$$\begin{aligned}\frac{\mathcal{L}^{\mathrm{MEESF}}}{\mathcal{E}^{\mathrm{MEESF}}} &= \frac{\mathcal{L}^{\mathrm{MEESF}}/\pi^{\mathrm{MEESF}}(\widetilde{B}_K)}{\mathcal{E}^{\mathrm{MEESF}}/\pi^{\mathrm{MEESF}}(\widetilde{B}_K)} \\ &= \frac{\sum_{j'=1}^{K} \mu_{j'} \sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K} P'(n')}{\sum_{j'=1}^{K} \varepsilon_{j'} \sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K} P'(n')}.\end{aligned} \tag{51}$$

Clearly, given $\mathcal{L}^{\mathrm{E}^*}/\mathcal{E}^{\mathrm{E}^*}$ in the form of (47) and $\mathcal{L}^{\mathrm{MEESF}}/\mathcal{E}^{\mathrm{MEESF}}$ in the form of (51), for the inequality in

(41) to hold, it requires

$$\sum_{j=1}^{\widehat{K}}\sum_{j'=1}^{K}\mu_j\varepsilon_{j'}\sum_{n=j}^{\widetilde{B}_{j-1}}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n')$$
$$+\sum_{j=1}^{K}\sum_{j'=1}^{K}\mu_j\varepsilon_{j'}\sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n')$$
$$>\sum_{j=1}^{\widehat{K}}\sum_{j'=1}^{K}\varepsilon_j\mu_{j'}\sum_{n=j}^{\widetilde{B}_{j-1}}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n')$$
$$+\sum_{j=1}^{K}\sum_{j'=1}^{K}\varepsilon_j\mu_{j'}\sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n'). \quad (52)$$

First, we show that

$$\sum_{j=1}^{K}\sum_{j'=1}^{K}\mu_j\varepsilon_{j'}\sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n')$$
$$>\sum_{j=1}^{K}\sum_{j'=1}^{K}\varepsilon_j\mu_{j'}\sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n'). \quad (53)$$

In particular, we observe in (53) that:

- For $j = j'$, we have
$$\mu_j\varepsilon_{j'}\sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n')$$
$$=\varepsilon_j\mu_{j'}\sum_{n=\widetilde{B}_{j-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n'). \quad (54)$$
- For any two integers x and y where $1 \le x < y \le K$, we have
$$\mu_x\varepsilon_y\sum_{n=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K}P'(n')$$
$$+\mu_y\varepsilon_x\sum_{n=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K}P'(n')$$
$$-\varepsilon_x\mu_y\sum_{n=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K}P'(n')$$
$$-\varepsilon_y\mu_x\sum_{n=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K}P'(n')$$
$$=(\mu_x\varepsilon_y-\varepsilon_x\mu_y)\Bigg[\sum_{n=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K}P'(n')$$
$$-\sum_{n=\widetilde{B}_{y-1}+1}^{\widetilde{B}_K}P(n)\sum_{n'=\widetilde{B}_{x-1}+1}^{\widetilde{B}_K}P'(n')\Bigg]. \quad (55)$$

Then, it follows from Lemma 2 that the right-hand side of (55)
$$\begin{cases} > 0, & 1 \le x \le \widehat{K}-1 \\ = 0, & x \ge \widehat{K}. \end{cases}$$
where the inequality holds if $\mu_x/\varepsilon_x > \mu_y/\varepsilon_y$.

Now, for the inequality in (52) to hold, it is sufficient to have

$$\sum_{j=1}^{\widehat{K}}\sum_{j'=1}^{K}\mu_j\varepsilon_{j'}\sum_{n=j}^{\widetilde{B}_{j-1}}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n')$$
$$\ge\sum_{j=1}^{\widehat{K}}\sum_{j'=1}^{K}\varepsilon_j\mu_{j'}\sum_{n=j}^{\widetilde{B}_{j-1}}P(n)\sum_{n'=\widetilde{B}_{j'-1}+1}^{\widetilde{B}_K}P'(n'). \quad (56)$$

We observe in (56) that:

- For $j = j'$, we have $\mu_j\varepsilon_{j'} = \varepsilon_j\mu_{j'}$.
- For $j = 1, 2, \ldots, \widehat{K}$ and $j' = j+1, j+2, \ldots, K$, we have $\mu_j\varepsilon_{j'} - \varepsilon_j\mu_{j'} \ge 0$.
- For $j = 2, 3, \ldots, \widehat{K}$ and $j' = 1, 2, \ldots, j-1$, we have $\mu_j\varepsilon_{j'} - \varepsilon_j\mu_{j'} \le 0$.

Therefore, for the inequality in (56) to hold, it is sufficient to have (42), which enforces

$$\mu_j\varepsilon_{j'}-\varepsilon_j\mu_{j'} = 0,\ j = 2, 3, \ldots, \widehat{K},\ j' = 1, 2, \ldots, j-1. \quad (57)$$

This completes the proof. ∎

From Proposition 3, we can obtain the following corollary.

Corollary 1. *If $\mu_j/\varepsilon_j = c$ for $j = 1, 2, \ldots, K$, we have*
$$\frac{\mathcal{L}^{\mathrm{E}^*}}{\mathcal{E}^{\mathrm{E}^*}} = \frac{\mathcal{L}^{\mathrm{MEESF}}}{\mathcal{E}^{\mathrm{MEESF}}}. \quad (58)$$

Corollary 1 suggests that, if all servers in the system are equally energy efficient, the energy efficiency of the system under the E* policy is equivalent to that under the MEESF policy. Nevertheless, even in such case of a homogeneous server farm, E* is guaranteed to yield a higher job throughput than that of MEESF. On the other hand, Proposition 3 suggests that, if at least two servers in a heterogeneous server farm are equally most energy efficient, E* is guaranteed to outperform MEESF in terms of the energy efficiency of the system. We argue that the latter is a realistic scenario since in practice a server farm is likely to comprise multiple servers of the same type purchased at a time.

VII. Numerical Results

In this section, we provide extensive numerical results to demonstrate the effectiveness of the E* policy and the RM policy proposed in this paper. For convenience of describing the experiment setting, we denote by $\rho = \lambda/\sum_{j=1}^{K}\mu_j$ the offered traffic per server in the system.

A. Verification of the exact analysis

In this experiment, we consider a system with four servers. We set $\rho = 0.8$, and for each server j we set its buffer size $B_j = 3$. We generate a sequence of K random deviates according to a Pareto distribution with unit mean and arrange

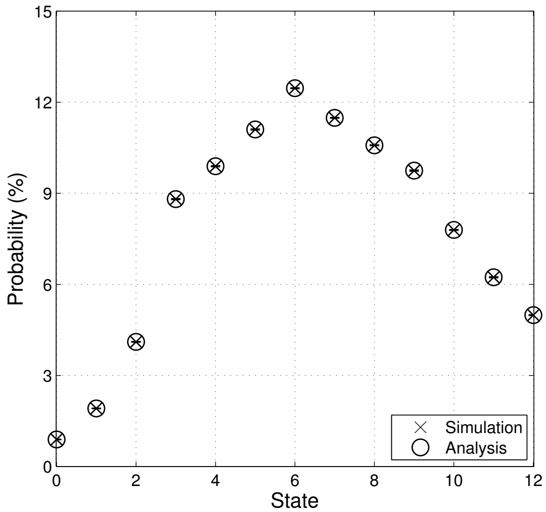


Fig. 3. Verification of the exact analysis of the stationary distribution.

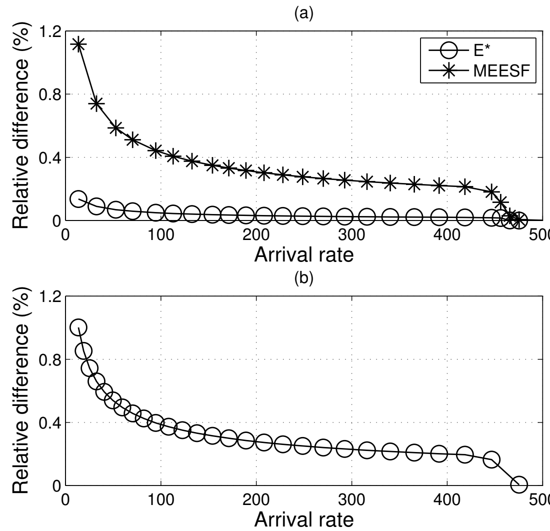


Fig. 4. Performance comparison in terms of the energy efficiency of the system in the special case with respect to the job arrival rate. (a) Relative difference of $\mathcal{L}^{\mathrm{E}^*}/\mathcal{E}^{\mathrm{E}^*}$ and $\mathcal{L}^{\mathrm{MEESF}}/\mathcal{E}^{\mathrm{MEESF}}$ to that of the optimal policy. (b) Relative difference of $\mathcal{L}^{\mathrm{E}^*}/\mathcal{E}^{\mathrm{E}^*}$ to $\mathcal{L}^{\mathrm{MEESF}}/\mathcal{E}^{\mathrm{MEESF}}$.

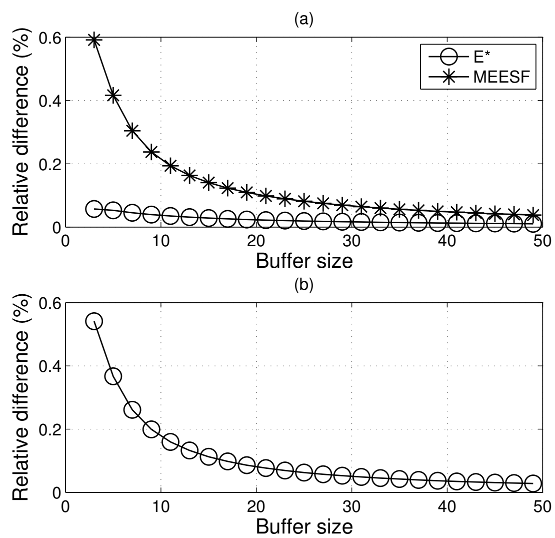


Fig. 5. Performance comparison in terms of the energy efficiency of the system in the special case with respect to the buffer size. (a) Relative difference of $\mathcal{L}^{\mathrm{E}^*}/\mathcal{E}^{\mathrm{E}^*}$ and $\mathcal{L}^{\mathrm{MEESF}}/\mathcal{E}^{\mathrm{MEESF}}$ to that of the optimal policy. (b) Relative difference of $\mathcal{L}^{\mathrm{E}^*}/\mathcal{E}^{\mathrm{E}^*}$ to $\mathcal{L}^{\mathrm{MEESF}}/\mathcal{E}^{\mathrm{MEESF}}$.

them in a non-increasing order of values. The energy efficiency μ_j/ε_j of server j is set as the j-th value in the ordered sequence. The energy consumption rate of server j is chosen to be $\varepsilon_j = 10 + j^2$, from which we obtain the service rate μ_j of server j using the corresponding energy efficiency value.

Fig. 3 provides both simulation results and exact analytical results for the stationary distribution of the number of jobs in the system under the MEESF policy. The simulation results are obtained from the multi-queue system with exponential job size distribution, and presented in the form of an observed mean from ten independent runs of the experiment. The confidence intervals at the 95% level based on the Student's t-distribution are found to be within $\pm 0.4\%$ of the observed mean. The analytical results are obtained from the logically-combined queue by solving the steady-state equations of the Markov chain presented in Fig. 2. We observe in Fig. 3 that the analytical results are all within the simulation confidence intervals, demonstrating a clear agreement between the simulation and the exact analysis.

*B. Effectiveness of E**

Here, we demonstrate the effectiveness of the E* policy by comparing it with the baseline MEESF policy as well as the optimal policy [15] under various system parameters. First, we consider the special case where the energy consumption rate of each server j is related to its service rate by $\varepsilon_j = \mu_j^3$. Recall that, with this setting, MEESF is equivalent to the SSF policy proposed in [15].

For the set of experiments in Fig. 4 and Fig. 5, we have a system with 100 servers; the service rate of each server j is $\mu_j = 0.1j$. Fig. 4 is obtained with the buffer size $B_j = 10$ for each server j and the job arrival rate λ varied from μ_1 to $\mu_1 + \cdots + \mu_{100}$. In Fig. 5, we set $\rho = 0.8$ and all servers have the same buffer size which is varied from 3 to 49 at a step of 2. We observe in Fig. 4(a) and Fig. 5(a) that both MEESF and E* are close to the optimal policy in terms of the energy efficiency of the system, with relative difference less than 1.2% and 0.3%, respectively. We also observe in Fig. 4(b) and Fig. 5(b) that E* outperforms MEESF in all experiments, although the improvement is only up to 1% in this special case. These observations are consistent with the argument of [15] and our analysis in this paper.

For the set of experiments in Fig. 6, we have a system with 50 servers. For each server j, its buffer size B_j is randomly chosen from the set $\{10, 11, \ldots, 15\}$, and its service rate μ_j is randomly generated from the range $[0.1, 10]$ and arranged in a non-decreasing order. For each such random configuration of server speed, we set the job arrival rate such

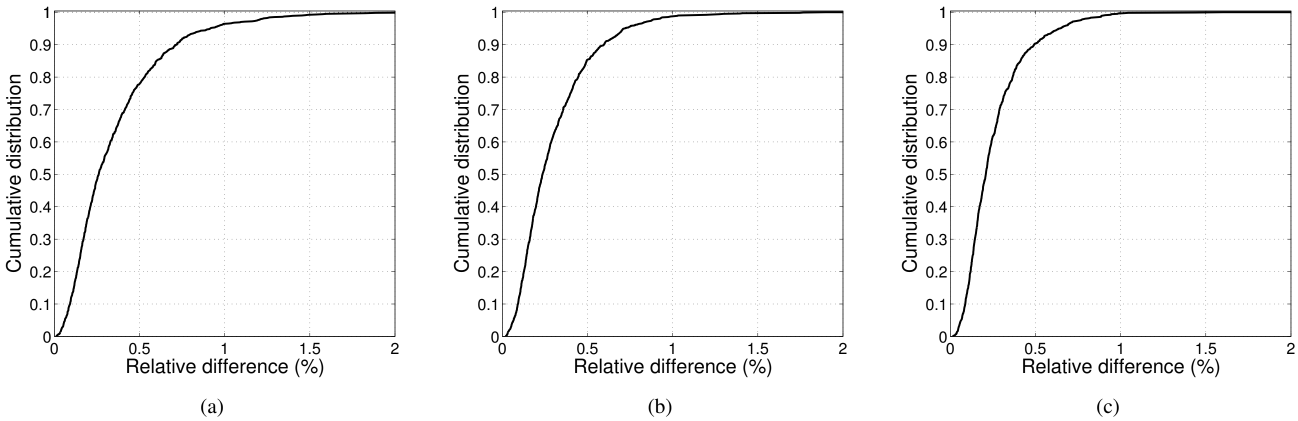


Fig. 6. Cumulative distribution of the relative difference of $\mathcal{L}^{\mathrm{E}^*}/\mathcal{E}^{\mathrm{E}^*}$ to $\mathcal{L}^{\mathrm{MEESF}}/\mathcal{E}^{\mathrm{MEESF}}$ in the special case. (a) $\rho = 0.4$. (b) $\rho = 0.6$. (c) $\rho = 0.8$.

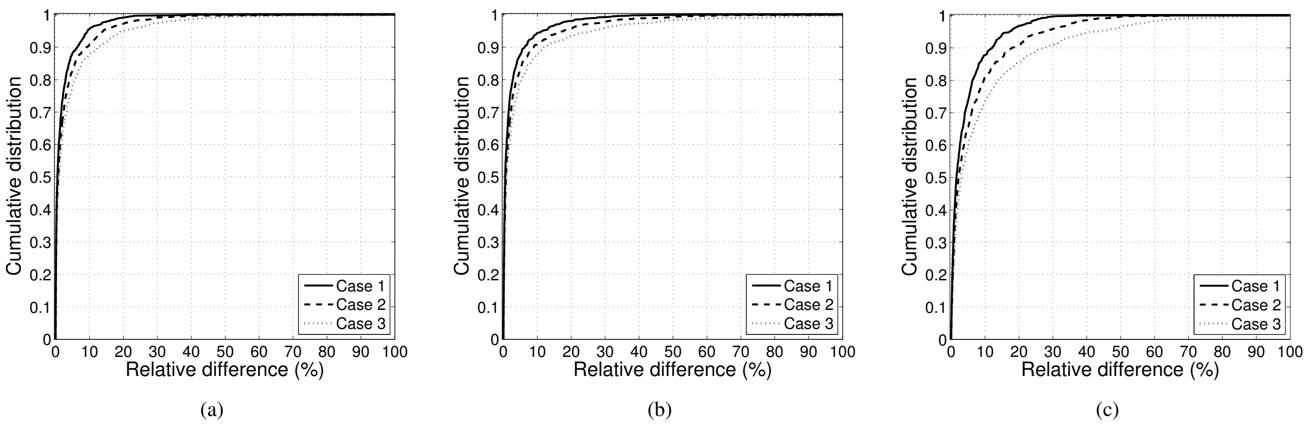


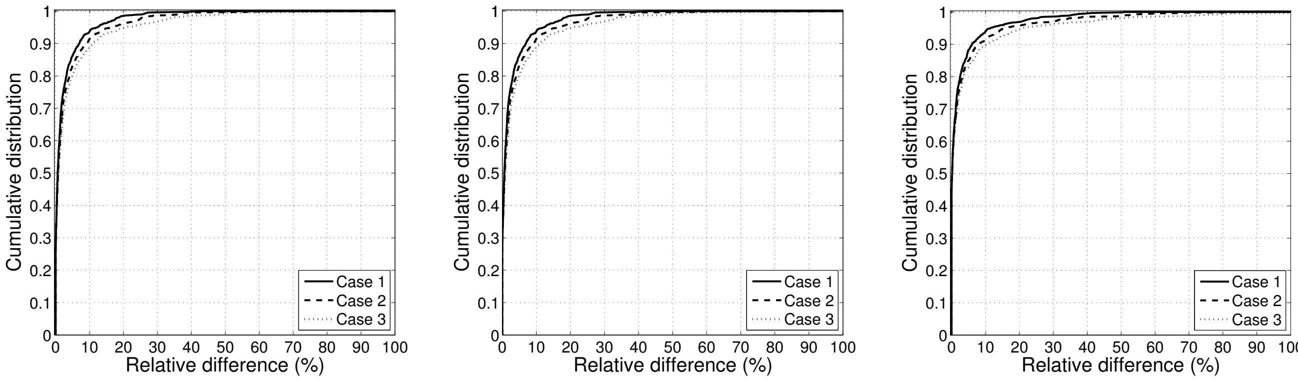
Fig. 7. Cumulative distribution of the relative difference of $\mathcal{L}^{\mathrm{E}^*}/\mathcal{E}^{\mathrm{E}^*}$ to $\mathcal{L}^{\mathrm{MEESF}}/\mathcal{E}^{\mathrm{MEESF}}$ in the general case. (a) $\rho = 0.4$. (b) $\rho = 0.6$. (c) $\rho = 0.8$.

that $\rho = 0.4, 0.6, 0.8$. We compare E* and MEESF in terms of the energy efficiency of the system. Results are obtained from 1000 experiments and are plotted in Fig. 6 in the form of cumulative distribution of the relative difference. From these results, we have similar observations of the two policies to those found in Fig. 4 and Fig. 5 for this special case.

Next, we consider the more general case where the energy consumption rate of each server and its service rate do not necessarily follow the assumption of (2). We shall see that, although the improvement of E* over MEESF in terms of the energy efficiency of the system is very limited in the special case, it can be significantly improved in the general case with independently and randomly generated service rates and energy consumption rates.

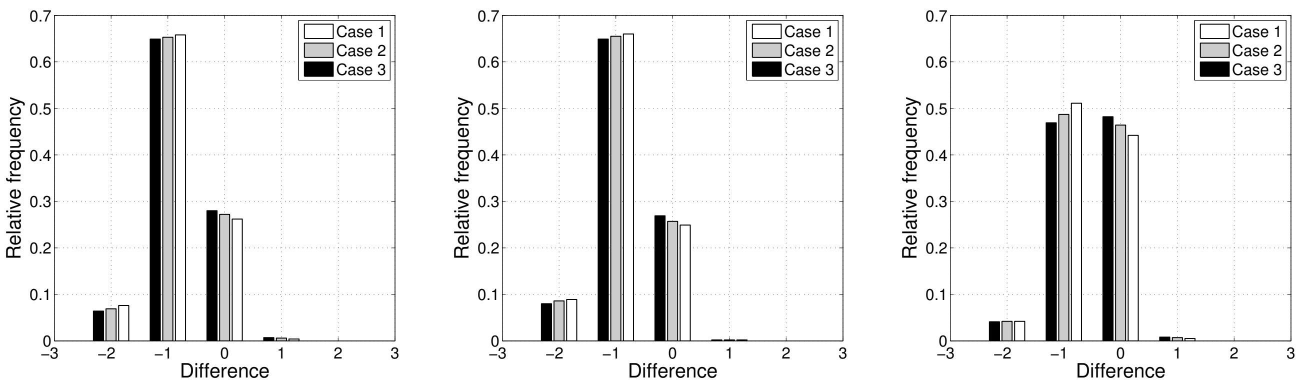
In this set of experiments, we have a system with 50 servers that are categorized into ten server groups. Each server group i, $i = 1, 2, \ldots, 10$, consists of five servers that have the same service rate, energy consumption rate and buffer size, denoted by $\bar{\mu}_i$, $\bar{\varepsilon}_i$ and $\bar{B}_i$, respectively. We randomly generate a set of ratios r_i, $i = 2, 3, \ldots, 10$, from the range $[0.1, 1]$. With $\bar{\mu}_1/\bar{\varepsilon}_1 = 100$, the energy efficiency of each server in server group i is set to be $\bar{\mu}_i/\bar{\varepsilon}_i = r_i^{\alpha}\bar{\mu}_{i-1}/\bar{\varepsilon}_{i-1}$, $i = 2, 3, \ldots, 10$, where we consider three cases for the α value, i.e., $\alpha = 1$ for case 1, $\alpha = 1.2$ for case 2, and $\alpha = 1.4$ for case 3. Note that different values of α in this context represent different levels of server heterogeneity. The set of service rates $\bar{\mu}_i$ is randomly generated from the range $[0.1, 10]$ and is arranged in a non-increasing order. The set of buffer sizes $\bar{B}_i$ is also randomly chosen from $\{10, 11, \ldots, 15\}$. Such a setting can be justified in a way that a more recently purchased server is likely to have a higher service rate and a higher energy efficiency. In practice, a server farm is likely to comprise multiple servers of the same type purchased at a time.

Results in Fig. 7 are again obtained from 1000 experiments and are plotted in the form of cumulative distribution of the relative difference of E* to MEESF in terms of the energy efficiency of the system. We observe in Fig. 7 that, in such a general case, E* significantly outperforms MEESF by up to 100%, which is a substantial improvement of the performance compared to merely 2% in Fig. 6 for the special case. It can also be observed from Fig. 7(c) that E* outperforms MEESF by more than 10% in nearly 27% of the experiments for case 3. In addition, we observe in Fig. 7 that, as the level of server heterogeneity becomes higher, the performance improvement of E* over MEESF becomes larger.



(a) (b) (c)

Fig. 8. Cumulative distribution of the relative difference of $\mathcal{L}^{\mathrm{E}^*}/\mathcal{E}^{\mathrm{E}^*}$ to $\mathcal{L}^{\mathrm{RM}}/\mathcal{E}^{\mathrm{RM}}$ in the general case. (a) $\rho = 0.4$. (b) $\rho = 0.6$. (c) $\rho = 0.8$.



(a) (b) (c)

Fig. 9. Histogram of the difference between $\widehat{K}^*$ and $\widehat{K}^{\mathrm{RM}}$. (a) $\rho = 0.4$. (b) $\rho = 0.6$. (c) $\rho = 0.8$.

C. *Effectiveness of RM*

Here, we demonstrate the effectiveness of the RM policy by comparing it with the E* policy using the same experiment settings for the general case in Fig. 7. Results are plotted in Fig. 8 in the form of cumulative distribution of the relative difference of E* to RM in terms of the energy efficiency of the system. We observe that, in all cases, the relative difference is less than 5% in nearly 80% of the experiments. We also observe that the relative difference is not very sensitive to the value of ρ and the level of server heterogeneity.

In Fig. 9, we present the histogram of the difference between $\widehat{K}^*$ and $\widehat{K}^{\mathrm{RM}}$, i,e., $\widehat{K}^* - \widehat{K}^{\mathrm{RM}}$, obtained from the corresponding experiments for each value of ρ and each level of server heterogeneity. We observe in Fig. 9 that the difference between $\widehat{K}^*$ and $\widehat{K}^{\mathrm{RM}}$ varies within a small range of $\{-3, -2, \ldots, 1\}$. In addition, in Fig. 9, as the value of ρ grows from 0.6 to 0.8, the variance of the distribution of $\widehat{K}^* - \widehat{K}^{\mathrm{RM}}$ decreases. This demonstrates that the optimal value $\widehat{K}^*$ is likely to be within a small range of $\widehat{K}$ values defined by the empirical value $\widehat{K}^{\mathrm{RM}}$. Therefore, the performance of the RM policy can be further improved by attempting each value of $\widehat{K}$ within such a significantly reduced range.

VIII. Conclusion

We have proposed a new approach that gives rise to an insensitive job-assignment policy for the popular server farm model comprising a parallel system of finite-buffer PS queues with heterogeneous server speeds and energy consumption rates. Unlike the straightforward MEESF approach that greedily chooses the most energy-efficient servers for job assignment, one important feature of the more robust E* policy is to aggregate an optimal number of most energy-efficient servers as a virtual server. E* is designed to give preference to this virtual server and utilize its service capacity in such a way that both the job throughput and the energy efficiency of the system can be improved. We have provided a rigorous analysis of the E* policy where it is shown that E* has always a higher job throughput than that of MEESF and there exist realistic and sufficient conditions under which E* is guaranteed to outperform MEESF in terms of the energy efficiency of the system. We have further proposed a rule of thumb to form the virtual server by simply matching its aggregate service rate to the job arrival rate. Extensive experiments based on random settings have confirmed the effectiveness of the resulting RM policy. Noting that the fundamentally important model of parallel PS queues has broader applications in communication systems, our proposed solution for insensitive energy-efficient job assignment has potentially wider applicability to green communications and networking.

Acknowledgment

The authors wish to thank the anonymous reviewers for their valuable comments that contributed to the improved quality of this paper.

References

[1] M. Arregoces and M. Portolani, *Data center fundamentals*. Cisco Press, 2003.

[2] US Environmental Protection Agency, Washington, DC, USA, "EPA report on server and data center energy efficiency," Tech. Rep., 2007.

[3] The Climate Group, "SMART 2020: Enabling the low carbon economy in the information age," Tech. Rep., 2008.

[4] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in Computers*, vol. 82, no. 2, pp. 47–111, 2011.

[5] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *Proc. IEEE FOCS*, Milwaukee, WI, USA, Oct. 1995, pp. 374–382.

[6] N. Bansal, K. Pruhs, and C. Stein, "Speed scaling for weighted flow time," in *Proc. ACM-SIAM SODA*, New Orleans, LA, USA, Jan. 2007, pp. 805–813.

[7] N. Bansal, H.-L. Chan, and K. Pruhs, "Speed scaling with an arbitrary power function," in *Proc. ACM-SIAM SODA*, New York, NY, USA, Jan. 2009, pp. 693–701.

[8] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 2007–2015.

[9] L. L. H. Andrew, M. Lin, and A. Wierman, "Optimality, fairness and robustness in speed scaling designs," in *Proc. ACM SIGMETRICS*, New York, NY, USA, Jun. 2010, pp. 37–48.

[10] M. Andrews, S. Antonakopoulos, and L. Zhang, "Energy-aware scheduling algorithms for network stability," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 1359–1367.

[11] I. Mitrani, "Managing performance and power consumption in a server farm," *Ann. Oper. Res.*, vol. 202, no. 1, pp. 121–134, Jan. 2013.

[12] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1378–1391, Oct. 2013.

[13] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," in *Proc. ACM SIGMETRICS*, Seattle, USA, Jun. 2009, pp. 157–168.

[14] 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*, vol. 75-76, pp. 17–35, May-Jun. 2014.

[15] Z. Rosberg, Y. Peng, J. Fu, J. Guo, E. W. M. Wong, and M. Zukerman, "Insensitive job assignment with throughput and energy criteria for processor-sharing server farms," *IEEE/ACM Trans. Netw.*, vol. 22, no. 4, pp. 1257–1270, Aug. 2014.

[16] V. Cardellini, E. Casalicchio, M. Colajanni, and P. S. Yu, "The state of the art in locally distributed web-server systems," *ACM Computing Surveys*, vol. 34, no. 2, pp. 263–311, Jun. 2002.

[17] V. Gupta, K. Sigman, M. Harchol-Balter, and W. Whitt, "Insensitivity for PS server farms with JSQ routing," *ACM SIGMETRICS PER*, vol. 35, no. 2, pp. 24–26, Sep. 2007.

[18] V. Gupta, M. Harchol-Balter, K. Sigman, and W. Whitt, "Analysis of join-the-shortest-queue routing for web server farms," *Performance Evaluation*, vol. 64, no. 9-12, pp. 1062–1081, Oct. 2007.

[19] E. Altman, U. Ayesta, and B. Prabhu, "Load balancing in processor sharing systems," *Telecommun. Syst.*, vol. 47, no. 1-2, pp. 35–48, Jun. 2011.

[20] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, Dec. 1997.

[21] S. Gunawardena and W. Zhuang, "Service response time of elastic data traffic in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 3, pp. 559–570, Mar. 2013.

[22] F. Liu, K. Zheng, W. Xiang, and H. Zhao, "Design and performance analysis of an energy-efficient uplink carrier aggregation scheme," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 2, pp. 197–207, Feb. 2014.

[23] J. Ousterhout, P. Agrawal, D. Erickson, C. Kozyrakis, J. Leverich, D. Mazières, S. Mitra, A. Narayanan, G. Parulkar, M. Rosenblum, S. M. Rumble, E. Stratmann, and R. Stutsman, "The case for RAMClouds: Scalable high-performance storage entirely in DRAM," *SIGOPS Operating Systems Review*, vol. 43, no. 4, pp. 92–105, Dec. 2009.

[24] A. M. Caulfield, L. M. Grupp, and S. Swanson, "Gordon: Using flash memory to build fast, power-efficient clusters for data-intensive applications," *ACM SIGPLAN Notices*, vol. 44, no. 3, pp. 217–228, Mar. 2009.

[25] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 909–928, Second Quarter 2013.

[26] J. Fu, J. Guo, E. W. M. Wong, and M. Zukerman, "Energy-efficient heuristics for job assignment in processor-sharing server farms," in *Proc. IEEE INFOCOM*, Hong Kong SAR, China, Apr. 2015.

[27] F. A. Haight, "Two queues in parallel," *Biometrika*, vol. 45, no. 3/4, pp. 401–410, Dec. 1958.

[28] W. Winston, "Optimality of the shortest line discipline," *J. Appl. Probab.*, vol. 14, no. 1, pp. 181–189, Mar. 1977.

[29] R. R. Weber, "On the optimal assignment of customers to parallel servers," *J. Appl. Probab.*, vol. 15, no. 2, pp. 406–413, Jun. 1978.

[30] A. Ephremides, P. Varaiya, and J. Walrand, "A simple dynamic routing problem," *IEEE Trans. Autom. Control*, vol. 25, no. 4, pp. 690–693, Aug. 1980.

[31] F. Bonomi, "On job assignment for a parallel system of processor sharing queues," *IEEE Trans. Comput.*, vol. 39, no. 7, pp. 858–869, Jul. 1990.

[32] W. Whitt, "Deciding which queue to join: some counterexamples," *Oper. Res.*, vol. 34, no. 1, pp. 55–62, Feb. 1986.

[33] V. Gupta, "Stochastic models and analysis for resource management in server farms," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 2011.

[34] Y. Zhao and W. K. Grassmann, "Queueing analysis of a jockeying model," *Oper. Res.*, vol. 43, no. 3, pp. 520–529, May–Jun. 1995.

[35] I. J. B. F. Adan, J. Wessels, and W. H. M. Zijm, "Matrix-geometric analysis of the shortest queue problem with threshold jockeying," *Oper. Res. Lett.*, vol. 13, no. 2, pp. 107–112, Mar. 1993.

[36] Y. Sakuma, "Asymptotic behavior for MAP/PH/c queue with shortest queue discipline and jockeying," *Oper. Res. Lett.*, vol. 38, no. 1, pp. 7–10, Jan. 2010.

[37] E. Hyytiä, R. Righter, and S. Aalto, "Energy-aware job assignment in server farms with setup delays under LCFS and PS," in *Proc. ITC 26*, Karlskrona, Sweden, Sep. 2014.

[38] K. Li, X. Tang, and K. Li, "Energy-efficient stochastic task scheduling on heterogeneous computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2867–2876, Nov. 2014.

[39] D. Bertsekas, *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995.

[40] S. A. Lippman, "Applying a new device in the optimization of exponential queuing systems," *Oper. Res.*, vol. 23, no. 4, pp. 687–710, Jul.–Aug. 1975.

[41] J. Wijngaard and S. Stidham, "Forward recursion for Markov decision processes with skip-free-to-the-right transitions, part I: theory and algorithm," *Math. Oper. Res.*, vol. 11, no. 2, pp. 295–308, May 1986.

[42] S. Stidham and R. R. Weber, "Monotonic and insensitive optimal policies for control of queues with undiscounted costs," *Oper. Res.*, vol. 37, no. 4, pp. 611–625, Jul.–Aug. 1989.

[43] J. M. George and J. M. Harrison, "Dynamic control of a queue with adjustable service rate," *Oper. Res.*, vol. 49, no. 5, pp. 720–731, Sep.–Oct. 2001.

[44] S. Albers and H. Fujiwara, "Energy-efficient algorithms for flow time minimization," *ACM Transactions on Algorithms*, vol. 3, no. 4, Nov. 2007, Article No. 49.

[45] J. Fu, "Energy-efficient job assignment in server farms," Ph.D. dissertation, City University of Hong Kong, Hong Kong SAR, China, 2016.

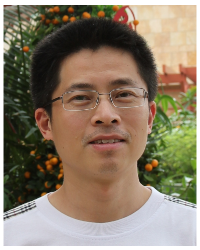
[46] F. P. Kelly, "Networks of queues," *Adv. Appl. Probab.*, vol. 8, no. 2, pp. 416–432, Jun. 1976.

[47] A. D. Barbour, "Networks of queues and the method of stages," *Adv. Appl. Probab.*, vol. 8, no. 3, pp. 584–591, Sep. 1976.

[48] P. G. Taylor, "Insensitivity in stochastic models," in *Queueing Networks*, R. J. Boucherie and N. M. van Dijk, Eds. New York, NY: Springer, 2011, ch. 3, pp. 121–140.

Jing Fu (S'15) received the B.Eng. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2011, and is currently pursuing the Ph.D. degree in electronic engineering at City University of Hong Kong, Hong Kong.

Her research interest is now focused on energy efficiency in server farms.



Jun Guo (S'01–M'06) received the B.E. degree in automatic control engineering from Shanghai University of Science and Technology, Shanghai, China, in 1992, and the M.E. degree in telecommunications engineering and Ph.D. degree in electrical and electronic engineering from the University of Melbourne, Melbourne, Australia, in 2001 and 2006, respectively.

He was with the School of Computer Science and Engineering, The University of New South Wales, Kensington, Australia, as a Senior Research Associate from 2006 to 2008 and on an Australian Postdoctoral Fellowship (APD) supported by the Australian Research Council (ARC) from 2009 to 2011. Since 2012, he has been with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, where he is now a Visiting Assistant Professor. His research is currently focused on green communications and networking, teletraffic theory and its applications in service sectors, and survivable network topology design.

Eric W. M. Wong (S'87–M'90–SM'00) received the B.Sc. and M.Phil. degrees in electronic engineering from the Chinese University of Hong Kong, Hong Kong, in 1988 and 1990, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, MA, USA, in 1994.

He is an Associate Professor with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. His research interests include analysis and design of telecommunications and computer networks, energy-efficient data center design, green cellular networks and optical switching.

Moshe Zukerman (M'87–SM'91–F'07) received the B.Sc. degree in industrial engineering and management and the M.Sc. degree in operations research from the Technion—Israel Institute of Technology, Haifa, Israel, in 1976 and 1979, respectively, and the Ph.D. degree in engineering from the University of California, Los Angeles, CA, USA, in 1985.

He was an independent consultant with the IRI Corporation and a Postdoctoral Fellow with the University of California, Los Angeles, from 1985 to 1986. He was with the Telstra Research Laboratories (TRL), Melbourne, Australia, first as a Research Engineer from 1986 to 1988, and as a Project Leader from 1988 to 1997. He also taught and supervised graduate students with Monash University, Melbourne, Australia, from 1990 to 2001. From 1997 to 2008, he was with The University of Melbourne, Melbourne, Australia. In 2008, he joined City University of Hong Kong, Hong Kong, as a Chair Professor of Information Engineering and a team leader.

Prof. Zukerman has served on various editorial boards such as Computer Networks, IEEE Communications Magazine, IEEE Journal of Selected Areas in Communications, IEEE/ACM Transactions on Networking, and the International Journal of Communication Systems.