Title	Minimizing equilibrium expected sojourn time via performance-based mixed threshold demand allocation in a multiple-server queueing environment
Author(s)	Choi, SM; Huang, X; Ching, WK
Citation	Journal of Industrial and Management Optimization, 2012, v. 8 n. 2, p. 299-323
Issued Date	2012
URL	http://hdl.handle.net/10722/146397
Rights	Creative Commons: Attribution 3.0 Hong Kong License

JOURNAL OF INDUSTRIAL AND
MANAGEMENT OPTIMIZATION
Volume **8**, Number **2**, May **2012**

doi:10.3934/jimo.2012.8.299

pp. **299–323**

MINIMIZING EQUILIBRIUM EXPECTED SOJOURN TIME VIA PERFORMANCE-BASED MIXED THRESHOLD DEMAND ALLOCATION IN A MULTIPLE-SERVER QUEUEING ENVIRONMENT

Sin-Man Choi*

Department of Industrial Engineering and Operations Research
University of California, Berkeley, USA

Ximin Huang

College of Management
Georgia Institute of Technology
800 West Peachtree Street NW
Atlanta, Georgia 30308-0520, USA

Wai-Ki Ching

Advanced Modeling and Applied Computing Laboratory
Department of Mathematics
The University of Hong Kong, Pokfulam Road, Hong Kong, China

(Communicated by Wuyi Yue)

Abstract. We study the optimal demand allocation policies to induce high service capacity and achieve minimum expected sojourn times in equilibrium in a queueing system with multiple strategic servers. We propose the mixed threshold allocation policy as an optimal state-dependent policy that induces optimal service capacity from strategic servers. Compensation to the server can be paid at customer allocation or upon job completion. Our study focuses on the use of a multiple-server mixed threshold allocation policy to replicate the demand of a given state-independent policy to achieve a symmetric equilibrium with lower expected sojourn time. The results indicate that, under both payment schemes, for any given multiple-server state-independent policy, there exists a multiple-server threshold policy that produces identical demand allocation and Nash equilibrium (if any). Moreover, the policy can be designed to minimize the expected sojourn time at a symmetric equilibrium. Furthermore, under the payment-at-allocation scheme, our results, combining with existing results on the optimality of the multiple-server linear allocation policy, show that the mixed threshold policy can achieve the maximum feasible service capacity and thus the minimum feasible equilibrium expected sojourn time. Hence, our results agree with previous two-server results and affirm that a trade-off between incentives and efficiency need not exist in the case of multiple servers.

2000 *Mathematics Subject Classification.* Primary: 60K25, 68M20; Secondary: 91A80.

Key words and phrases. Queueing system, demand allocation, threshold allocation, strategic server, principal-agent problem.

1. **Introduction.** The problem of finding the optimal control policy for a queueing system has been widely studied in the literature, see for instance [5, 9, 10]. Recent studies have focused on queueing systems with strategic servers [2, 7], particularly on deriving an optimal policy to induce high service capacities in a competitive environment [1, 3, 6]. In these systems, the servers decide their own service capacities and compete with each other for higher market share and profit. For example, this can be used to model service systems composed of independently-operating service providers, or supply chains with make-to-order suppliers who make their own operating decisions. It is then of interest what kind of policy for customer allocation and compensation can be used to induce high service capacities from the servers with minimum cost.

With performance-based demand allocation, the buyer decides the amount of demand to be allocated to a server based on the service capacities of all servers. This has been identified as a plausible option among different means to motivate faster service, as it requires little bargaining power of the buyer when compared with other motivators, like imposing late fees or offering a higher price per job [1]. The common-queue and separate-queue allocation studied in [6] are examples of such demand allocation policies. In the former, a common queue is maintained for two strategic servers and the demand allocated to a server is endogenously determined. In the latter, separate queues are maintained for each server and the demand is allocated to the two queues in proportions such that the expected waiting times in the two queues are the same. Extension of these two policies to the case of multiple servers have been studied in [3].

While both allocation policies discussed in [6] may be implemented without observing the servers' capacities, demand allocation policies that explicitly account for the servers' chosen service capacities may give the buyer a greater power to control servers' incentives and could be designed to induce the maximum feasible service capacity from servers. This has been shown in [1], where several state-dependent and state-independent allocation policies are studied and compared. Under the assumption that payment to the servers are made at customer allocation, it was concluded that the optimal policies in the two classes, respectively the linear allocation and the mixed threshold allocation, can induce the same maximum feasible service capacity, and thus there are no trade-off between incentives and efficiency.

The optimal state-independent policy in [1] has been extended to the case of multiple servers in [14], whereas the optimality argument of the two-server mixed threshold policy is significantly more difficult to generalize to the case of multiple servers and has not been considered in the literature.

The main aim of this paper is to generalize the mixed threshold policy proposed by [1] to the multiple-server mixed threshold policy, and study to what extent these policies can replicate the demand allocation of state-independent policies. We address both cases of the payment-at-allocation and payment-upon-completion scheme. Our result shows that, if we prohibit server overloading, then the multiple-server mixed threshold policies can replicate the demand allocation of any policy. Furthermore, under the payment-at-allocation scheme, the replication of the demand allocation of a state-independent policy with server overloading is feasible if we allow ourselves to include single-sourcing with some probability in the mixed threshold policy. Assuming that all servers are identical, in the Nash equilibrium,

the expected sojourn time with our mixed threshold policy is optimal with the equilibrium service capacities. In other words, our results concur with the two-server results of [1] and indicate that there is no trade-off between incentives and efficiency.

The rest of the paper is structured as follows. Section 2 summarizes related literature. Section 3 introduces the multiple-server demand allocation problem and review previous results obtained by [1] and [14]. In Section 4, we generalize the two-server threshold policy to an n-server threshold policy and find the set of allocated demand vectors that can be replicated using an n-server mixed threshold policy. In Section 5, we summarize the result and give a discussion on further research issues.

2. **Literature review.** Game theoretic analysis of equilibrium service capacities of two or more strategic servers has been considered in [2, 7]. Later studies focused on choosing an optimal policy to induce high service capacities in a competitive environment [1, 3, 6]. In these systems, the servers decide their own service capacities and compete with each other for market share and profit. Game theory [11] is used to model the interactions between strategic servers so as to find out the equilibrium service capacities and profits.

In [6], the common-queue and separate-queue allocation were compared with two strategic servers in a principal-agent framework (see [8]) to minimize cost needed to maintain expected sojourn time at or below a required level. The equilibrium service capacities chosen by the servers were found and compared. It was shown that the separate-queue allocation may give lower costs than the common-queue allocation, suggesting that there is a trade-off between efficiency and incentives. Extension of these two policies and the corresponding comparison to the case of multiple servers have been done in [3].

The study in [1] considered various demand allocation policies that explicitly account for the servers' chosen service capacities. The principal-agent problem studied is based on a two-server Markovian queueing system, where the buyer would like to induce a high service capacity from strategic servers through a performance-based allocation of demand and a compensation proportional to allocated demand. Two classes of allocation policies, namely *state-independent allocation policies* and *state-dependent allocation policies* are studied and compared. The model under each allocation policy is considered as a multiple-player strategic game and the Nash equilibrium, if any, is identified. Assuming payment to servers is made at customer allocation, they show that the linear allocation policy is an optimal state-independent policy and induces the maximum feasible service capacity from servers. They further argue that by randomizing between two-server threshold allocation policies, one could achieve an allocation identical to the linear allocation policy. Thus an optimal state-dependent policy that induces the maximum feasible service capacity can be obtained. However, we remark that in cases where the capacity of the primary server is lower than its allocated demand, the mixed threshold policy under the payment-at-allocation scheme implies that we allocate customers only to the primary server, which makes the system unstable even when the total service capacity is greater than the total demand rate, and at the same time we would be paying the server for more customers than it can actually serve. Similar optimality results have not been obtained in their study for the case where servers are paid upon job completion.

The optimality of the multiple-server linear allocation policy, again under the payment-at-allocation scheme, has been proved in [14]. However, the optimality

argument of the two-server mixed threshold policy, as proposed by [1], has not been considered in the case of multiple servers and the extension is much more mathematically complicated. The main difficulty lies in the complexity of the queueing system under an n-server threshold policy. With non-strategic servers, there have been studies on the optimality of threshold-type policies for heterogeneous server systems [10, 13]. However, the steady-state probabilities of the system cannot be obtained explicitly, and it is not straightforward to see how the demand allocation changes with the thresholds. Therefore, given a fixed state-independent policy, showing the existence of a mixed threshold policy that gives the same identical allocation is much more difficult in the case of multiple servers as the allocation vector (with respect to each chosen service capacity vector) is of higher dimension than in the two-server case. The study in this paper focuses on showing the set of allocation vectors that can be achieved by mixed threshold policies and establishing a similar result of optimality of the mixed threshold policies in the multiple-server case.

3. **The multiple-server demand allocation problem.** We consider a queueing system with n identical strategic servers. Customers arrive to the system according to a Poisson process with rate λ. Each server chooses its own service capacity μ_i and incurs a cost at the rate of $c(\mu_i)$, where $c(0) = 0$ and $c(.)$ is assumed to be strictly increasing and convex, i.e. $c'(.) > 0$ and $c''(.) \geq 0$. The time that Server i serves a customer is, independent of all other service times, exponentially distributed with mean rate μ_i. The buyer pays each server an amount of R for each customer it completes serving. The aim of the buyer is to select a demand allocation policy, through which the customers are assigned to the servers, that minimizes the expected sojourn time for a customer in the equilibrium. We assume

$$c\left(\frac{\lambda}{n}\right) < \frac{\lambda R}{n},$$

which is the necessary condition for the expected waiting times to be finite in an equilibrium where the n servers split the demand equally. Moreover, as a benchmark for comparison, we define the *maximum feasible service capacity* as $\bar{\mu}$ where $c(\bar{\mu}) = \lambda R/n$. In other words, the maximum feasible service capacity is the service capacity at which, when chosen by all servers, each server receives equal share of the demand and earns zero profit.

We consider two different payment schemes here. The first one is the *payment-at-allocation scheme*, where a server is paid when the customer is allocated to the server. The second one is the *payment-upon-completion scheme*, where a server is paid for a customer when it completes the service for the customer. When the service capacity of a server exceeds or equals its allocated demand rate, the two payment schemes essentially pay the same amount to the servers in the long run. However, if the service capacity μ_i of a server i is lower than its allocated demand rate λ_i, i.e. $\mu_i < \lambda_i$, the payment-at-allocation scheme will be paying the server at R times its allocated demand rate, i.e. $R\lambda_i$, while the payment-upon-completion scheme will be paying the server at R times its service rate, i.e. $R\mu_i$, which is lower than in the former case. It should be noted that, under the payment-at-completion scheme, since a server i is paid at most at the rate of $R\mu_i$ even if $\lambda_i > \mu_i$, we can only consider allocation policies with $\lambda_i \leq \mu_i$, i.e. we never need to overload a server, as overloading the server does not pay more to the server and thus do not help to give higher incentives to the server.

3.1. **State-independent and state-dependent allocation policies.** As proposed in [1], demand allocation policies can be divided into two classes, namely the *state-independent* and *state-dependent* allocation policies. The class of state-independent policies is characterized by the fact that under such policies, customer allocation is only based on the service capacities of the servers, but not the states of the servers (i.e., whether a server is busy or idle). Consequently, there is no difference between allocating a customer to a server immediately upon its arrival or not. We then assume that a First-In-First-Out (FIFO) queue is maintained for each server, and customers are immediately allocated to the queue of a server upon arrival. We further assume that the arrival of customers to each of these servers follows a Poisson process with rate λ_i. This assumption holds, for instance, when we allocate each customer to Server i with probability λ_i/λ. Examples of state-independent policies with multiple servers are the separate-queue allocation [3, 6], the linear allocation and the proportional allocation [1, 14]. In particular, [14] proved that the n-server linear allocation policy is optimal under the payment-at-allocation scheme. The other class of allocation policies, the state-dependent policies, are policies that allow customer allocation to depend on the state of the servers. Consequently, a customer may not be allocated to a server immediately upon arrival. The most common example is the common-queue allocation policy [3, 6], but here we will focus on a multiple-server generalization of the two-server mixed threshold policy discussed in [1].

3.2. **State-independent policies: A review of the multiple-server linear allocation policy.** Under the payment-at-allocation scheme, the two-server linear allocation policy proposed by [1] has been shown to be an optimal state-independent policy when appropriate parameters are chosen. Under the same payment scheme, the multiple-server linear allocation policy and its optimality have been studied by [14]. Under the n-server linear allocation policy, the allocation to Server i is given by

$$\lambda_i(\boldsymbol{\mu}) = \begin{cases} \theta\mu_i^{\rho} - \dfrac{1}{\hat{n}}\left(\theta\sum_{j=1}^{\hat{n}}\mu_j^{\rho} - \lambda\right) & i \le \hat{n}. \\ 0 & i > \hat{n}, \end{cases}$$

where the servers' capacities are sorted in a decreasing order, $\theta > 0$, $0 < \rho \le 1$ and $\hat{n} \le n$ is the largest integer such that $\lambda_{\hat{n}} \ge 0$ and $\mu_{\hat{n}} > 0$.

It should be noted that under this n-server linear allocation, the demand allocated to Server i can be greater than the service capacity chosen by Server i, i.e., $\lambda_i(\boldsymbol{\mu}) > \mu_i$ for some capacity vector $\boldsymbol{\mu}$. In other words, with the policy under the payment-at-allocation scheme, there are cases where a server is paid for more customers than it can actually serve, but such cases do not occur in the Nash equilibrium of the game.

Under the payment-at-allocation assumption (i.e. servers are paid for the job at allocation), [14] modelled the decision of the servers' capacities as an n-player strategic game and proved the existence and uniqueness of a Nash equilibrium in which the service capacity equals to the maximum feasible service capacity when the appropriate values of θ and ρ are chosen. Specifically, when the cost function $c(.)$ is strictly convex, [14] proved that a unique equilibrium exists with

$$\theta = \frac{nc'(\bar{\mu})}{R(n-1)} \quad \text{and} \quad \rho = 1$$

when $R > r_1 = b$. In the equilibrium $\mu_i = \bar{\mu}$ for all i and the expected service times are finite. For the case where the cost function $c(.)$ is linear, i.e., $c(\mu_i) = b\mu_i$ $(b > 0)$, [14] proved that a unique equilibrium exists with

$$\theta = \frac{n}{n-1}\left(\frac{2b\bar{\mu}^{1/2}}{R}\right) \quad \text{and} \quad \rho = \frac{1}{2}$$

when $R > r_1 = c(\lambda/n)/(\lambda/n)$. In the equilibrium $\mu_i = \bar{\mu}$ for all i and the expected service times are finite. We remark that similar results have not been obtained under the payment-upon-completion scheme.

3.3. **The state-dependent policies: A review of the two-server mixed threshold allocation policy.** Although the multiple-server linear allocation policy has been proved to induce the maximum feasible capacity from the servers, we are interested in investigating whether the same equilibrium service capacity can be induced by a state-dependent allocation policy. The main reason is that linear allocation, being a state-independent policy, does not allow for demand pooling, and so it is possible for a customer to be waiting for a busy server while another server is idle, even when the idle server could provide a lower expected sojourn time for the customers. A state-dependent allocation policy that induces the same level of service capacity could possibly give a lower expected sojourn time of the customers in the equilibrium when compared to the linear allocation policy. For the case of two servers, Cachon and Zhang ([1]) have shown that a mixed threshold allocation policy achieves this goal.

The two-server threshold allocation has first been studied as a control policy with non-strategic servers in the literature. In particular, it has been proved in [9] that the buyer's optimal allocation with two heterogeneous non-strategic servers is of threshold type. Under a two-server threshold allocation, a single queue is maintained for the two servers, but a job may not be allocated immediately to a server upon arrival, even if the server is idle. Job allocation is based on the designation of the primary (and secondary) server and a threshold parameter m. When a job arrives, it is allocated to the primary server if it is idle or has fewer than m jobs in queue and allocated to the secondary server only if it is idle, the primary server is busy, and has m jobs in queue. The advantage of a threshold allocation over a common-queue allocation is that, in some cases an idle server may be so slow that waiting for the another busy but faster server may yield a lower expected sojourn time. A numerical method for evaluating the system's performance under threshold allocation has been studied in [12]. It can be seen that, when different values of m are chosen, the demand allocated to the servers would be different. This allows us to parametrize the policy to create the appropriate level of competition that induces the desired service capacity in the equilibrium.

In Cachon and Zhang's study [1] of the two-server allocation problem with strategic servers, they proposed randomizing between threshold policies with different parameters to replicate the demand allocation of the linear allocation policy, so that the maximum feasible service capacity can be attained in the Nash equilibrium. Specifically, they argued that the buyer can allocate any portion of the buyer's demand to the primary server by varying which server is designated the primary server and randomizing between different threshold values m. They supported their claim by the fact that the primary server's allocated demand increases with m and when m is infinity, the primary server earns the buyer's entire demand.

The above argument is valid with many choices of service capacities, particularly when the service capacities are close enough to the equilibrium ones, which assures the existence of the desired Nash equilibrium. However, when the primary server's service capacity μ_1 is less than the total demand λ, with any finite values of m, the secondary server is allocated at least $\lambda - \mu_1$ of demand. The limit of the primary server's demand, as m goes to infinity, is μ_1. The only way to allocate more than μ_1 of the demand to the primary server is not to use the secondary server at all, i.e. setting $m = \infty$ and making $\lambda_1 = \lambda$, and to pay for the customers to the server at allocation instead of service completion. However, this will cause the system to be unstable, even in cases where $\mu_1 + \mu_2 > \lambda$, and is therefore undesirable.

If we prohibit server overloading (either by only allocating $\lambda_i \leq \mu_i$ to Server i or by not allowing the buyer to pay at customer allocation), then some allocated demand vectors cannot be replicated by a two-server mixed threshold policy. It is then important to know which allocated demand vectors can be replicated. We will extend the two-server mixed threshold allocation policies to the case of n servers and address the issue in the following sections.

4. Multiple-server threshold policies.

In this section, we will generalize the two-server threshold policy to an n-server threshold policy. We will assume that the buyer pays the server for a customer when the service is completed.

4.1. The n-server policy.

With n servers, where $n \geq 2$, it is natural to extend the two-server case by assigning the servers as the $1^{st}, 2^{nd}, \ldots, n^{th}$ servers and specifying $n-1$ threshold parameters. Similar control policies for non-strategic servers have been studied in [10]. In some of these studies the threshold parameters may depend on the state of the other servers. (More precisely, the threshold for the i^{th} server can depending on the state of the $(i+1)^{th}, \ldots, n^{th}$ servers). However, for simplicity and because randomization gives enough flexibility for parameterizing the policy, we shall assume that m_i is a constant in each policy in our study.

An n-server (pure) threshold allocation policy T is specified by an assignment of the Servers $1, 2, \ldots, n$ as the $1^{st}, 2^{nd}, \ldots, n^{th}$ servers and the thresholds $m_2, \ldots, m_n$ where each m_i is a nonnegative integer. We define $m_1 = 0$. A single queue is maintained in the system. When a customer arrives, it is assigned to Server 1 if it is idle. If Servers $1, 2, \ldots, i-1$ are all busy and the number of waiting customers (including the new arrival) is more than $m_1 + \ldots + m_i$, the customer is assigned to Server i. (Alternatively, we can also assign the first customer in the queue for Server i and let the new arrival wait in the queue.) Otherwise, it waits in the queue. When Server i completes service of a customer, if the number of waiting customers is more than $m_1 + \ldots + m_i$, then the first customer in the queue is assigned to Server i. If $m_i = \infty$ for some i, then no customer is allocated to Servers $i, i+1, \ldots, n$.

Given any service capacity vector

$$\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_n)$$

the demand allocated to the servers via the threshold policy T is

$$\boldsymbol{\lambda}^{(T)} = (\lambda_1^{(T)}, \lambda_2^{(T)}, \ldots, \lambda_n^{(T)})$$

where $\lambda_i^{(T)}$ is defined to be Server i's expected rate of receiving customers. In each state, if Server i is idle, its rate of receiving customers is the arrival rate of customers multiplied by the probability that an arriving customer is allocated to Server i. On the other hand, when Server i is busy, its rate of receiving customers is μ_i if there

are waiting customers that can be assigned to Server i upon service completion of the current customer, and is zero otherwise. Because the n-server policy is much more complicated, it is not straightforward to see what demand allocation can be achieved by the pure policy and by randomizing between some n-server threshold policies. In the following, we give some properties of an n-server pure threshold allocation policy and its allocated demand $\boldsymbol{\lambda}$.

Lemma 4.1. *Given n servers with service capacity vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ where $\sum_{i=1}^{n} \mu_i > \lambda$. Suppose Server i is designated as the i^{th} server. Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ be the allocated demand of an n-server pure threshold policy with the thresholds $m_2, \ldots, m_n$. We have the following results:*

(i) Let $k = \max\{i : 1 \le i \le n, m_j < \infty \quad \forall j = 1, 2, \ldots, i\}$. Then the system is stable if and only if $\sum_{i=1}^{k} \mu_i > \lambda$. When the system is stable, we have $\lambda = \sum_{i=1}^{n} \lambda_i$.

(ii) If $m_i = \infty$ for some i, then $\lambda_j = 0$ for all $j = i, i+1, \ldots, n$.

(iii) Suppose we fix $i = 2, \ldots, n$ and finite values of m_j for $j = 1, \ldots, i-1$. Then for any $\epsilon > 0$, there exists m_i^* such that for any $m_{i+1}, \ldots, m_n$ and $m_i > m_i^*$ we have

$$\min\left(\sum_{j=1}^{i-1} \mu_j, \lambda\right) \ge \sum_{j=1}^{i-1} \lambda_j > \min\left(\sum_{j=1}^{i-1} \mu_j, \lambda\right) - \epsilon.$$

We have seen that the demand allocation to the servers can be varied by adjusting the thresholds of a policy. However, because the thresholds only take integral values, the demand allocation is limited to a countable set of points. To enable us to select from a wider range of demand allocation, we introduce the n-server mixed threshold policy, which randomizes between a number of pure threshold policies.

Definition 4.2. An n-server mixed threshold allocation policy τ is specified by an integer $k \ge 1$, real numbers $\alpha_1, \ldots, \alpha_k$ such that $\sum_{i=1}^{k} \alpha_i = 1$ and k n-server threshold policies $T_1, \ldots, T_k$. When the mixed threshold allocation policy is used, each of the threshold policy T_i is used with probability α_i. The demand allocated via the mixed threshold policy τ is then denoted by $\boldsymbol{\lambda}^{(\tau)} = (\lambda_1^{(\tau)}, \lambda_2^{(\tau)}, \ldots, \lambda_n^{(\tau)})$ and given by

$$\lambda_j^{(\tau)} = \sum_{i=1}^{k} \alpha_i \lambda_j^{(T_i)}$$

for any server $j = 1, 2, \ldots, n$.

It is clear that the set of demand vectors that can be allocated by a pure threshold policy when $\sum_{i=1}^{n} \mu_i > \lambda$ is contained in the set

$$S_{\boldsymbol{\mu}} = \left\{ \boldsymbol{\lambda}^t : 0 \le \lambda_i^t \le \min(\mu_i, \lambda) \quad \text{and} \quad \sum_{i=1}^{n} \lambda_i^t = \lambda \right\}.$$

Since $S_{\boldsymbol{\mu}}$ is a convex set, it follows immediately that the set of demand vectors that can be allocated by a mixed threshold policy is also contained in $S_{\boldsymbol{\mu}}$. In the following, we explore which allocation vectors in $S_{\boldsymbol{\mu}}$ can be achieved by some mixed threshold policy given a fixed service capacity vector $\boldsymbol{\mu}$ such that $\sum_{i=1}^{n} \mu_i > \lambda$. Unless otherwise specified, in the following we shall assume such a fixed service capacity vector.

Suppose we have a target demand allocation vector $\boldsymbol{\lambda}^t$ such that $\sum_{i=1}^n \lambda_i^t = \lambda$. We say that an allocation policy τ with demand allocation $\boldsymbol{\lambda}^{(\tau)}$ is $\boldsymbol{\lambda}^t$-*dominated in the order* $(i_1, i_2, \ldots, i_n)$ if

$$\sum_{j=l}^{n} \lambda_{i_j}^{(\tau)} \leq \sum_{j=l}^{n} \lambda_{i_j}^t \quad \text{for all } l = 2, \ldots, n,$$

where $i_1, i_2, \ldots, i_n \in \{1, 2, \ldots, n\}$ and are distinct.

Also note that the above condition implies that $\lambda_{i_1}^{(\tau)} \geq \lambda_{i_1}^t$ since

$$\sum_{j=1}^{n} \lambda_{i_j}^{(\tau)} = \sum_{j=1}^{n} \lambda_{i_j}^t = \lambda.$$

Lemma 4.3. *Suppose we have n servers with service capacity vector* $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ *such that*

$$\sum_{i=1}^{n} \mu_i > \lambda,$$

and an allocation vector

$$\boldsymbol{\lambda}^t = (\lambda_1^t, \lambda_2^t, \ldots, \lambda_n^t)$$

such that

$$\sum_{i=1}^{n} \lambda_i^t = \lambda.$$

If $\lambda_j^t < \min(\mu_j, \lambda)$ *for all* $j = 1, 2, \ldots, n$, *then there exists an n-server (pure) threshold policy that is* $\boldsymbol{\lambda}^t$*-dominated in the order* $(1, 2, \ldots, n)$.

The pure threshold policies in Lemma 4.3 will be used in the following to compose a mixed threshold policy that gives the our target demand allocation. To illustrate the idea of we have obtained in the lemma, note that we can represent an allocated demand in a diagram as in Figure 1 by showing each $\sum_{j=k}^{n} \lambda_{i_j}$ for $k = 1, 2, \ldots, n$. Then a $\boldsymbol{\lambda}^t$-dominated policy in the order $(i_1, i_2, \ldots, i_n)$, with demand allocation $\boldsymbol{\lambda}^{(\tau)}$ have each of these quantities less than or equal to that of $\boldsymbol{\lambda}^t$, as shown in Figure 2.

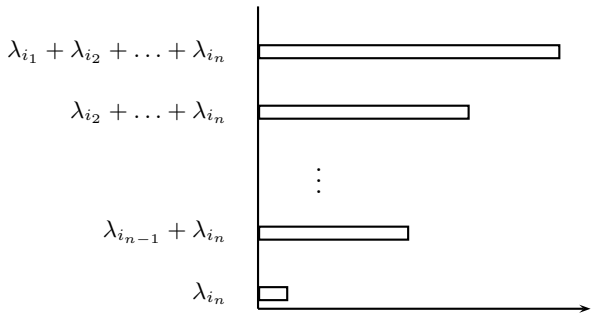

FIGURE 1. Expressing an allocation vector $\boldsymbol{\lambda}$ as a diagram of the $\sum_{j=k}^{n} \lambda_{i_j}$ given $(i_1, i_2, \ldots, i_n)$, with $k = 1, 2, \ldots, n$.

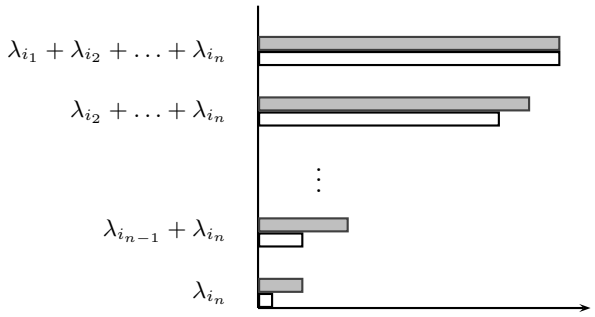

FIGURE 2. Illustrating the idea of an allocation policy τ that is λ^t-dominated in the order $(i_1, i_2, \ldots, i_n)$. The gray bars represent the target allocation $\boldsymbol{\lambda}^t$ and the white bars represent the demand allocation $\boldsymbol{\lambda}^{(\tau)}$.

The following two lemmas are used to show that we can construct mixed threshold policies with some nice properties for the construction of the one giving the target demand allocation.

To facilitate our discussion, we say that an allocation policy τ with demand allocation $\boldsymbol{\lambda}^{(\tau)}$ is *$\boldsymbol{\lambda}^t$-dominated and m-smaller in the order* $(i_1, i_2, \ldots, i_n)$ if the policy is $\boldsymbol{\lambda}^t$-dominated in the order $(i_1, i_2, \ldots, i_n)$ and

$$\lambda_{i_j}^{(\tau)} \leq \lambda_{i_j}^t$$

for all $j = m, m+1, \ldots, n$, where m is an integer such that $2 \leq m \leq n$ and $i_1, i_2, \ldots, i_n \in \{1, 2, \ldots, n\}$ are distinct.

Note that in the above definition, the property is equivalent up to any permutation of $i_m, i_{m+1}, \ldots, i_n$. We also note that any policy $\boldsymbol{\lambda}^t$-dominated in the order $(i_1, i_2, \ldots, i_n)$ is $\boldsymbol{\lambda}^t$-dominated and n-smaller in the order $(i_1, i_2, \ldots, i_n)$. Therefore, from Lemma 4.3 we have obtained a set of $\boldsymbol{\lambda}^t$ and n-smaller policies in different orders $(i_1, i_2, \ldots, i_n)$. The idea of an $\boldsymbol{\lambda}^t$-dominated and m-smaller policy in the order $(i_1, i_2, \ldots, i_n)$ is illustrated in Figure 3.

In the following lemma, we show that, given policies that are $\boldsymbol{\lambda}^t$-dominated and m-smaller in all possible orders $(j_1, j_2, \ldots, j_n)$, we can obtain $\boldsymbol{\lambda}^t$-dominated and $(m-1)$-smaller policies in any order $(i_1, i_2, \ldots, i_n)$.

Lemma 4.4. *For fixed $\boldsymbol{\mu}$ and $m \in \{3, \ldots, n\}$, suppose for each $k = m-1, m, \ldots, n$, we have a mixed threshold policy $\tau_{m,k}$ that is $\boldsymbol{\lambda}^t$-dominated and m-smaller in the order $(1, 2, \ldots, m-2, k, m-1, m, \ldots, k-1, k+1, \ldots, n)$. Then there exists a mixed threshold policy τ_{m-1} that is $\boldsymbol{\lambda}^t$-dominated and $(m-1)$-smaller in the order $(1, 2, \ldots, n)$.*

Considering different orders of $(i_1, i_2, \ldots, i_n)$ and using Lemma 4.4 for induction from $m = n$ down to $m = 2$, we can obtain $\boldsymbol{\lambda}^t$-dominated and 2-smaller policies in any order $(i_1, i_2, \ldots, i_n)$. It can be seen easily that for such a policy, we have $\lambda_{i_1}^{(\tau)} \geq \lambda_{i_1}^t$, as illustrated in Figures 4 and 5. Such a policy can be restated in a

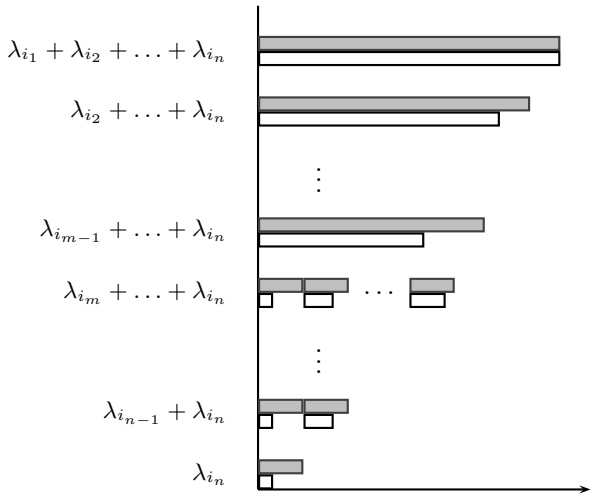

FIGURE 3. The gray bars represent the target allocation $\boldsymbol{\lambda}^t$ and the white bars represent the demand allocation $\boldsymbol{\lambda}^{(\tau)}$. In additional to the requirement of a λ^t-dominated policy, this policy needs to satisfy $\lambda_{i_j} \leq \lambda^t_{i_j}$ for $j = m, m+1, \ldots, n$, as illustrated in the diagram by breaking up the sum $\lambda_{i_m} + \lambda_{i_{m+1}} + \ldots + \lambda_{i_n}$ into small blocks of λ_{i_m}, $\lambda_{i_{m+1}}$, $\ldots$, λ_{i_n} and having each of the small blocks corresponding to $\boldsymbol{\lambda}^{(\tau)}$ smaller than or equal to those corresponding to $\boldsymbol{\lambda}^t$.

simpler way as requiring

$$\begin{cases} \lambda_k \geq \lambda^t_k \\ \lambda_j \leq \lambda^t_j \quad \forall j \neq k. \end{cases}$$

The following lemma states the existence of such a policy.

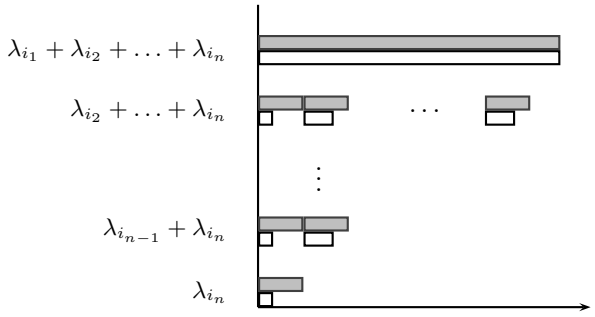

FIGURE 4. A λ^t-dominated 2-smaller allocation policy in the order $(i_1, i_2, \ldots, i_n)$. It then follows immediately that $\lambda_{i_1} \geq \lambda^t_{i_1}$.

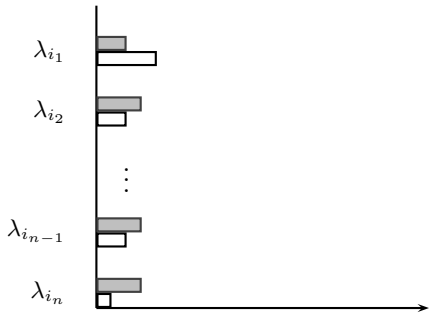

FIGURE 5. A λ^t-dominated 2-smaller allocation policy in the order $(i_1, i_2, \ldots, i_n)$ is equivalent to a policy that we are interested in in Lemma 4.5 with $k = i_1$.

Lemma 4.5. *Suppose we have n servers with service capacity vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ such that*

$$\sum_{i=1}^{n} \mu_i > \lambda,$$

and an allocation vector

$$\boldsymbol{\lambda}^t = (\lambda_1^t, \lambda_2^t, \ldots, \lambda_n^t)$$

such that

$$\sum_{i=1}^{n} \lambda_i^t = \lambda.$$

If $\lambda_i^t < \min(\mu_i, \lambda)$ for all $i = 1, 2, \ldots, n$, then for any fixed k, there exists an n-server mixed threshold policy with allocated demand $\boldsymbol{\lambda}$ such that

$$\begin{cases} \lambda_k \geq \lambda_k^t \\ \lambda_j \leq \lambda_j^t \quad \forall j \neq k. \end{cases} \tag{1}$$

Lemma 4.5 provides us with a set of policies that are close enough to the target demand allocation in the sense that only one of the servers could possibly receive more demand than the targeted one, with the other servers all receiving an equal or less amount of demand compared to the targeted demand allocation. Using these policies, we can find a mixed threshold policy that gives exactly our target demand allocation. This is shown in the following proposition.

Proposition 1. *Suppose we have some fixed service capacity vector*

$$\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) \quad \textit{with} \quad \sum_{i=1}^{n} \mu_i > \lambda$$

and target allocation vector

$$\boldsymbol{\lambda}^t = (\lambda_1^t, \ldots, \lambda_n^t) \quad \textit{with} \quad \sum_{i=1}^{n} \lambda_i^t = \lambda$$

and

$$0 < \lambda_i^t < \min(\mu_i, \lambda) \quad \textit{for} \quad i = 1, 2, \ldots, n.$$

Then there exists a mixed threshold allocation policy with allocated demand $\boldsymbol{\lambda}$ *such that* $\lambda_i = \lambda_i^t$ *for all* $i = 1, 2, \ldots, n$.

We have shown that for any $\boldsymbol{\mu}$ with $\sum_{i=1}^n \mu_i > \lambda$, any demand allocation vector set in the interior of the set $S_{\boldsymbol{\mu}}$ is the allocated demand of some mixed threshold policy. Moreover, if $\lambda_i^t = 0$ for some i, the demand allocation can be achieved by removing all servers i with $\lambda_i^t = 0$ and considering a mixed threshold policy for the reduced number of servers. On the other hand, if $\lambda_i^t = \lambda \leq \mu_i$ for some i, then it can be achieved by assigning all customers to Server i. Therefore, the set

$$S'_{\boldsymbol{\mu}} = \left\{ \boldsymbol{\lambda}^t : 0 \leq \lambda_i^t < \mu_i, \lambda_i^t \leq \lambda \quad \text{and} \quad \sum_{i=1}^n \lambda_i^t = \lambda \right\}$$

is achievable. It remains to investigate whether we could find a mixed threshold policy that achieves $\boldsymbol{\lambda}^t$ where $\sum_{i=1}^n \lambda_i^t = \lambda$ and $\lambda_i^t = \mu_i < \lambda$ for some i. However, this is impossible, because whenever the system is stable (i.e. when the sum of service capacities of all servers with finite thresholds is greater than the total demand rate), the demand allocated to Server i, λ_i, must be strictly less than μ_i (as the proportion of time of the system having no customers is positive). Therefore, in order to have λ_i equal μ_i, the system must be unstable. This implies that $\sum_{i=1}^n \lambda_i \leq \sum_{i=1}^n \mu_i < \lambda$. The remaining demand $\lambda - \sum_{i=1}^n \mu_i$ then cannot be allocated. Thus it is impossible to allocate to Server i exactly a demand of μ_i using a mixed threshold allocation if all demand has to be allocated.

Nevertheless, the problem can be solved in two ways. First, note that λ_i approaches μ_i in the limit as the threshold m_{i+1} goes to infinity if $\sum_{j=1}^i \mu_j < \lambda$. For any $\epsilon > 0$, we can find a value of the threshold such that $|\mu_i - \lambda_i| < \epsilon$. Alternatively, we can use a state-independent allocation and assign a proportion of μ_i/λ of the arrivals to Server i for such cases.

4.2. **Analysis on unstable queueing system.** In the above sections, we have mainly focused on the case where the total service capacities exceed the total demand rate and so all demand are allocated, i.e. $\sum_{i=1}^n \lambda_i = \lambda$. If the sum of the chosen service capacities are less than the total demand rate, $\mu_1 + \ldots + \mu_n \leq \lambda$, the queueing system is not stable regardless of the values of $m_2, \ldots, m_n$. Although it is natural to utilize the servers as much as possible when the system is not stable, the alternative of allocating strictly less than the service capacity of a server to it may be useful with strategic servers to induce the servers to switch to higher service capacities in the long run, since we are mainly concerned with the equilibrium service capacities. Technically, designing an allocation policy that assigns $\lambda_i < \mu_i$ to Server i in these cases may help to avoid the existence of an undesirable Nash equilibrium where the queueing system is unstable.

In [1], under the state-independent linear allocation, a server may be given an allocated demand more than, equal to or less than its service capacity when the queueing system is not stable. We remark that with threshold allocation, when the system is unstable, it remains impossible to allocate to a server a demand level that is higher than its capacity, because a customer is only assigned to the server when it is idle. Thus any demand allocation where $\lambda_i > \mu_i$ is not possible. As a pure strategy, the buyer can choose to allocate a demand of zero or μ_i to Server i by setting m_i to be infinite or finite, respectively. Under the condition that

$$\mu_1 + \mu_2 + \ldots + \mu_n \leq \lambda$$

the threshold m_i does not affect the allocated demand of other servers. Consequently, we can randomize between the values of m_i and obtain any allocated demand λ_i such that $0 \leq \lambda_i \leq \mu_i$. Therefore we conclude that the set of feasible allocation when

$$\mu_1 + \mu_2 + \ldots + \mu_n < \lambda$$

is the set of allocation vectors satisfying $0 \leq \lambda_i \leq \mu_i$.

4.3. **Efficient mixed threshold policies.** We have shown that the set of demand allocation vectors

$$S_{\boldsymbol{\mu}} = \begin{cases} \{\boldsymbol{\lambda}^t : 0 \leq \lambda_i^t \leq \min(\mu_i, \lambda) \quad \text{and} \quad \sum_{i=1}^n \lambda_i^t = \lambda\} & \text{if } \sum_{i=1}^n \mu_i > \lambda \\ \{\boldsymbol{\lambda}^t : 0 \leq \lambda_i^t \leq \min(\mu_i, \lambda)\} & \text{if } \sum_{i=1}^n \mu_i \leq \lambda \end{cases}$$

can be replicated by mixed threshold policies. However, it is not yet certain whether such policies perform better than state-independent policies. It has been shown that for servers with different service capacities, the optimal policy that gives the lowest expected sojourn time is of threshold type [10], but some thresholds may depend on the states of the other servers, and the mixed threshold policy we have may not give the lowest expected sojourn time with respect to the chosen service capacities. Indeed, in order to design an allocation policy that induces the server to choose the maximum feasible capacity and thus minimizes equilibrium expected sojourn times, efficiency must be given up with some out-of-equilibrium choices of service capacities. Hence, our aim in this section is to find out whether the mixed threshold policy can give a lower expected sojourn time *in equilibrium* when compared to the state-independent policies.

As we deal with identical servers, we expect a symmetric equilibrium, where all servers choose the same service capacity and receive equal share of the demand. It is desirable that our mixed threshold policy gives the minimal expected sojourn time in this case, which will be shown in the following two propositions.

Proposition 2. *When $\mu_1 = \mu_2 = \ldots = \mu_n = \mu_c > \lambda/n$ one can randomize among some threshold allocation policies with zero thresholds to obtain the demand allocation $\lambda_1 = \lambda_2 = \ldots = \lambda_n = \lambda/n$.*

Proposition 3. *In an n-server queueing system, given that $\mu_1 = \mu_2 = \ldots = \mu_n = \mu_c$, any n-server threshold allocation with all thresholds being zero gives the same expected sojourn time as an n-server common-queue system where each server has service capacity μ_c.*

Finally, note that because any pure threshold policy with all threshold being zeros has an expected sojourn time identical to that of the n-server common queue, any mixed policy that is composed of such pure threshold policies would have the same expected sojourn time too. Combining with Proposition 2, we have shown that the mixed threshold policy used to replicate a state-independent policy could be designed to have minimal sojourn times in a symmetric equilibrium, which is better than the state-independent policy. Thus the use of a mixed threshold policy could indeed help to improve efficiency and lower the expected sojourn time in the equilibrium.

4.4. **Interpretations and discussions.** We have shown that for any fixed service capacity vector $\boldsymbol{\mu}$ and any target demand allocation vector $\boldsymbol{\lambda}$ such that $0 \leq \lambda_i \leq \lambda$, $\lambda_i < \mu_i$ and $\sum_{i=1}^n \lambda_i = \lambda$ (if $\sum_{i=1}^n \mu_i > \lambda$) it is possible to choose a mixed threshold policy that gives the demand allocation $\boldsymbol{\lambda}$. For the case where $\lambda_i = \mu_i$, it can

be catered for by using a state-independent allocation for that case. Applying the respective policy for each service capacity vector $\boldsymbol{\mu}$ when it is observed, we have a state-dependent policy that gives the demand allocation $\boldsymbol{\lambda}(\boldsymbol{\mu})$. In other words, for any state-independent policy P_1 with demand allocation $\boldsymbol{\lambda}$ such that $0 \leq \lambda_i \leq \min(\mu_i, \lambda)$, there exists a state-dependent policy that replicates the demand allocation of the policy P_1. Moreover, from the discussion in Section 3.3, we see that the expected sojourn time under the state-dependent policy is lower than that under policy P_1. We conclude that for any state-independent policy that does not overload the servers, i.e., $\lambda_i \leq \mu_i$, there exists a state-dependent policy that replicates the same demand allocation, thus giving the same Nash equilibrium but a lower expected sojourn time in the equilibrium.

The arguments above apply to both the payment-at-allocation and payment-upon-completion cases. We note that server overloading under the payment-upon-completion scheme is not meaningful as the server only receives payment for the customers that it finishes serving, the same is not true under the payment-at-allocation scheme. In this case, server overloading needs to be considered as that would result in a higher compensation rate to the server. In the following, we assume the payment-at-allocation scheme and discuss the case where server overloading is permitted. If we relax the conditions $\lambda_i \leq \mu_i$ for $i = 1, 2, \ldots, n$ and use the payment-at-allocation scheme, as we have seen in [14], there could be a state-independent allocation that gives an equilibrium with the maximum feasible service capacity with $\lambda_i > \mu_i$ in some cases. To replicate the allocation of such policies, we must allow servers to be overloaded and use the payment-at-allocation scheme.

If we assign all the demand to one server, say Server i, and pay the server at customer allocation, then the demand allocated to Server i and its rate of revenue, would be λ and λR respectively. Randomizing this allocation with other mixed threshold policies, it is possible to achieve any target demand allocation $\boldsymbol{\lambda}$ such that $0 \leq \lambda_i \leq \lambda$ and $\sum_{i=1}^{n} \lambda_i = \lambda$. This can be easily proved by noting that allocating all demand to Server i gives the demand allocation

$$\boldsymbol{\lambda} = \lambda \mathbf{e}_i = (0, \ldots, 0, \underbrace{\lambda}_{i^{th}\text{entry}}, 0, \ldots, 0),$$

for $i = 1, \ldots, n$, and any target demand allocation can be expressed as a convex combinations of these vectors. However, such an allocation results in infinite waiting times and should be avoided as far as possible. Thus, for demand vectors such that $0 \leq \lambda_i \leq \min(\mu_i, \lambda)$, we can apply the results in previous subsections and use a mixed threshold policy that comprises of only threshold policies with finite waiting times to replicate the demand allocation. In particular, at equilibrium we only need to randomize between threshold policies with zero thresholds, so that the expected sojourn time is equal to that in an n-server common queue system with the maximum feasible service capacity chosen.

4.5. **Comparison of expected sojourn times.** In previous subsections, we have shown that under the payment-at-allocation scheme, a mixed threshold allocation policy, if allowed to overload servers, can attain the same equilibrium service capacity as the linear allocation policy while giving a lower (and in fact minimum) expected sojourn time. However, the implementation of a mixed threshold allocation policy is complicated and may be costly, especially when the number of servers is large. In this subsection, we investigate how the ratio of the expected sojourn time of the two policies changes when the number of servers, n, becomes very large.

This would provide insight into whether it is worthwhile to implement the mixed threshold allocation policy when there is a higher implementation cost compared to the linear allocation policy.

Assume fixed $n \geq 2$ and R such that

$$c(\lambda/n) < \frac{\lambda R}{2}.$$

Let W_{si} and W_{sd} be the expected sojourn time in equilibrium under the optimal state-independent allocation and the corresponding replication by the threshold allocation policy, assuming both allocation yields a unique symmetric equilibrium.

By standard results of an $M/M/1$ queue, with demand λ/n allocated to each server, we have

$$W_{si} = \frac{1}{\bar{\mu}_n - \lambda/n} \tag{2}$$

By Proposition 3 and standard results of a $M/M/n$ system, we also have

$$W_{sd} = \frac{1}{\bar{\mu}_n}\left[\frac{\lambda}{n\bar{\mu}_n - \lambda}\left(\sum_{k=0}^{n-1}(k+1)!\begin{pmatrix} n-1 \\ k \end{pmatrix}\left(\frac{\bar{\mu}_n}{\lambda}\right)^k\right)^{-1} + 1\right]. \tag{3}$$

Combining Equations (2) and (3) we have

$$\frac{W_{sd}}{W_{si}} = \left[\frac{a}{n}\left(\sum_{k=0}^{n-1}(k+1)!\begin{pmatrix} n-1 \\ k \end{pmatrix}\left(\frac{1}{a}\right)^k\right)^{-1} + \frac{n-a}{n}\right] \tag{4}$$

where $a = \lambda/\bar{\mu}_n$.

We are interested in the limit of the ratio in (4) as n goes to infinity.

Proposition 4. *For a given value R such that $c(\lambda/2) < \lambda R/2$, if $c'(0) = 0$, then*

$$\lim_{n\to\infty} \frac{W_{sd}}{W_{si}} = 1.$$

Otherwise,

$$\lim_{n\to\infty} \frac{W_{sd}}{W_{si}} \leq 1 - \frac{c'(0)}{R}(1 - e^{-R/c'(0)}) < 1.$$

To understand the implications of the proposition, we first note that the maximum feasible service capacity, $\bar{\mu}$, becomes arbitrarily small as n goes to infinity. Then $c'(0)$ represents the marginal cost of increasing capacity at such a level. From Proposition 4, we know that if this marginal cost is zero, i.e., if $c'(0) = 0$, the advantage of using the optimal state-dependent policy over the optimal state-independent policy vanishes as the number of servers approaches infinity. On the other hand, if $c'(0) > 0$, the ratio of the expected sojourn time under the optimal state-dependent policy to that under the optimal state-independent policy approaches a limit that is strictly below one when the number of servers approaches infinity.

5. **Concluding remarks.** In this paper, we have extended the two-server mixed threshold allocation policy proposed by [1] to the case of n servers. For any state-independent policy that prohibits server overloading, we have shown that it is possible to replicate the allocated demand by a mixed threshold policy. We consider two payment schemes: the payment-at-allocation scheme and the payment-upon-completion scheme. Under the payment-at-allocation scheme, where server-overloading is possible, we have shown that a mixed threshold allocation policy can

replicate the allocated demand if we include a single-sourcing strategy in the mixed policy and allow payment at customer allocation. For the payment-upon-completion scheme, although we do not know whether the mixed threshold policy can give the maximum feasible service capacity $\bar{\mu}$, our results do show that the mixed threshold policy can perform as well as any other state-independent or state-dependent policy in terms of the induced service capacity. For identical servers, the mixed threshold policy at the symmetric equilibrium can be composed of only threshold policies with zero thresholds. As a result, the policy yields the minimal expected sojourn time with the equilibrium service capacities.

Our results concur with existing two-server results that there are no trade-off between incentives and efficiency. Whether or not we allow server overloading, we can find a n-server mixed threshold policy that induces the same service capacity from the servers as any given state-independent policy. Moreover, in the symmetric equilibrium, the mixed threshold policy can always give a lower expected sojourn time.

Our extension of the mixed threshold policy to multiple servers is natural, but the proof that the n-server mixed threshold policy can replicate any other policy is significantly more difficult than its two-server counterpart. The technical hurdle lies in that, in the inductive process of constructing the desire mixed threshold policy, we need to match the demand of a server while keeping the previously matched ones the same. This concern was not present in the two-server case and made the proof much more complicated in both finding the appropriate component pure policies and in matching the target demand allocation.

Our results have been derived in a framework where servers are identical, i.e. they have the same cost function $c(\mu)$. Nevertheless, the results in Sections 4.1 – 4.2 are independent of the cost structure of the servers. Therefore, with asymmetric servers, it is also possible to replicate the demand allocation of any state-independent policy that prohibits server overloading by an n-server threshold allocation policy. However, because the Nash equilibrium, when exists, may not be symmetric, it has yet to be investigated whether a suitable n-server threshold allocation policy performs better than a state-independent policy in terms of achieving a lower expected sojourn time.

Our results are based on a Markovian queueing system. We believe that a similar analysis can be carried out in the cases with more general distributions of the inter-arrival times or service times. However, the actual computation of the n-server mixed threshold policy may be more complicated due to the difficulty in the computation of the allocated demand in an n-server threshold system.

In our model, it is assumed that the service capacities chosen by the servers can be observed by the buyer. However, in reality the buyer has to infer the service capacities from realized service times. In our study we have not considered how statistical errors may affect our results. Thus it is an interesting future research issue.

In Zhang's work [14], it has been shown that the multiple linear allocation can achieve the maximum feasible service capacity $\bar{\mu}$ in the Nash equilibrium, under the payment-at-allocation scheme with server overloading permitted. It then follows from our results that, if we also allow server overloading and payment at allocation, there exists an n-server mixed threshold allocation policy that achieves the maximum feasible service capacity with the minimum feasible expected sojourn time at equilibrium. However, as mentioned in earlier sections, server overloading

and payment at allocation cause unnecessary infinite-waiting times at some out-of-equilibrium plays, and may be undesirable. It still remains to be investigated whether there exists a state-independent policy without server overloading that achieves the maximum feasible service capacity under the payment-upon-completion scheme. Nevertheless, our results still show that the mixed threshold policy can perform as well as any other state-independent or state-dependent policy in terms of the induced service capacities. Thus, if a policy that induces the maximum feasible service capacity exists, then our results would imply that an optimal mixed threshold allocation policy without server overloading (i.e. $\lambda_i \leq \mu_i$ in all allocation) also exists.

Our work has proved the existence of an n-server threshold policy that replicates any given state-independent policy that prohibits server overloading. For any fixed service capacity and given target demand allocation, it is desirable to find a mixed policy that not only gives the minimum expected sojourn time, but also randomizes between minimum number of policies. Finding an efficient way to identify such a mixed policy may be a direction for future research. Since the n-server mixed threshold policy involves a set of parameters for each service capacity vectors, another future research issue may be to investigate whether there could be simpler state-dependent policy with fewer parameters that gives the same incentives and efficiency.

6. **Proof of Lemmas and Propositions.**

6.1. **Proof of Lemma 4.1.**

Proof. For statement (i), When $\mu_1 + \mu_2 + \ldots + \mu_k < \lambda$, in any state of the queueing system, the arrival rate of the queueing system is λ, which is greater than the total service rate, which is at most $\mu_1 + \mu_2 + \ldots + \mu_k$. Considering that a birth-and-death process with birth rate λ and death rate $\mu_1 + \mu_2 + \ldots + \mu_k$ is unstable, we see that the long-run number of customers in the queueing system will be infinite with probability 1. On the other hand, when the total number of customers in the system is more than $k + m_2 + m_3 + \ldots + m_k$, Servers $1, 2, \ldots, k$ will always be in use and so the process of the additional number of customers behaves as a birth-and-death process with birth rate λ and death rate $\mu_1 + \mu_2 + \ldots + \mu_k$. Therefore, $\mu_1 + \mu_2 + \ldots + \mu_k > \lambda$ is sufficient for the system to be stable. When the system is stable, all customers are served with probability 1. Thus $\lambda = \sum_{i=1}^{n} \lambda_i$.

Statement (ii) is straightforward since, by definition, no customer is allocated to join the servers $i, i+1, \ldots, n$ as $m_i = \infty$.

For Statement (iii), the fact that

$$\sum_{j=1}^{i-1} \lambda_j \leq \min\left(\sum_{j=1}^{i-1} \mu_j, \lambda\right)$$

is straightforward. For the other side, we consider two cases.

Case I: If $\sum_{j=1}^{i-1} \mu_j > \lambda$, we compare the system with the delay system (subject to the same inter-arrival times and service times) where the $i^{th}, (i+1)^{th}, \ldots, n^{th}$

servers are not used. This is equivalent to the case where $m_i = \infty$. Let Y be the number of waiting customers in the system. Then we have

$$\lim_{k\to\infty} P(Y \geq k) = 0$$

since the system is stable given $\sum_{j=1}^{i-1} \mu_j > \lambda$.

Now consider the original threshold system with threshold m_i. For fixed i, let Y_{m_i} be the total number of customers waiting in the first $m_2 + m_3 + \ldots + m_i$ positions of the queue. Since in this system some customers are allocated to Servers $i, i+1, \ldots, n$ while no customer is lost to other servers in the previous system, we have $Y_{m_i} \leq Y$ for any outcome of the inter-arrival times and service times. Thus the event $Y_{m_i} \geq m_2 + m_3 + \ldots + m_i$ implies $Y \geq m_2 + m_3 + \ldots + m_i$. This results in

$$P\left(Y_{m_i} \geq m_2 + m_3 + \ldots + m_i\right) \leq P\left(Y \geq m_2 + m_3 + \ldots + m_i\right).$$

But then the left-hand-side is nonnegative and the right-hand side approaches zero as $m_i \to \infty$. Clearly the right-hand-side is independent of $m_{i+1}, \ldots, m_n$. For any $\epsilon > 0$, we have m_i^* such that

$$P\left(Y \geq m_2 + m_3 + \ldots + m_i\right) < \frac{\epsilon}{\lambda}$$

for any $m_i > m_i^*$. Finally by $\sum_{j=i}^{n} \lambda_j \leq \lambda P(Y \geq m_2 + m_3 + \ldots + m_i)$ we have

$$\sum_{j=1}^{i-1} \lambda_j = \lambda - \sum_{j=i}^{n} \lambda_j > \lambda - \epsilon.$$

Case II: If $\sum_{j=1}^{i-1} \mu_j \leq \lambda$, once again, consider Y_{m_i}, the number of customers waiting in first $m_2 + m_3 + \ldots + m_i$ positions of the queue under the given threshold policy, and $Y_{m_i}^l$, the number of waiting customers in a loss system that consists of the first i servers with thresholds $m_1, m_2, \ldots, m_{i-1}$ and queue length $m_2 + m_3 + \ldots + m_i$, with both systems subject to the same inter-arrival times and service times. Then we have $Y_{m_i} \geq Y_{m_i}^l$ for any outcome of the inter-arrival times and service times. Thus we have

$$P\left(Y_{m_i} \geq m_2 + m_3 + \ldots + m_{i-1} + 1\right) \geq P\left(Y_{m_i}^l \geq m_2 + m_3 + \ldots + m_{i-1} + 1\right).$$

Note that the left-hand-side is at most 1, while the right-hand-side is independent of $m_{i+1}, \ldots, m_n$ and converges to 1 as $m_i \to \infty$ because it approaches the case of a delay system, which is unstable given $\sum_{j=1}^{i-1} \mu_j \leq \lambda$.

Then for any $\epsilon > 0$, we have m_i^* such that for any $m_i > m_i^*$, we have

$$P(Y_{m_i}^l \geq i + m_2 + \ldots + m_{i-1}) > 1 - \frac{\epsilon}{\sum_{j=1}^{i-1} \mu_j}.$$

Thus we have

$$\sum_{j=1}^{i-1} \lambda_j \geq \sum_{j=1}^{i-1} \mu_j P(Y_{m_i} \geq i + m_2 + \ldots + m_{i-1}) > \sum_{j=1}^{i-1} \mu_j \left(1 - \frac{\epsilon}{\sum_{j=1}^{i-1} \mu_j}\right) = \sum_{j=1}^{i-1} \mu_j - \epsilon.$$

□

6.2. **Proof of Lemma 4.3.**

Proof. By statement (iii) of Lemma 4.1, we can find m_2^* such that

$$\sum_{j=2}^{n} \lambda_j \leq \sum_{j=2}^{n} \lambda_j^t \quad \text{for} \quad m_2 = m_2^*.$$

Assuming that we can find $m_2^*, \ldots, m_k^*$, $k < n$ such that

$$\sum_{j=l}^{n} \lambda_j \leq \sum_{j=l}^{n} \lambda_j^t \quad \text{for all } l = 2, \ldots, k$$

when $m_2 = m_2^*, m_3 = m_3^*, \ldots, m_k = m_k^*$. Then again by the statement (iii) of Lemma 4.1, we can also find m_{k+1}^* such that

$$\sum_{j=l}^{n} \lambda_j \leq \sum_{j=l}^{n} \lambda_j^t \quad \text{for all } l = 2, \ldots, k, k+1$$

when $m_2 = m_2^*, m_3 = m_3^*, \ldots, m_k = m_k^*, m_{k+1} = m_{k+1}^*$. By induction we can find $m_2^*, \ldots, m_n^*$ such that

$$\sum_{j=l}^{n} \lambda_j \leq \sum_{j=l}^{n} \lambda_j^t \quad \text{for all } l = 2, \ldots, n$$

when $m_2 = m_2^*, m_3 = m_3^*, \ldots, m_n = m_n^*$. □

6.3. **Proof of Lemma 4.4.**

Proof. Let Q_0 be the statement that there exists a mixed threshold policy τ_{m-1} that is $\boldsymbol{\lambda}^t$-dominated and $(m-1)$-smaller in the order $(1, 2, \ldots, n)$.

Moreover, for $q \geq m-1$, let Q_q be the statement that, for any $l = q, q+1, \ldots, n$, there exists a mixed threshold policy τ_l that is λ^t-dominated and m-smaller in the order $(1, 2, \ldots, m-2, l, m-1, m, \ldots, l-1, l+1, \ldots, n)$ with demand allocation $\boldsymbol{\lambda}^{(l)}$ such that

$$\lambda_i^{(l)} = \lambda_i^t \quad \text{for} \quad i = m-1, \ldots, q-1. \tag{5}$$

Then we see that Q_{m-1} is true by hypothesis, since the index set of i in (5) is empty when $q = m-1$. We want to show that if Q_q is true, then either Q_0 is true or Q_{q+1} is true.

Now suppose Q_q is true. If among any of the policies τ_l we have $\lambda_l^{(l)} \leq \lambda_l^t$, then it is $\boldsymbol{\lambda}^t$-dominated and $(m-1)$-smaller in the order $1, 2, \ldots, n$ and Q_0 is true. Otherwise, we have $\lambda_l^{(l)} > \lambda_l^t$ for all $l = q, q+1, \ldots, n$. However, as policy τ_j is λ^t-dominated and m-smaller in the order $(1, 2, \ldots, m-2, j, m-1, m, \ldots, j-1, j+1, \ldots, n)$, we also have $\lambda_q^{(j)} \leq \lambda_q^t$ for $j = q+1, q+2, \ldots, n$. This implies that for any $j = q+1, q+2, \ldots, n$, there exists $0 \leq \alpha_j \leq 1$ such that $\alpha_j \lambda_q^{(q)} + (1-\alpha_j)\lambda_q^{(j)} = \lambda_q^t$. Then by mixing policy $\tau^{(q)}$ with probability α_j and $\tau^{(j)}$ with probability $(1-\alpha_j)$, we have a policy that is $\boldsymbol{\lambda}^t$-dominated and m-smaller in the order $(1, 2, \ldots, m-2, j, m-1, m, \ldots, j-1, j+1, n)$ because

$$\alpha_j \lambda_i^{(q)} + (1-\alpha_j)\lambda_i^{(j)} \leq \lambda_i^t \quad \text{for all} \quad i = q+1, \ldots, n \text{ and } i \neq j$$

and

$$\alpha_j \lambda_i^{(q)} + (1-\alpha_j)\lambda_i^{(j)} = \lambda_i^t \quad \text{for all} \quad i = m-1, m, \ldots, q.$$

It is also clear that Q_{q+1} is true.

Finally, note that if Q_n is true, it immediately follows that Q_0 is true because

$$\lambda_i = \lambda_i^t \quad \text{for all} \quad i = m-1, \dots, n-1$$

together with $\sum_{i=m-1}^{n} \lambda_i \le \sum_{i=m-1}^{n} \lambda_i^t$, implying $\lambda_n \le \lambda_n^t$. Hence the statement ($Q_n$ or Q_0) implies Q_0 itself.

If we let $P(q)$ be the statement that Q_0 or Q_q is true, for $q = m-1, m, \dots, n$. Then the above implies that we have $P(m-1)$ being true, and that $P(q)$ implies $P(q+1)$ for $q = m-1, \dots, n-1$. Inductively we have $P(n)$ is true, which implies Q_0 is true. □

6.4. **Proof of Lemma 4.5.**

Proof. Without loss of generality, we assume $k = 1$ (for notational convenience). By Lemma 4.3, we can find $m_2^*, \dots, m_n^*$ such that the threshold policy is $\boldsymbol{\lambda}^t$-dominated in the order $(1, 2, \dots, n)$ whenever $m_2 = m_2^*, m_3 = m_3^*, \dots, m_n = m_n^*$. However, the result can be applied to any permutation of the $n-1$ servers (except Server k) by re-labeling. In the following, we demonstrate that we can get a mixed threshold policy that satisfies condition (1) by randomizing between these threshold policies.

Define $i_1 = 1$ for notational convenience. Let $P(m)$ be the statement that for any distinct $i_2, i_3, \dots, i_n \in \{2, 3, \dots, n\}$, there exists a mixed threshold policy $\tau_{m,i_1,i_2,\dots,i_{n-1},i_n}$ that is $\boldsymbol{\lambda}^t$-dominated and m-smaller in the order $(i_1, i_2, \dots, i_n)$. From the above we established $P(n)$.

Now, suppose $P(m)$ is true for some $2 < m \le n$. Then for any distinct $i_2, \dots, i_n \in \{2, 3, \dots, n\}$, there exists a mixed threshold policy $\tau_{m,i_1,i_2,\dots,i_n}$ that is $\boldsymbol{\lambda}^t$-dominated and m-smaller in the order $(i_1, i_2, \dots, i_n)$.

Now, for any distinct $i_2, \dots, i_n \in \{2, 3, \dots, n\}$, there exists threshold policies

$$\tau^{(l)} \equiv \tau_{m,i_1,i_2,\dots,i_{m-2},i_l,i_{m-1},\dots,i_{l-1},i_{l+1},\dots,i_{n-1}} \quad \text{for any } l = m-1, m, \dots, n$$

that is $\boldsymbol{\lambda}^t$-dominated and m-smaller in the following order

$$(i_1, i_2, \dots, i_{m-2}, i_l, i_{m-1}, \dots, i_{l-1}, i_{l+1}, \dots, i_n).$$

Then by Lemma 4.4 (with re-labelling), there exists a mixed threshold policy $\tau_{m-1,i_1,i_2,\dots,i_n}$ that is $\boldsymbol{\lambda}^t$-dominated and $(m-1)$-smaller in the order $(i_1, i_2, \dots, i_n)$, i.e. $P(m-1)$ is true. By induction, we have $P(2)$ is true, i.e., there exists a policy τ such that $\lambda_j \le \lambda_j^t$ for all $j = 2, 3, \dots, n-1$. It follows immediately from $\sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \lambda_i^t = \lambda$ that $\lambda_1 \ge \lambda_1^t$. □

6.5. **Proof of Proposition 1.**

Proof. For $m = 1, 2, \dots, n$, let $P(m)$ be the statement that for any fixed $i = m, m+1, \dots, n$, we have a mixed threshold policy $\tau_{m,i}$ such that

$$\begin{cases} \lambda_j^{(m,i)} = \lambda_j^t & j = 1, 2, \dots, m-1 \\ \lambda_i^{(m,i)} \ge \lambda_i^t & \\ \lambda_j^{(m,i)} \le \lambda_j^t & j = m, m+1, \dots, n \quad \text{and} \quad j \ne i. \end{cases}$$

Then from Lemma 4.5 we have $P(1)$ is true. Suppose $P(k)$ is true for some $k < n$. Then we have the mixed threshold policies $\tau_{k,k}, \tau_{k,k+1}, \dots, \tau_{k,n}$ such that

$$\begin{cases} \lambda_j^{(k,i)} = \lambda_j^t & j = 1, 2, \dots, k-1 \\ \lambda_i^{(k,i)} \ge \lambda_i^t & \\ \lambda_j^{(k,i)} \le \lambda_j^t & j = k, k+1, \dots, n \text{ and } j \ne i \end{cases}$$

for $i = k, k+1, \ldots, n$.

Then for $i = k+1, k+2, \ldots, n$, there exists a constant $0 \leq \alpha_{k,i} \leq 1$ such that

$$\alpha_{k,i}\lambda_k^{(k,k)} + (1-\alpha_{k,i})\lambda_k^{(k,i)} = \lambda_k^t.$$

Let $\tau_{k+1,i}$ denote the policy that mixes $\tau_{k,k}$ and $\tau_{k,i}$ with probability $\alpha_{k,i}$ and $1-\alpha_{k,i}$. Then we also have

$$\lambda_j^{(k+1,i)} = \begin{cases} \alpha_{k,i}\lambda_j^{(k,k)} + (1-\alpha_{k,i})\lambda_j^{(k,i)} = \lambda_j^t \\ \quad \text{for } j = 1, 2, \ldots, k-1, k. \\ \alpha_{k,i}\lambda_j^{(k,k)} + (1-\alpha_{k,i})\lambda_j^{(k,i)} \leq \alpha_{k,i}\lambda_j^t + (1-\alpha_{k,i})\lambda_j^t = \lambda_j^t \\ \quad \text{for } j = k+1, \ldots, n \text{ and } j \neq i. \end{cases}$$

Moreover,

$$\lambda_i^{(k+1,i)} = \lambda - \sum_{j \neq i} \lambda_j^{(k+1,i)} \geq \lambda - \sum_{j \neq i} \lambda_j^t = \lambda_i^t.$$

Therefore $P(k+1)$ is true. By the principle of mathematical induction, we have $P(m)$ is true for all $m = 1, 2, \ldots, n$. Note that $P(n)$ means that there exists a mixed threshold policy $\tau_{n,n}$ such that

$$\begin{cases} \lambda_j^{(n,n)} = \lambda_j^t & j = 1, 2, \ldots, n-1 \\ \lambda_n^{(n,n)} \geq \lambda_n^t \end{cases}$$

However, since we have

$$\lambda_n^{(n,n)} = \lambda - \sum_{j \neq n} \lambda_j^{(n,n)} = \lambda - \sum_{j \neq n} \lambda_j^t = \lambda_n^t,$$

the result follows. □

6.6. **Proof of Proposition 2.**

Proof. Let $\boldsymbol{\lambda}^{(1)}$ be the allocated demand vector of a pure threshold policy with $m_2 = m_3 = \ldots = m_n = 0$ and Server i being the i^{th} server, $i = 1, 2, \ldots, n$. Note that since $\mu_c > \lambda/n$ and all servers are used, the system is stable and thus $\sum_{i=1}^n \lambda_i^{(1)} = \lambda$.

For $j = 0, 1, \ldots, n-1$, let T_j be the pure threshold policy with $m_2 = m_3 = \ldots = m_n = 0$ and Server i being the $(i+j)^{th}$ server when $i+j \leq n$, and the $(i+j-n)^{th}$ server otherwise. Then we have

$$\lambda_i^{T_j} = \begin{cases} \lambda_{i+j}^{(1)} & i+j \leq n \\ \lambda_{i+j-n}^{(1)} & i+j > n \end{cases}$$

Let τ_j be the mixed threshold policy that is comprised $T_0, T_1, \ldots, T_{n-1}$, each with probability $1/n$ being used. Then we have for any $i = 1, 2, \ldots, n$

$$\lambda_i^\tau = \sum_{j=0}^{n-1} \lambda_i^{T_j}/n = \frac{1}{n} \cdot \left(\sum_{j=0}^{n-i} \lambda_{i+j}^{(1)} + \sum_{j=n-i+1}^{n-1} \lambda_{i+j-n}^{(1)} \right) = \frac{1}{n} \cdot \left(\sum_{j=i}^{n} \lambda_j^{(1)} + \sum_{j=1}^{i-1} \lambda_j^{(1)} \right) = \frac{\lambda}{n}.$$

□

6.7. **Proof of Proposition 3.**

Proof. First, because all servers have the same service capacity and all thresholds are zero, the state of the system can be represented by the number of customers in the system. Moreover, the designation of $1^{st}, 2^{nd}, \ldots, n^{th}$ servers do not affect the expected number of customers in the system because all servers have the same service capacities. Let X_c denote the number of customers in the n-server common queue system with all service capacities being μ_c. Also, let X_t denote the number of customers in the system under an n-server threshold allocation policy with all thresholds being zero. Suppose the system is subject to the same arrivals and service times, then we have $X_c = X_t$ Taking expectation, we have $E[X_c] = E[X_t]$.

The Little's Queueing formula states that in a stable system we have $L = \lambda W$, where L is the long-term average number of customers in the system and W is the long-term average time a customer spends in the system. Since λ is the same for both systems under consideration, we have the expected sojourn times equal, i.e. $W_c = W_t$ where W_c and W_t are, respectively, the expected sojourn time in the n-server common-queue system and under the threshold allocation policy with all threshold being zeros. □

6.8. **Proof of Proposition 4.**

Proof. Since $\bar{\mu}_n$ is given by

$$c(\bar{\mu}_n) = \frac{\lambda R}{n},$$

$a_n = \lambda/\bar{\mu_n}$ goes to infinity as n goes to infinity.

To find out the limit, we first note that

$$\lim_{n\to\infty} \frac{a_n}{n} = \lim_{n\to\infty} \frac{\lambda/n}{\bar{\mu}_n} = \lim_{n\to\infty} \frac{\lambda/n}{c^{-1}(\lambda R/n)} = \lim_{x\to\infty} \frac{\lambda/x}{c^{-1}(\lambda R/x)}.$$

Applying L'Hôpital's rule we get

$$\lim_{x\to\infty} \frac{\lambda/x}{c^{-1}(\lambda R/x)} = \lim_{x\to\infty} \frac{-\lambda/x^2}{-\lambda R(c^{-1})'(\lambda R/x)/x^2} = \lim_{x\to\infty} \frac{1}{R(c^{-1})'(\lambda R/x)}.$$

Finally note that

$$\frac{1}{(c^{-1})'(\lambda R/n)} = c'(\bar{\mu}_n)$$

by the inverse function theorem. Thus

$$\lim_{n\to\infty} \frac{a_n}{n} = \lim_{n\to\infty} \frac{c'(\bar{\mu}_n)}{R} = \frac{c'(0)}{R},$$

since $\lim_{n\to\infty} \bar{\mu}_n = 0$ and $c'(.)$ is continuous.

Case 1: $c'(0) = 0$

Then we have

$$\lim_{n\to\infty} \frac{a_n}{n} = \frac{c'(0)}{R} = 0$$

$$\begin{aligned} \sum_{k=0}^{n-1}(k+1)! \left(\begin{array}{c} n-1 \\ k \end{array} \right) \left(\frac{1}{a_n}\right)^k &\geq \sum_{k=0}^{n-1} \left(\begin{array}{c} n-1 \\ k \end{array} \right) \left(\frac{1}{a_n}\right)^k \\ &= \exp\left\{ \ln\left(1+\tfrac{1}{a_n}\right)^{n-1} \right\} \end{aligned} \tag{6}$$

For any real $x > 0$,

$$\begin{aligned}
&\lim_{x\to\infty} \ln\left(1+\frac{1}{a_x}\right)^{x-1} \\
&= \lim_{x\to\infty} \left[(x-1)\ln\left(1+\frac{1}{a_x}\right)\right] \\
&= \lim_{x\to\infty} \left[x\ln\left(1+\frac{1}{a_x}\right) - \ln\left(1+\frac{1}{a_x}\right)\right] \\
&= \lim_{x\to\infty} \frac{\ln(1+1/a_x)}{1/x} - 0 = \lim_{x\to\infty} \frac{\ln\left(1+c^{-1}(\lambda R/x)/\lambda\right)}{1/x} \\
&= \lim_{x\to\infty} \frac{\left(1+c^{-1}(\lambda R/x)/\lambda\right)^{-1}\cdot(1/\lambda)\cdot(c^{-1})'(\lambda R/x)\cdot(\lambda R)\cdot(-1/x^2)}{-1/x^2} \\
&= \lim_{x\to\infty} \frac{R(c^{-1})'(\lambda R/x)}{1+c^{-1}(\lambda R/x)/\lambda} = \lim_{x\to\infty} \frac{\lambda R/c'(\bar{\mu}_x)}{\lambda+\bar{\mu_x}} = \infty
\end{aligned}$$

since $c'(0) = 0$ and $\lim_{x\to\infty} \bar{\mu}_x = 0$.

Consequently we have

$$\lim_{n\to\infty} \left(\sum_{k=0}^{n-1}(k+1)!\begin{pmatrix} n-1 \\ k \end{pmatrix}\left(\frac{1}{a_n}\right)^k\right)^{-1} = 0$$

which then yields

$$\begin{aligned}
\lim_{n\to\infty} \frac{W_{sd}}{W_{si}} &= \lim_{n\to\infty} \left[\frac{a_n}{n}\left(\sum_{k=0}^{n-1}(k+1)!\begin{pmatrix} n-1 \\ k \end{pmatrix}\left(\frac{1}{a_n}\right)^k\right)^{-1} + \frac{n-a_n}{n}\right] \\
&= 0\cdot 0 + 1 - 0 = 1.
\end{aligned}$$

Case 2: $c'(0) \neq 0$

Then we have

$$\lim_{n\to\infty} \frac{a_n}{n} = \frac{c'(0)}{R} \neq 0$$

Using similar methods as in Case 1, we obtain

$$\lim_{x\to\infty} \ln\left(1+\frac{1}{a_x}\right)^{x-1} = \lim_{x\to\infty} \frac{\lambda R/c'(\bar{\mu}_x)}{\lambda+\bar{\mu_x}} = \frac{R}{c'(0)}$$

Using again inequality (6), we have

$$\lim_{n\to\infty} \left(\sum_{k=0}^{n-1}(k+1)!\begin{pmatrix} n-1 \\ k \end{pmatrix}\left(\frac{1}{a_n}\right)^k\right)^{-1} \leq e^{-R/c'(0)}.$$

Finally we have

$$\begin{aligned}
\lim_{n\to\infty} \frac{W_{sd}}{W_{si}} &= \lim_{n\to\infty} \left[\frac{a_n}{n}\left(\sum_{k=0}^{n-1}(k+1)!\begin{pmatrix} n-1 \\ k \end{pmatrix}\left(\frac{1}{a_n}\right)^k\right)^{-1} + \frac{n-a_n}{n}\right] \\
&\leq \frac{c'(0)}{R}e^{-R/c'(0)} + 1 - \frac{c'(0)}{R} = 1 - \frac{c'(0)}{R}(1-e^{-R/c'(0)}) < 1.
\end{aligned}$$

□

Acknowledgments. The authors would like to thank all the anonymous reviewers for their helpful comments, corrections and suggestions. The preliminary version of the paper has been presented and published in the proceedings of the 40th International Conference on Computer & Industrial Engineering (CIE2010) 2010, Hyogo, Japan [4]. Research supported in part by RGC Grant 7017/07P and HKU Strategic Research Theme Fund on Computational Sciences.

REFERENCES

[1] G. Cachon and F. Zhang, *Obtaining fast service in a queueing system via performance-based allocation of demand*, Management Science, **53** (2007), 408–420.
[2] W. Ching, S. Choi and M. Huang, *Optimal service capacity in a multiple-server queueing system: A game theory approach*, Journal of Industrial and Management Optimization, **6** (2010), 73–102.
[3] S. Choi, X. Huang, W. Ching and M. Huang, *Incentive effects of multiple-server queueing networks: the principal-agent perspective*, East Asian Journal on Applied Mathematics, **1** (2011), 379–402.
[4] S. Choi, X. Huang and W. Ching, *Inducing optimal service capacities via performance-based allocation of demand in a queueing system with multiple servers*, in "The 40^{th} International Conference on Computers and Engineering," Hyogo, Japan, 2010.
[5] T. Crabill, D. Gross and M. Magazine, *A classified bibliography of research on optimal design and control of queues*, Operations Research, **25** (1977), 219–232.
[6] S. Gilbert and Z. Weng, *Incentive effects favor nonconsolidating queues in a service system: The principal-agent perspective*, Management Science, **44** (1998), 1662–1669.
[7] E. Kalai, M. Kamien and M. Rubinovitch, *Optimal service speeds in a competitive environment*, Management Science, **38** (1992), 1554–1163.
[8] J. Laffont and D. Martimort, "The Theory of Incentives: The Principal-Agent Model," Princeton University Press, Princeton, NJ, 2002.
[9] W. Lin and P. Kumar, *Optimal control of a queueing system with two heterogeneous servers*, IEEE Trans. Automatic Control, **29** (1984), 696–703.
[10] H. Luh and I. Viniotis, *Threshold control policies for heterogeneous server systems*, Mathematical Methods of Operations Research, **55** (2002), 121–142.
[11] M. Osborne, "An Introduction to Game Theory," Oxford University Press, New York, 2004.
[12] M. Rubinovitch, *The slow server problem*, Journal of Applied Probability, **22** (1985), 205–213.
[13] F. Véricourt and Y. Zhou, *On the incomplete results for the heterogeneous server problem*, Queueing Systems, **52** (2006), 189–191.
[14] F. Zhang, "Coordination of Lead Times in Supply Chains," Dissertation, University of Pennsylvania, Philadelphia, PA, 2004.

Received October 2010; 1st revision June 2011; 2nd revision August 2011.

E-mail address: kelly.smchoi@berkeley.edu
E-mail address: hehe1121@gmail.com
E-mail address: wching@hku.hk