ON THE OPTIMALITY OF THE STE RULE FOR MULTIPLE SERVER QUEUES THAT SERVE CUSTOMERS WITH DEADLINES

Shivendra S. Panwar and Don Towsley

COINS Technical Report 88-81

July 1988

ON THE OPTIMALITY OF THE STE RULE FOR MULTIPLE SERVER QUEUES THAT SERVE CUSTOMERS WITH DEADLINES *

Shivendra S. Panwar
Department of Electrical Engineering
Polytechnic University
Brooklyn, NY 11201

Don Towsley
Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

July 1988

Abstract

Many problems in real-time systems can be modeled by multiple server queues serving customers with deadlines. If a task is not completed within a certain time interval of its arrival in such a system, it is useless and need not be served. It is therefore desirable to schedule the customers such that the fraction of customers served within their respective deadlines is maximized. For this measure of performance it is shown that the shortest time to extinction (STE) policy is optimal for a class of continuous and discrete time nonpreemptive G/M/c queues that do not allow unforced idle times for the case that customers must either begin service by their deadline or complete service by their deadline. When unforced idle times are allowed, the best policies belong to the class of shortest time to extinction with

*This work was supported by the National Science Foundation under Grant NSF ECS 83-10771, by the Office of Naval Research under Contract N0014-87-K-0796, and by the Center for Advanced Technology in Telecommunications, Polytechnic University, Brooklyn, NY.

inserted idle time (STEI) policies. This result is shown to be true for the G/G/c queue in the case that deadlines are to the beginning of service and the G/M/c queue for the case that deadlines are to the end of service. Here an STEI policy requires that the customer closest to his deadline be scheduled whenever a customer is scheduled. An STEI policy also has the choice of inserting idle times while the queue is nonempty. We is also show that STE is optimal for the continuous and discrete time preemptive G/M/c queue where deadlines are to the end of service. This last result for the preemptive G/M/c queue is generalized to a queue in which the servers are allowed to take vacations.

1 Introduction

Increasing interest has been shown recently on the design and analysis of real-time multiprocessor systems. The workloads to these systems consist of customers that have real-time constraints, i.e., customers must complete or enter service by specified deadlines. For some systems it is unacceptable for any task to miss its deadline. In these systems task service demands are usually well understood and a substantial literature has focussed on the development and evaluation of scheduling policies for these workloads, [10,11]. Other workloads consist of tasks for which it is not critical that all tasks meet their constraints. Usually, the service requirements and the arrival patterns are not as well understood and the objective is to design policies that will minimize the fraction of tasks that miss their deadlines. The purpose of this paper is to study optimal policies for this second class of workloads.

In this paper we consider as our model for a multiprocessor, a multiple server queue that serves customers with deadlines. We wish to determine the class of policies that maximizes the fraction of customers which successfully complete service, i.e., do not miss their deadlines. We consider two classes of workloads, 1) those workloads consisting of customers that must begin service by their deadline and 2) those workloads in which customers must complete service by their deadline. For the first class of workloads we obtain the following results for G/GI/c queues that do not allow preemptions. We show that for any arbitrary policy, there exists a policy from the class of shortest time to extinction with unforced idle times (STEI) policies. If an optimal policy does exist, then it must be an STEI policy. Here an STEI policy is one that, whenever the queue is not empty, may choose to schedule either no customer or the customer closest to its deadline. When we restrict ourselves to the class of policies that do not allow the processor to remain idle when there are customers in the queue, then the shortest time to extinction (STE) policy is optimal for the G/M/c queue. Here the STE policy schedules the customer closest to its deadline.

For the second class of systems, where the deadlines are to the end of service, we have results for the class of preemptive policies as well as non-preemptive policies. For the first class of policies we show that STE is the optimum policy for the G/M/c queue. For the class of non-preemptive policies, the best policies belong to the class of STEI policies for the G/M/c queue. When we restrict ourselve to non-idling non-preemptive policies then STE is the optimal policy for the G/M/c queue. We extend the optimality results for the preemptive G/M/c queue where deadlines are to the end of service to queues where servers take vacations.

All of the above results are shown to hold for both continuous time and discrete time queues. These latter results are particularly useful in the context of communication networks where discrete time queues are standard models for statistical multiplexers and concentrators, [17].

The shortest time to extinction (STE) policy, which will be described in Section 2, is very similar to the earliest due date (EDD) scheduling policy proposed by Jackson [9]. Consider a set of n tasks $\{T_i, 1 \leq i \leq n\}$ with the corresponding n due dates $\{d_i, 1 \leq i \leq n\}$. Let the finishing times under schedule S be $f_i(S)$. Then the lateness of T is defined as $f_i(S) - d_i$ and the tardiness is defined as $max\{0, f_i(S) - d_i\}$. Jackson showed that the maximum lateness and maximum tardiness are minimized by sequencing the tasks in the order of non-decreasing due dates. As we shall see in Section 3, STE scheduling differs from EDD scheduling in that it never schedules tasks which are already past their due dates. Note that the tasks and their due dates are known a priori under Jackson's model. Using the same a priori information, Moore [12] devised an algorithm to minimize the number of late tasks. Pinedo [16] considered the problem of minimizing the number of late jobs (customers) when the processing times are exponentially distributed and the deadlines are randomly distributed. He assumed that no new jobs are allowed into the system once the processing begins. Su and Sevcik [18] consider the problem of scheduling customers with deadlines in a queue. They showed that EDD scheduling minimized performance parameters such as expected lateness and tardiness.

Pierskalla and Roach [15] showed that a policy similar to the STE policy is an optimal issuing policy under the conditions which prevail in blood banks. Here, the additions to the blood bank ("customer arrivals") are random as is the demand ("customer service times") and the issuing policy should be such that the amount of blood which becomes unusable as a result of being stored too long is minimized. More recently, while considering scheduling problems which arise in the area of real time systems, Dertouzos [6] has shown that for any arbitrary set of arrivals with arbitrary processing times and deadlines the EDD policy is optimal if preemptions are allowed. Here a (real time) scheduling policy is considered optimal if it produces a feasible schedule whenever a clairvoyant scheduling policy (which is aware of future job arrivals) can do so. Interestingly, the EDD policy is under study by GTE, Inc., which is considering this policy for its integrated packet-switched networks [21]. In queueing theory literature, queues with impatient customers have been usually analyzed assuming a FCFS scheduling policy [1,4,7].

In [14], we have considered the problem of a single server queue with impatient customers under the assumption that deadlines are until customers enter service. We show that the STE policy is optimal for a large class of single server queues. The shortest time to extinction with unforced idle times (STEI) class of policies are shown to be optimal for a larger class of queues. Similar results for the continuous time single server queue when the deadlines are to the end of service can be found in [13,2]. The results found in this paper extend the results in [13,2] in several ways. First, our results are for multiple server queues. Last, neither of the above references consider queues in which servers take vacations.

This paper is organized as follows. Section 2 contains a model of the system under study along with definitions of the different scheduling policies of interest to us. The main results of the paper are contained in sections 3, 4, and 5. Section 3 contains the results for systems with deadlines to the beginning of service, section 4 contains results for systems with deadlines to the end of service that allow preemptions and section 5 contains results for systems with deadlines to the end of service without preemptions. In Section 6, we conclude the paper by summarizing our results.

2 Definitions and Notation

We consider three different multiple server queues,

- Nonpreemptive queues with deadlines to the beginning of service,
- Nonpreemptive queues with deadlines to the end of service where a customer that misses its deadline while in service is aborted,
- Preemptive queues with deadlines to the end of service.

In all of these systems let T_i denote the arrival time of the i-th customer. Let A_i denote the time between the arrivals of the $(i-1)$-th and i-th customers. We assume that A_i is a random variable with arbitrary distribution. Let E_i denote the extinction time of the i-th customer (i.e., the time by which it must be served). Here $E_i = T_i + D_i$ where D_i is a random variable with a general distribution. We shall refer to D_i as the real time constraint or the relative deadline for customer i. Last, let $\{B_i\}_{1\le i}$ be an independent and identically distributed (i.i.d.) sequence of random variables with a general distribution which will be used to assign service times to customers.

We shall use the notation $\boldsymbol{A}_N = \{A_i\}_{1\le i\le N}$, $\boldsymbol{D}_N = \{D_i\}_{1\le i\le N}$, $\boldsymbol{B}_N = \{B_i\}_{1\le i\le N}$, and $\boldsymbol{S}_N = (\boldsymbol{A}_N, \boldsymbol{D}_N, \boldsymbol{B}_N), 1 \le N$. In addition, whenever we focus on a specific sample realization of the above r.v.'s, we shall use lowercase notation (i.e., a_i for A_i, etc ...). Furthermore, we shall let $\boldsymbol{a} = \{a_i\}_{1\le i}$, $\boldsymbol{b} = \{b_i\}_{1\le i}$, $\boldsymbol{d} = \{d_i\}_{1\le i}$, $\boldsymbol{a}_N = \{a_i\}_{1\le i\le N}$, $\boldsymbol{b}_N = \{b_i\}_{1\le i\le N}$, and $\boldsymbol{d}_N = \{d_i\}_{1\le i\le N}$. Last, let $s = (\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b})$ and $s_N = (\boldsymbol{a}_N, \boldsymbol{d}_N, \boldsymbol{b}_N)$, $N = 1, \ldots$. These last two quantities will be referred to as an input sample and finite input sample respectively.

At this point in the paper we will not specify how service times from the sequence $\{B_i\}$ are assigned to customers. The assignment rule will depend on which system we are interested in and what property we wish to prove with regard to that system. We use the notation A/B/C + D to denote a queue with customer deadlines where A, B and C has the same meaning as in Kendall's notation while D gives the distribution of the relative deadlines.

We make the following additional assumptions

A1 $\{B_i | 1 \le i\}$ is independent of $\{A_i\}$ and $\{D_i\}$,

A2 $\lim_{n\to\infty} P[\sum_{i=1}^{n} A_i < t] = 0$ for any $0 \le t < \infty$.

The second assumption is not particularly restrictive. For example, it is valid for a Poisson arrival process.

Lemma 1 *If arrivals are described by a Poisson process with intensity λ then* $\lim_{n\to\infty} P[\sum_{i=1}^{n} A_i < t] = 0$ *for any* $0 \le t < \infty$.

Proof. We perform the following calculation.

$$\lim_{n\to\infty} nP[\sum_{i=1}^{n} A_i < t] = \lim_{n\to\infty} n \sum_{k=n}^{\infty} (\lambda t)^k e^{-\lambda t}/k!$$

It is possible to bound the sum on the right hand side by an expression of the form $C\alpha^{n-n_0}\sum_{k=0}^{\infty}\alpha^k$ for $n > n_0$ where n_0 is chosen so that $(\lambda t)/n_0 < \alpha < 1$. Thus we have

$$\begin{aligned}\lim_{n\to\infty} nP[\sum_{i=1}^{n} A_i < t] &< \lim_{n\to\infty} nC\alpha^{n-n_0}\sum_{k=0}^{\infty}\alpha^k, \\ &= \lim_{n\to\infty} nC\alpha^{n-n_0}/(1-\alpha), \\ &= 0.\end{aligned}$$

QED

Similar arguments can be used to extend the above result in a number of different directions. First, renewal processes with interarrival times given by a phase type distribution also satisfy A2. Second, customers may arrive in batches so long as the mean batch size is finite. Third, many non-renewal processes whose structure can be described by a discrete state Markov process can also be shown to exhibit this property. An example of such a process is the *Markov- modulated Poisson process* [8].

Let π be a policy that determines what customer in the queue is to be executed (if any) whenever the server is free. This policy makes its decision based on the customers that are eligible for service as well as on the past history of the system. We wish to choose π so that we maximize the fraction of customers beginning service before their respective extinction times. Consider a system in which exactly N customers arrive for service. We define $V_N(\pi)$ to be the number of customers served for this system. We are interested in the fraction, $\overline{V}_N(\pi) = E[V_N(\pi)]/N$, of customers served in this system. We define the fraction of customers served in the system as $N \longrightarrow \infty$ (under policy π) to be

$$\overline{V}(\pi) = \liminf_{N\to\infty} \overline{V}_N(\pi).$$

Finally, let $\overline{V} = \sup_\pi \overline{V}(\pi)$. A policy π^* is *optimal* if $\overline{V}(\pi^*) = \overline{V}$.

We are also interested in the fraction of customers served by time t. Let $V_t(\pi)$ denote the number of customers that make their deadlines by time t. Let $\overline{V}_t(\pi) = E[V_t(\pi)]$.

We find it easier to work with $\overline{V}_N(\pi)$. Fortunately, all of the results that we prove for $\overline{V}_N(\pi)$ also hold for $\overline{V}_t(\pi)$ as a consequence of assumption A2.

A customer is eligible under policy π at time t if it has neither exceeded its deadline nor ended service. Consequently, the set of customers of interest at any time t will be denoted by $C_\pi(t) = \{c_{j_1}, c_{j_2}, \cdots, c_{j_n}\}$ consisting of all the eligible customers at time $t, j_i \geq 1,\ 1 \leq i \leq n$. Here c_i denotes the i-th customer to arrive to the system. The set of extinction times of these customers will be denoted by $\boldsymbol{E}_\pi(t)$.

Consider the actions that policy π can take at time t. If all the servers are busy, then π takes no action if preemptions are not allowed. If any server is idle at time t or if preemptions are allowed, then π can either schedule no customer or schedule customers from $C_\pi(t)$. Policy π is allowed to choose one of these actions according to some distribution that depends on $\pi, C_\pi(t)$ and the previous history $\boldsymbol{H}_t$ (to be defined later in this section). We define $p_j(\pi, t, \boldsymbol{H}_t)$ to be the probability that π schedules customer $c_j \in C_\pi(t), 1 \leq i \leq n$ on an idle server and $p_0(\pi, t, H_t)$ to be the probability that π chooses to schedule no customer.

If π chooses not to schedule a customer at time t and $C_\pi(t) \neq 0$, then it delays making a new scheduling decision by a random amount of time τ with some arbitrary distribution function $F_\tau(x|H_t)$ (τ takes on discrete values in the case of a discrete time queue). The policy does not perform a scheduling decision until either τ time units elapse or an arrival occurs. Without loss of generality, we may impose one last constraint on π, namely, π is prohibited from scheduling two successive idle times on the same server when the queue is nonempty unless they are separated by the arrival of one or more customers.

In the case that π is allowed to preempt customers, we introduce some additional parameters. If π decides to schedule a customer at time t, then $q(\pi, t, \boldsymbol{H}_t)$ is the probability that the customer will not be preempted in the absence of customer arrivals and service completions. The customer is scheduled for preemption with probability $1-q(\pi, t, \boldsymbol{H}_t)$ and is provided with τ units of service where τ has cumulative distribution function $H_\tau(x|\boldsymbol{H}_t)$. The customer is preempted after τ units of time provided it has not completed by that time and there have been no arrivals or service completions of other customers. If an arrival or a service completion occurs, then π is allowed to reschedule the customer if it so desires.

The history of the system up to time t may be defined by $\boldsymbol{H}_t = (\boldsymbol{a}_t, \boldsymbol{d}_t, \boldsymbol{r}_t, \boldsymbol{f}_t, \boldsymbol{e}_t, \boldsymbol{u}_t)$ where $\boldsymbol{a}_t$ is an ordered set of arrival times of all customers that arrive prior to $t, \boldsymbol{d}_t$ is an ordered set of relative deadlines corresponding to the customers that arrive prior to $t, \boldsymbol{r}_t$, $\boldsymbol{f}_t$, and $\boldsymbol{e}_t$ are ordered sets containing the times of all scheduling decisions prior to time t, the identities of the customers and the servers to which they were scheduled respectively. In addition, $\boldsymbol{u}_t$ is an ordered set of the service times for customers completed prior to time t.

We now define the policies that we will study in this paper. Let t'_k denote the time of the k^{th} scheduling decision since time $t = 0$.

Definition 1 *Policy π is the shortest time to extinction (STE) policy if at time $t'_k, (1 \leq k)$, it always schedules the eligible customer with the smallest deadline on any one of the available servers. In addition, the server is always busy as long as eligible customers are available which have not yet been served, i.e $p_0(\pi, t) = 0$ whenever the server is available and $C_\pi(t) \neq \phi$.*

An example of how the STE policy schedules a given set of arrivals is shown in Fig. 1(a) for a single server system when deadlines are to beginning of service.

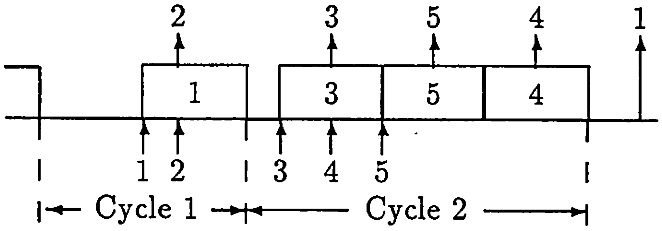
Definition 2 *Unforced idle times are time intervals when any server is idle while eligible customers are available.*

Definition 3 *Policy π is a shortest time to extinction with unforced idle times (STEI) policy if, at time t'_k, it schedules the eligible customer with the smallest deadline on any one of the available servers. In other words, $p_0(\pi, t'_k) \geq 0, p_j(\pi, t'_k) \geq 0$ if $j = \arg\min_{j \text{ s.t. } c_j \in C_\pi(t'_k)} E_\pi(t'_k)$ and $p_j(\pi, t'_k) = 0$ otherwise.*

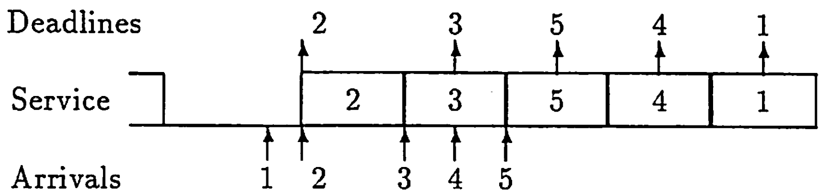
The STE policy, defined earlier, is an example of a STEI policy. Fig.1(b) shows how an STEI policy might schedule the same set of arrivals as shown in Fig. 1(a). Note that the STEI policy schedules all the arrivals while the STE policy leads to the loss of one arrival in this particular case. Fig. 1(c) illustrates how a FCFS (first-come, first-served) policy schedules the arrivals.

3 The Nonpreemptive Queue with Deadlines to Beginning of Service

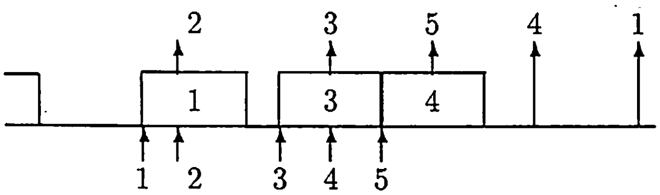
In this section we show that there is no class of policies better than the STEI policies for the non-preemptive G/G/c+G queue when the deadline is to the beginning of service.



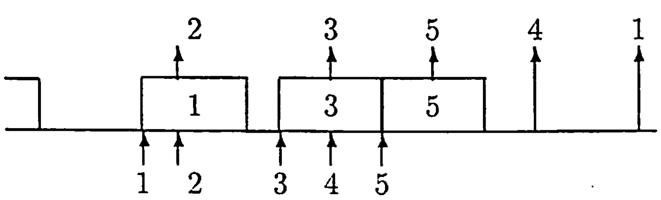
(a)



(b)



(c)



(d)

Figure 1: (a) Behavior of STE, (b) Behavior of STEI, (c) Behavior of FCFS, (d) STEI emulating FCFS

We also show that STE is the best policy out of the class of *non-idling policies* for this class of systems when service times are restricted to be independent and identically distributed exponential random variables with parameter μ (geometrically distributed in the case of a discrete time queue). In the course of proving these results, we shall compare sets of extinction times and show that one set *dominates* another set. Consequently, the first step is to define dominance and establish some properties that are satisfied by this relation.

Consider two sets of nonnegative real numbers $R = \{x_1, x_2, \cdots, x_n\}$ and $S = \{y_1, y_2, \cdots, y_m\}$ each ordered so that $x_i \geq x_{i+1}$, $i = 1, \cdots n$ and $y_i \geq y_{i+1}$, $i = 1, \cdots m$.

Definition 4 *We say that R dominates S ($R \succ S$) if $n \geq m$ and $x_i \geq y_i$, $i = 1, 2, \cdots m$.*

We define the following three operations

- $Large(R, k) = \{x_1, x_2, \cdots, x_k\}, \;\; 0 \leq k \leq n.$
- $Small(R, k) = \{x_{n-k+1}, \cdots, x_n\}, \;\; 0 \leq k \leq n.$
- $Shift(R, x) = \{x_i - x \mid x_i \geq x\}.$

The following lemma gives conditions under which dominance is preserved when set operations, the *Large* operation, and the *Shift* operation are performed on R and S.

Lemma 2 *If $R \succ S$, then:*

1. $R + \{x\} \succ S + \{x\}$, *for* $x > 0$,
2. $R - \{x\} \succ S$, *where* $x = \min_{1 \leq i \leq n}\{x_i\}$ *and* $n > m$,
3. $R \succ S - \{y\}$, *where* $y \in S$,
4. $R - \{x\} \succ S - \{y\}$, *where* $x \in R$, $y \in S$, *and* $x \leq y$,
5. *Assume that* $R = \{x_1, \cdots, x_n\}$ *where* $x_i \geq x_{i+1}$, $1 \geq i < n$ *and* $S = \{y_1, \cdots, y_m\}$ *where* $y_i \geq y_{i+1}$, $1 \geq i < m$. *Then* $R - \{x_k\} \succ S - \{y_j\}$ *for* $k \geq j$,
6. $Shift(R, x) \succ Shift(S, x)$.

7. $Large(R, m) \succ S$.

Proof: The proof of 1, 2, 3, and 6 may be found in [14]. Properties 4, 5 and 7 follow from the operations performed on R and S and the definition of "$\succ$".
QED

We first show that any non-STEI policy π^* can be emulated by some STEI policy π in the sense that $\overline{V}_N(\pi) = \overline{V}_N(\pi^*)$ for all N and $\overline{V}_t(\pi) = \overline{V}_t(\pi^*)$ for all t. Consequently, the STEI class of policies contains the best policies, i.e., those with the highest performance. Thus the designer of a real-time system need only consider this class of policies.

Theorem 1 *For any policy π, there exists an STEI policy π^* such that $\overline{V}_N(\pi^*) = \overline{V}_N(\pi)$, $0 < N$, $\overline{V}(\pi^*) = \overline{V}(\pi)$, and $\overline{V}_t(\pi^*) = \overline{V}_t(\pi)$, $0 < t$ for the $G/G/c + G$ queue without preemptions and with deadline to beginning of service.*

Proof: Consider any policy π not in the class of STEI policies We shall construct an STEI policy π^* that exhibits the same performance as that of π. Policy π^* is defined as follows:

1. π^* maintains an ordered list of customers at time $t, \mathcal{A}(t)$ that would be eligible under π at that time when provided the same input sample, i.e., $\mathcal{A}(t) = \mathcal{C}_\pi(t)$.
2. π^* maintains a history H_t' identical to the history that π would produce when given the same input sample, i.e., $H_t' = H_t$.
3. π^* makes scheduling decisions according to the following rules
   (a) At time t, it schedules the customer closest to its deadline with probability $1 - p_0(\pi, t, H_t')$.
   (b) At time t, it schedules no customer with probability $p_0(\pi, t, H_t')$.
4. π^* modifies $\mathcal{A}(t)$ as follows,
   (a) customer c_i is removed from $\mathcal{A}(t)$ either 1) when its deadline expires, or 2) with probability $p_i(\pi, t, H_t')$ at a time t when π^* schedules a customer,
   (b) customer c_i is added to $\mathcal{A}(t)$ when it arrives to the system.

5. π^* modifies H_t' as follows,

   (a) at the time of an arrival the arrival time and relative deadline the customer are added to $\boldsymbol{a}_t$ and $\boldsymbol{d}_t$.

   (b) at the time of a departure, the service time of the customer is added to $\boldsymbol{d}_t$.

   (c) at the time that π^* assigns a customer to service, the identity of the customer removed from $\mathcal{A}(t)$ (see 4.(a) above) and the time of the assignment are added to $\boldsymbol{r}_t$ and $\boldsymbol{e}_t$, respectively.

We focus on the behavior of π and π^* given $\boldsymbol{S} = \boldsymbol{s}$. Here service times are assigned service times in the order that they are scheduled, i.e., the i-th customer scheduled is given b_i as its service time.

Policy π^* exhibits the same behavior as π (i.e., $E[V_N(\pi^*)|\boldsymbol{S} = s] = E[V_N(\pi)|\boldsymbol{S} = s]$, $N = 1, 2, \cdots$) provided that $\mathcal{A}(t)$ and $\mathcal{C}_\pi(t)$ exhibit the same behavior. This latter statement is true if $\boldsymbol{E}_{\pi^*}(t) \succ \boldsymbol{E}_\pi(t)$ for all s. We prove this last dominance relation by induction.

Consider policy π, policy π^* as defined above, and a single input sample s. We need only focus on the points of time that either a customer arrives, a customer departs, a customer misses a deadline, or a customer is scheduled into service in the systems operating under π and π^*. Let $t_0 = 0 \le t_1 \le \cdots \le t_i \le t_{i+1} \le \cdots$ denote these times.

It is useful to distinguish among the following events:

$\mathcal{E}_1$ - Arrival of a customer at both systems.

$\mathcal{E}_2$ - Service completion at one or both systems.

$\mathcal{E}_3$ - Loss of one or more customers at one or both systems due to missing of deadline.

$\mathcal{E}_4$ - Scheduling of a customer to service in one or both systems.

A more complete description of the history of both systems is given by the sequence of event-time pairs $(t_0, \sigma_0), (t_1, \sigma_1), \cdots, (t_i, \sigma_i), \cdots, (t_n, \sigma_n)$ where $\sigma_0 = \mathcal{E}_1$ and t_i is the time at which an event of type $\sigma_i \in \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\}$ $(1 \le i < n)$ occurs. If two types of events occur simultaneously, we represent them as separate events with identical event times. In some cases the order of the events is determined by the mechanics of the system, i.e., scheduling a server immediately after a departure. Otherwise *(*(the order in which they are listed is immaterial. Because of the simultaneity of events we will abuse the notation $\boldsymbol{E}_\pi(t_i)$ and $\boldsymbol{E}_\pi(t_i^-)$ in the case that $t_i = t_{i+1}$ or $t_i = t_{i-1}$. If $t_i = t_{i+1}$, then $\boldsymbol{E}_\pi(t_i)$ will denote the set of waiting customers immediately after processing event σ_i but before processing event σ_{i+1}. Similarly, $\boldsymbol{E}_\pi(t_i^-)$ denotes the state of the queue prior to processing σ_i but after processing event σ_{i-1}.

In addition to observing the behavior of $\boldsymbol{E}_\pi(t)$ and $\boldsymbol{E}_{\pi^*}(t)$, we also focus on the number of customers that have received some or all of their service and the remaining service times of customers in service. Let $I_\pi(t)$ and $I_\pi^*(t)$ denote the first of these quantities and $\boldsymbol{R}_\pi(t) = \{r_\pi^1(t), \cdots, r_\pi^{n_\pi(t)}(t)\}$ and $\boldsymbol{R}_{\pi^*}(t) = \{r_{\pi^*}^1(t), \cdots, r_{\pi^*}^{n_{\pi^*}(t)}(t)\}$ denote the latter quantities where $n_\pi(t)$ and $n_{\pi^*}(t)$ are the number of busy servers at time t under π and π^* respectively. In addition to showing $\boldsymbol{E}_{\pi^*}(t) \succ \boldsymbol{E}_\pi(t)$ we will also show $I_\pi(t) = I_{\pi^*}(t)$ and $\boldsymbol{R}_\pi(t) = \boldsymbol{R}_{\pi^*}(t)$ for $t \geq 0$. [1] We will abuse the notation $\boldsymbol{R}_\pi(t_i)$, $I_\pi(t_i)$, $\boldsymbol{R}_\pi(t_i^-)$, and $I_\pi(t_i^-)$ as well. Here, the primary purpose of I_π and I_{π^*} is to identify the next service time to assign to a customer.

First observe that whenever $\boldsymbol{E}_{\pi^*}(t_i) \succ \boldsymbol{E}_\pi(t_i)$, $I_{\pi^*}(t_i) = I_\pi(t_i)$, $\boldsymbol{R}_{\pi^*}(t_i) = \boldsymbol{R}_\pi(t_i)$, and $t_{i+1} > t_i$, then $\boldsymbol{E}_{\pi^*}(t) \succ \boldsymbol{E}_\pi(t)$, $I_{\pi^*}(t) = I_\pi(t)$, $\boldsymbol{R}_{\pi^*}(t) = \boldsymbol{R}_\pi(t)$, for $t_i \leq t < t_{i+1}$. The first and third of these relations are a consequence of property 5 in Lemma 2. The second holds true since no new customer begins service in the interval (t_i, t_{i+1}). Thus we need only show that $\boldsymbol{E}_{\pi^*}(t) \succ \boldsymbol{E}_\pi(t)$, $I_{\pi^*}(t) = I_\pi(t)$, $\boldsymbol{R}_{\pi^*}(t) = \boldsymbol{R}_\pi(t)$ for $t = t_0, t_1, \cdots$. This we do by induction.

Basis Step: As both systems are initially in the same state at $t = t_0 = 0$, the relations hold.

Inductive step: Let us assume that the hypothesis is true for t_k, $k = 0, 1, \cdots, i$. We now show that it is also holds for t_{i+1}. There are several cases according to the type of event that occurs at time t_{i+1}.

Case 1 ($\sigma_{i+1} = \mathcal{E}_1$): First, note that $\boldsymbol{E}_{\pi^*}(t_{i+1}^-) \succ \boldsymbol{E}_\pi(t_{i+1}^-)$. Application of property 1 in Lemma 2 then yields $\boldsymbol{E}_{\pi^*}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$. Neither the number of customers that have begun service nor the remaining service times are affected in this case.

Case 2 ($\sigma_{i+1} = \mathcal{E}_2$): If a service completion occurs under π, then a service completion occurs under π^* at the same time. Application of property 5 of Lemma 2 yields $\boldsymbol{E}_{\pi^*}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$. Similarly, application of the same property along with the re-

[1] More correctly, we prove these relations for $t < \lim_{n\to\infty} t_n$. This last quantity may be finite.

moval of the remaining service time corresponding to the completed customer yields $\boldsymbol{R}_{\pi^*}(t_{i+1}) = \boldsymbol{R}_\pi(t_{i+1})$. Last, clearly $I_{\pi^*}(t_{i+1}) = I_\pi(t_{i+1})$.

Case 3 ($\sigma_{i+1} = \mathcal{E}_3$): There are three subcases according to whether a customer is lost under π^*, π, or both policies. In all of these subcases the quantities $\boldsymbol{R}_\pi(t)$, $\boldsymbol{R}_{\pi^*}(t)$, $I_\pi(t)$, and $I_{\pi^*}(t)$ are unaffected by losses so that $\boldsymbol{R}_{\pi^*}(t_{i+1}) = \boldsymbol{R}_\pi(t_{i+1})$ and $I_{\pi^*}(t_{i+1}) = I_\pi(t_{i+1})$. Consider the case where a customer is lost under π^* but not π. For this to happen and the inductive hypothesis to hold, $\boldsymbol{E}_{\pi^*}(t_{i+1}^-)$ must contain at least one more customer than $\boldsymbol{E}_\pi(t_{i+1}^-)$. Consequently, property 2 of Lemma 2 can be applied to show that $\boldsymbol{E}_{\pi^*}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$.
If $\sigma_{i+1} = \mathcal{E}_3$ corresponds to the loss of a customer under π, then property 3 of Lemma 2 can be used to show $\boldsymbol{E}_{\pi^*}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$. Similarly, property 4 of Lemma 2 can be used in the case of loss of a customer under both π^* and π to show $\boldsymbol{E}_{\pi^*}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$.
Case 4 ($\sigma_{i+1} = \mathcal{E}_4$): Since $\boldsymbol{E}_{\pi^*}(t_{i+1}^-) \succ \boldsymbol{E}_\pi(t_{i+1}^-)$, policy π^* also schedules a customer at the same time as π. Thus $\boldsymbol{E}_{\pi^*}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$ by property 4 of Lemma 5. Since $I_{\pi^*}(t_{i+1}^-) = I_\pi(t_{i+1}^-)$, $I_{\pi^*}(t_{i+1}) = I_\pi(t_{i+1})$ and the customer scheduled under each of these policies are given the same service time, $b_{I_\pi(t_{i+1})}$. Consequently $\boldsymbol{R}_{\pi^*}(t_{i+1}) = \boldsymbol{R}_\pi(t_{i+1})$.

This completes the inductive step. We have shown that $\boldsymbol{E}_{\pi^*}(t) \succ \boldsymbol{E}_\pi(t)$ for $0 \le t$ for a sample path s. It follows that $E[V_N(\pi)|\boldsymbol{S} = s] = E[V_N(\pi^*)|\boldsymbol{S} = s]$, and $E[V_N(\pi)] = E[V_N(\pi^*)]$ for $N = 1, 2, \cdots$. It also follows that $\overline{V}_N(\pi^*) = \overline{V}_N(\pi)$ for $N = 1, 2, \cdots$ and $\overline{V}(\pi^*) = \overline{V}(\pi)$.

The argument that $\overline{V}_t(\pi^*) = \overline{V}_t(\pi)$ requires an additional calculation that accounts for all sample paths s for which $\lim_{n\to\infty} t_n < t$. This is because the induction argument does not cover such sample paths. Define $\Sigma(t) = \{s | \lim_{n\to\infty} t_n > t\}$ and $\overline{\Sigma}(t) = \{s | \lim_{n\to\infty} t_n \le t\}$. We write the expectation of $V_t(\pi)$ as

$$\begin{aligned}
\overline{V}_t(\pi) &= E[V_t(\pi)|s \in \Sigma(t)]P[s \in \Sigma(t)] + E[V_t(\pi)|s \in \overline{\Sigma}(t)]P[s \in \overline{\Sigma}(t)], \\
&= E[V_t(\pi)|s \in \Sigma(t)] + E[V_t(\pi)|s \in \overline{\Sigma}(t)]P[s \in \overline{\Sigma}(t)], \quad \textit{Consequence of A2} \\
&\le E[V_t(\pi)|s \in \Sigma(t)] + \lim_{n\to\infty} nP[\sum_{i=1}^{n} A_i \le t], \\
&= E[V_t(\pi)|s \in \Sigma(t)]. \quad \textit{Consequence of A2}
\end{aligned}$$

As a consequence of this calculation, we can ignore all sample paths belonging to $\overline{\Sigma}(t)$ as they do not contribute to $\overline{V}_t(\pi)$. Thus we conclude that $\overline{V}_t(\pi^*) = \overline{V}_t(\pi)$, $0 < t$.

QED

We complete the section on multiple server queues with deadlines until the beginning of service by proving that STE is the optimal policy out of the class of policies that do not allow a server to remain idle when there are jobs to be served for the G/M/c system. Before we prove this result we first discuss the method with which we will assign service times to jobs. Let $\boldsymbol{B}^{(j)} = \{B_{i,j}\}_{i=1,\cdots}$ $(1 \leq j \leq c)$ be c mutually independent sequences of i.i.d. exponential random variables with parameter μ. Let these sequences be mutually independent. If a customer is assigned to the kth server at time t, then it receives an amount of service equal to $\sum_{l=1}^{m} B_{l,k} - t$ where $m = \min\{i | \sum_{l=1}^{i} B_{l,k} > t\}$. We emphasize that, due to the assumptions on the service times, the service time received by this customer will be exponentially distributed and independent of other events in the system. We redefine $\boldsymbol{S}$ and s to be $\boldsymbol{S} = (\boldsymbol{A}, \boldsymbol{D}, \boldsymbol{B}^{(1)}, \cdots, \boldsymbol{B}^{(c)})$ and $s = (\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b}^{(1)}, \cdots, \boldsymbol{b}^{(c)})$

Theorem 2 *If π is any non-preemptive, non-idling policy, then $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ for $N = 1, \cdots$, $\overline{V}(STE) \geq \overline{V}(\pi)$, and $\overline{V}_t(STE) \geq \overline{V}_t(\pi)$, $0 < t$ for the G/M/c+G system.*

Proof: Define $\boldsymbol{R}_\pi(t) = (r_\pi^{(1)}(t), \cdots, r_\pi^{(c)}(t))$ where $r_\pi^{(j)}(t) = 1$ if server j is busy under π at time t and 0 otherwise. We focus on the behavior of π and STE given $\boldsymbol{S} = s$.

STE exhibits better performance than π provided

- $\boldsymbol{E}_{STE}(t) \succ \boldsymbol{E}_\pi(t)$, for $0 \leq t$,
- $\boldsymbol{R}_{STE}(t) \geq \boldsymbol{R}_\pi(t)$, for $0 \leq t$.

Here the latter inequality is defined componentwise. We prove these inequalities by induction.

We need only focus on the points of time that either a customer arrives, a customer departs, a customer misses a deadline, or a customer is scheduled into service in the systems operating under π and STE. Let $t_0 = 0 \leq t_1 \leq \cdots \leq t_i \leq t_{i+1} \leq \cdots$ denote these times.

It is useful to distinguish among the following events:

$\mathcal{E}_1$ - Arrival of a customer to both systems.

$\mathcal{E}_2$ - Service completion at one or both systems.

$\mathcal{E}_3$ - Loss of one or more customers at one or both systems due to missing of deadline.

$\mathcal{E}_4$ - Scheduling of a customer to service in one or both systems.

A more complete description of the history of both systems is given by the sequence of event-time pairs $(t_0, \sigma_0), (t_1, \sigma_1), \cdots, (t_i, \sigma_i), \cdots, (t_n, \sigma_n)$ where $\sigma_0 = \mathcal{E}_1$ and t_i is the time at which an event of type $\sigma_i \in \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\}$ $(1 \leq i < n)$ occurs. If two types of events occur simultaneously, we represent them as separate events with the identical event times. In some cases the order of the events is determined by the mechanics of the system, i.e., scheduling a server immediately after a departure. Otherwise the order in which they are listed is immaterial.

We note as in Theorem 1 that if $\boldsymbol{E}_{STE}(t_i) \succ \boldsymbol{E}_\pi(t_i)$, $\boldsymbol{R}_{STE}(t_i) \geq \boldsymbol{R}_\pi(t_i)$, and $t_i < t_{i+1}$, then $\boldsymbol{E}_{STE}(t) \succ \boldsymbol{E}_\pi(t)$ and $\boldsymbol{R}_{STE}(t) \geq \boldsymbol{R}_\pi(t)$ for $t_i \leq t < t_{i+1}$.

We proceed with our inductive argument.

Basis Step: The hypothesis is trivially true for $t = t_0$.

Inductive step: Assume that $\boldsymbol{E}_{STE}(t_k) \succ \boldsymbol{E}_\pi(t_k)$, $\boldsymbol{R}_{STE}(t_k) \geq \boldsymbol{R}_\pi(t_k)$ for $k \leq i$. We now show that the relations also hold for $i+1$. There are four cases according to the event type.

Case 1 $(\sigma_{i+1} = \mathcal{E}_1)$: In this case neither $\boldsymbol{R}_\pi$ nor $\boldsymbol{R}_{STE}$ are affected. Furthermore, $\boldsymbol{E}_{STE}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$ as a consequence of property 1 of Lemma 2.

Case 2 $(\sigma_{i+1} = \mathcal{E}_2)$: In this case, neither $\boldsymbol{E}_\pi$ nor $\boldsymbol{E}_{STE}$ are affected. The only way for $\boldsymbol{R}_{STE}(t_{i+1}) \not\geq \boldsymbol{R}_\pi(t_{i+1})$ is if there is a server occupied under both STE and π at t_{i+1}^- and a completion occurs only under STE. However, this is impossible under the rule used to assign service times to jobs. Therefore $\boldsymbol{R}_{STE}(t_{i+1}) \geq \boldsymbol{R}_\pi(t_{i+1})$.

Case 3 $(\sigma_{i+1} = \mathcal{E}_3)$: We have three subcases according to whether the customer misses his deadline under π, STE, or both policies. This event does not affect $\boldsymbol{R}_\pi$ or $\boldsymbol{R}_{STE}$. The proof that $\boldsymbol{E}_{STE}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$ in this case is identical to that provided in theorem 1.

Case 4 $(\sigma_{i+1} = \mathcal{E}_4)$: If there exists at least one zero element in $\boldsymbol{R}_{STE}(t_{i+1}^-)$, say the jth element, then it is also zero in $\boldsymbol{R}_\pi(t_{i+1}^-)$. In this case, both policies schedule this server (unless $\boldsymbol{E}_\pi(t_{i+1}^-) = \emptyset$). Clearly $\boldsymbol{R}_{STE}(t_{i+1}) \geq \boldsymbol{R}_\pi(t_{i+1})$ in this case. Property 2 or 4 from Lemma 2 guarantee $\boldsymbol{E}_{STE}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$. In the case that all servers are busy under STE, then again $\boldsymbol{R}_{STE}(t_{i+1}) \geq \boldsymbol{R}_\pi(t_{i+1})$ and property 3 of Lemma 2 guarantees $\boldsymbol{E}_{STE}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$.

This completes the inductive step. Since we have shown that $\boldsymbol{E}_{STE}(t) \succ \boldsymbol{E}_\pi(t)$ for $0 \leq t$ for any sample path s, it follows that $E[V_N(STE)|\boldsymbol{S} = s] \geq E[V_N(\pi)|\boldsymbol{S} = s]$ and $E[V_N(STE)] \geq E[V_N(\pi)]$ for $N = 1, 2, \cdots$. It also follows that $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ for $N = 1, 2, \cdots$ and $\overline{V}(STE) \geq \overline{V}(\pi)$.
The argument that $\overline{V}_t(STE) \geq \overline{V}_t(\pi),\ 0 < t$ is similar to that given at the end of theorem 1. **QED**

Last, we state an analogous result for the discrete time G/M/c+G queue where service times form an i.i.d. sequence of geometric r.v.'s.

Theorem 3 *If π is any non-preemptive, non-idling policy, then $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ for $N = 1, \cdots$, $\overline{V}(STE) \geq \overline{V}(\pi)$, and $\overline{V}_t(STE) \geq \overline{V}_t(\pi),\ 0 < t$ for the discrete time G/M/c+G system.*

Proof: The argument is identical to the one given for the previous theorem. We note that an alternate induction argument can be based on each discrete time unit rather than on events in order to show the dominance relation.
QED

4 Preemptive Systems with Deadline to End of Service

In this section we show that STE is the best policy for the preemptive continuous time and discrete time G/M/c+G queue when deadlines are to the end of service. We conclude the section by generalizing this result to queues in which servers take vacations. Proof s of these results necessitate the introduction of the notation $\boldsymbol{X}_\pi(t) = (n_\pi(t), \boldsymbol{E}_\pi(t))$ where $n_\pi(t)$ is the number of customers successfully completed by time t. We refer to this as the state of the system at time t under policy π. We introduce the following notion of dominance between states.

Definition 5 *We say that $X_{\pi_1}(t)$ dominates $X_{\pi_2}(t)$ ($X_{\pi_1}(t) \succ X_{\pi_2}(t)$) iff*

1. $n_{\pi_1}(t) \geq n_{\pi_2}(t)$,

2. $E_{\pi_1}(t) \succ Small(E_{\pi_2}(t), |E_{\pi_2}(t)| + n_{\pi_2}(t) - n_{\pi_1}(t))$.

Before we prove the main result of this section we describe some guidelines used to assign customers to servers and service times to customers. First, we restrict ourselves to policies that satisfy the following rules.

- If the number of customers being served at some point in time is $i < c$, then the first i servers are busy.

- If servers i and j are occupied where $i < j$, then the deadlines of the customers assigned to these servers must be in non-decreasing order.

If policy π does not satisfy the above rules, we can always construct a policy π^* that satisfies these rules so that $\overline{V}_N(\pi) = \overline{V}_N(\pi^*)$ for all N, $\overline{V}(\pi) = \overline{V}(\pi^*)$, and $\overline{V}_t(\pi) = \overline{V}_t(\pi^*)$ for all t. There also exists an STE policy that satisfies the above rules.

We now discuss the method by which we will assign service times to jobs. Divide B into $c+1$ sequences, $B^{(j)} = \{B_{i,j}\}_{i=1,\dots}$, $j = 1, 2, \cdots, c+1$. Consider the i-th customer. Let m_i' denote the number of times it is scheduled. Let $s_{i,1}, s_{i,2}, \cdots, s_{i,m_i'}$ be the times at which it is scheduled, $q_{i,1}, q_{i,2}, \cdots, q_{i,m_i'-1}$ be the times at which it is preempted, k_i the identity of the server at which it completes, and $m_i = \min\{j \,|\, \sum_{l=1}^{j} > s_{i,m_i'}\}$. If the i-th customer misses its deadline, then $k_i = 0$. The service time, X_i of the i-th customer is

$$X_i = \begin{cases} \sum_{l=1}^{m_i'-1}(q_{i,l} - s_{i,l}) + \sum_{l=1}^{m_i} B_{l,k} - s_{i,m_i'}, & k_i \neq 0, \\ \sum_{l=1}^{m_i'-1}(q_{i,l} - s_{i,l}) + B_{i,c+1}, & k_i = 0. \end{cases} \tag{1}$$

We claim that the service times received by customers according to this assignment rule are i.i.d. exponential r.v.'s with parameter μ.

Theorem 4 *STE is the optimum preemptive policy for the G/M/c+G queue when the deadlines are to the end of service, i.e.,* $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$, $N > 0$, $\overline{V}(STE) \geq \overline{V}(\pi)$, *and* $\overline{V}_t(STE) \geq \overline{V}_t(\pi)$, $0 < t$.

Proof: The proof is similar to that of earlier theorems and consists of an inductive argument on the times that the following types of events occur,

- $\mathcal{E}_0$ - arrival to both systems,
- $\mathcal{E}_1$ - completion of a job in either or both systems,
- $\mathcal{E}_2$ - job missing deadline under one or both policies,

Let $(t_0, \sigma_0), (t_1, \sigma_1), \cdots$ be the sequence of times and events that occur at those times, i.e., event σ_i occurs at time t_i where $\sigma_i \in \{\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2\}$.

We will demonstrate that $\boldsymbol{X}_{STE}(t) \succ \boldsymbol{X}_\pi(t)$ for every sample $\boldsymbol{S} = s$ and $t \geq 0$ provided that $\boldsymbol{X}_{STE}(0) \succ \boldsymbol{X}_\pi(0)$.

We define $\boldsymbol{R}_\pi(t)$ to be an ordered set of deadlines associated with the customers in service at time t under policy π. As described earlier, we lose no generality in assuming that the deadlines are in nondecreasing order and that if the number of customers, $|\boldsymbol{R}_\pi(t)|$, in service is less than c, the customers occupy the first $|\boldsymbol{R}_\pi(t)|$ servers.

We note as in Theorem 1 that if $\boldsymbol{X}_{STE}(t_i) \succ \boldsymbol{X}_\pi(t_i)$, and $t_i < t_{i+1}$, then $\boldsymbol{X}_{STE}(t) \succ \boldsymbol{X}_\pi(t)$ for $t_i \leq t < t_{i+1}$.

We proceed with our inductive argument.

Basis Step: The hypothesis is trivially true for $t = t_0$.

Inductive step: Assume that $\boldsymbol{X}_{STE}(t_l) \succ \boldsymbol{X}_\pi(t_l)$ for $l \leq i$. We now show that it also holds for $i + 1$. There are three cases according to the type of event.

Case 1 ($\sigma_{i+1} = \mathcal{E}_0$): In this case neither n_π nor n_{STE} are affected. Property 1 of Lemma 2 guarantees that $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_\pi(t_{i+1})$.

Case 2 ($\sigma_{i+1} = \mathcal{E}_1$): There are three subcases according to whether the completion is under π, STE, or both policies. If the completion is under π only, then it occurs on server j where $j > |\boldsymbol{E}_{STE}(t_{i+1}^-)|$. This implies that $|\boldsymbol{E}_\pi(t_{i+1}^-)| > |\boldsymbol{E}_{STE}(t_{i+1}^-)|$ which further implies that $n_{STE}(t_{i+1}^-) > n_\pi(t_{i+1}^-)$. A simple calculation yields that $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_\pi(t_{i+1})$.
If the completion is under STE only, then a simple calculation yields $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_\pi(t_{i+1})$.

If the completion is under both policies, then we have an additional two possiblities. Let the completion occur from the j-th server. Let $r_{STE}^{(j)}$ and $r_{\pi}^{(j)}$ denote the deadlines associated with these customers. If $r_{STE}^{(j)} \leq r_{\pi}^{(j)}$, then property 4 of Lemma 2 ensures that $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_{\pi}(t_{i+1})$. If $r_{STE}^{(j)} > r_{\pi}^{(j)}$, then the ordering that we imposed on $\boldsymbol{R}_{\pi}$ and $\boldsymbol{R}_{STE}$ ensures that we can apply property 5 of Lemma 2 to show $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_{\pi}(t_{i+1})$.

Case 3 ($\sigma_{i+1} = \mathcal{E}_2$): Again there are three subcases according to whether the customer misses his deadline under π, STE, or both policies. If under π, property 3 of Lemma 2 is applicable. If under STE, property 2 of Lemma 2 is applicable. Last, property 4 of Lemma 2 is applicable when the losses occur under both policies.

It follows that $E[V_N(STE)|\boldsymbol{S} = s] \geq E[V_N(\pi)|\boldsymbol{S} = s]$ and $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ for $N = 1, 2, \cdots$ and $\overline{V}(STE) \geq \overline{V}(\pi)$. The argument that $\overline{V}_t(STE) \geq \overline{V}_t(\pi)$ is similar again to that given at the end of Theorem 1.

QED

A similar result also can be proven for the discrete time bulk arrival G/M/c+G queue. Here the service time consists of an integer number of time units that is given by a geometric r.v. This model is of particular use in data communications in the case that the service time is always a single time unit. It forms the basis of most models of statistical multiplexers.

Theorem 5 *The STE policy is optimal for the discrete time G/M/c+G queue.*

Proof: The proof is similar to the one given for Theorem 4 and is omitted here.

QED

Remark. In the case that customers require a single time unit of service, there is no distinction between preemptive and non-preemptive systems. Furthermore, there is no distinction between systems in which customers must meet their deadlines either by the time service begins or by the time service completes.

We conclude this section with a generalization of Theorem 4 to include systems in which servers take vacations. This is of interest for at least two reasons. First, processors in any multiprocessor system are prone to failures. Second, systems in which servers take vacations can be used to model real-time systems with two classes of customers. For example, one class of tasks may be unable to tolerate missed deadlines. The second class of jobs may be able to tolerate some missed deadlines. If the tasks in the first class are well understood (i.e., known service times, arrival times), they can be given higher priority than the second class of tasks and scheduled independently of the second class. The second class of tasks are like the customers that we have considered in our model for which the object is to develop policies that will minimize the fraction of tasks that miss their deadlines. Thus tasks in the second class see a system where servers take vacations.

Let $\{U_{i,j}, W_{i,j}\}_{i=1,\cdots}$, $j = 1, 2, \cdots, c$ be families of r.v.'s such that $U_{i,j}$ is the length of the i-th time interval during which the j-th server is available for service and $W_{i,j}$ is the length of the i-th time interval during which the j-th server is on vacation (unavailable for service). We allow these sequences of r.v.'s to have arbitrary statistics so long as they are independent of $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{D}$. In this case we state the following result.

Theorem 6 *STE is the optimum policy for both the continuous and discrete time G/M/c+G queue with vacations when the deadlines are to the end of service, i.e.,* $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$, N 0, $\overline{V}(STE) \geq \overline{V}(\pi)$, *and* $\overline{V}_t(STE) \geq \overline{V}_t(\pi)$, $0 < t$.

Proof. The proof is similar to that given for Theorem 4 and is omitted here.

5 Non-Preemptive Systems with Deadline to End of Service

In this section we show that STE is the best policy from the class of non-idling policies for the non-preemptive G/M/c+G queue when deadlines are to the end of service. Furthermore, we show that there exists an STEI policy that provides performance better or equal to that of any non- STEI policy for the G/M/c+G queue. In both cases, we will use the "$\succ$" relation. In addition to using the properties found in Lemma 2 we will also require the following result.

Lemma 3 *Let R and S be a set of non-negative real numbers such that $R \succ S$. Let R and S be expressed as $R = R_1 + R_2$ and $S = S_1 + S_2$ such that $R_1 \succ S_1$ and $|R_2| = n$, $|S_2| = n'$ with $n \geq n'$. If we express R_2 and S_2 as $R_2 = (x_1, x_2, \cdots, x_n)$ and $S_2 = (y_1, y_2, \cdots, y_{n'})$ where $x_i \geq x_{i+1}$ and $y_i \geq y_{i+1}$, $i = 1, \cdots n-1$, then $R - \{x_i\} \succ S - \{y_i\}$ for $i = 1, \cdots, n'$.*

Proof: The argument is by induction on n, the cardinality of R_2. We first observe that the case $n > n'$ can be reduced to the case $n = n'$ by simply inserting $n - n'$ zero elements into R, S, R_1, and S_2. Thus we assume that $n = n'$.
Basis Step. When $n = 1$ and $S_2 = \emptyset$, the Lemma is trivially true. When $S_2 = \{y_1\}$, then $R - \{x_1\} = R_1 \succ S_1 = S - \{y_1\}$.
Inductive step. Assume that the Lemma is true for $|R_2| \leq n$. We now establish it for $|R_2| = n + 1$.There are three subcases according to the number of elements x_i in R_2 such that $x_i > y_i \in S_2$. If the number is zero, then according to property 4 of Lemma 2 $R - \{x_i\} \succ S - \{y_i\}$ for $1 \leq i \leq n'$. Consider the case that the number is two or more. Let x_i and y_i be elements such that $x_i > y_i$. Define $R_1' = R_1 + \{x_i\}$, $S_1' = S_1 + \{y_i\}$, $R_2' = R_2 - \{x_i\}$, and $S_2' = S_2 - \{y_i\}$. We have $R = R_1' + R_2'$, $S = S_1' + S_2'$. Since $x_i > y_i$, we also have $R_1' \succ S_1'$. Thus we can apply the inductive hypothesis to show that $R - \{x_j\} \succ S - \{y_j\}$ for $j \neq i$. Since there are two elements for which $x_i > y_i$, we can extend it to $j = i$.

We now consider the case where there is only one element in R_2 such that $x_i > y_i$. Let $r_1, r_2, \cdots, r_m$ denote the elements in R in non-increasing order and $s_1, s_2, \cdots, s_{m'}$ denote the elements in S also in non-increasing order. Here $m = |R|$ and $m' = |S|$. Let $\{k_1, k_2, \cdots, k_n\}$ and $\{j_1, j_2, \cdots, j_n\}$ be sets of integers such that $r_{k_l} = x_l$ and $s_{j_l} = y_l$, $l = 1, \cdots n$. Note that $x_i > y_i$ implies that $k_{i-1} + 1 < j_i \leq m' - (n - i)$. If $j_i \geq k_i$, then property 5 of Lemma 2 can be applied to yield $R - \{x_i\} \succ S - \{y_i\}$. Let us now consider the case $k_{i-1} + 1 < j_i < k_i$. The sets $R - \{x_i\}$ and $S - \{y_i\}$ can be expressed as $\{r_1', r_2', \cdots, r_{m-1}'\}$ and $\{s_1', s_2', \cdots s_{m'-1}'\}$ where

$$r_l' = \begin{cases} r_l, & 1 \leq l < k_i, \\ r_{l+1}, & k_i \leq l < m, \end{cases}$$

$$s_l' = \begin{cases} s_l, & 1 \leq l < j_i, \\ s_{l+1}, & j_i \leq l < m'. \end{cases}$$

Since $R \succ S$, it follows that $r_l' \geq s_l'$ when $1 \leq l \leq j_i - 1$ and when $k_i \leq l \leq m' - 1$ In addition $r_l' \geq s_l'$ when $k_i + 1 \leq l \leq j_i$ because $R_1 \succ S_1$ and $j_i > k_{i-1} + 1$. Therefore, we have shown that $R - \{x_i\} \succ S - \{y_i\}$ for this case. The relation $R - \{x_j\} \succ S - \{y_j\}$, $j \neq i$ follows from $x_j < y_j$ and property 5 of Lemma 2.
QED

Consider a policy π that is allowed to preempt a customer solely to move him to another server. We refer to this as a *limited preemption* policy and claim that the performance of this policy does not differ from a policy that uses the same scheduling rules except that it does not allow preemptions. We will find it easier to work with these limited preemption policies. Specifically, we consider limited preemption policies that enforce the following rules:

- If the number of customers in service, n is less than the number of servers, then they are placed on the first n servers.
- Customers are placed on servers such that the deadline associated with the customer on the i-th server is greater than or equal to that associated with the customer on the $(i+1)$-th server.

Customers are assigned service times according to the same rule used in analyzing the system that allows preemptions (see section 4).

Before we introduce our results, we find it useful to introduce the notation $\boldsymbol{R}_\pi(t)$ to be the set of deadlines associated with *all customers in the system.* The set of deadlines associated with the customers in service can be expressed as $\boldsymbol{R}_\pi(t) - \boldsymbol{E}_\pi(t)$.

Theorem 7 *STE provides the best performance of all non-preemptive, non-idling policies for the G/M/c+G queue when the deadlines are to the end of service, i.e.,* $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ $(1 \leq N)$, $\overline{V}(STE) \geq \overline{V}(\pi)$, *and* $\overline{V}_t(STE) \geq \overline{V}_t(\pi)$ $(0 < t)$.

Proof: The proof of our theorem is similar to that of earlier theorems and consists of showing that $\boldsymbol{E}_{STE}(t) \succ \boldsymbol{E}_\pi(t)$ and $\boldsymbol{R}_{STE}(t) \succ \boldsymbol{R}_\pi(t)$ for every sample path $\boldsymbol{S} = s$ by induction on the times of important events. We define the following events

- $\mathcal{E}_0$ - arrival to both systems,
- $\mathcal{E}_1$ - completion of a job in either or both systems,
- $\mathcal{E}_2$ - job missing deadline under one or both policies,

Let $(t_0, \sigma_0), (t_1, \sigma_1), \cdots$ be the sequence of times and events that occur at those times, i.e., event σ_i occurs at time t_i where $\sigma_i \in \{\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2\}$.

We note as in Theorem 1 that if $\boldsymbol{E}_{STE}(t_i) \succ \boldsymbol{E}_\pi(t_i)$ and $\boldsymbol{R}_{STE}(t_i) \succ \boldsymbol{R}_\pi(t_i)$ and $t_i < t_{i+1}$, then $\boldsymbol{E}_{STE}(t) \succ \boldsymbol{E}_\pi(t)$ and $\boldsymbol{R}_{STE}(t) \succ \boldsymbol{R}_\pi(t)$ for $t_i \leq t < t_{i+1}$.

We proceed with our inductive argument.

Basis Step: The hypothesis is trivially true for $t = t_0$.

Inductive step: Assume that $\boldsymbol{E}_{STE}(t_l) \succ \boldsymbol{E}_\pi(t_l)$ and $\boldsymbol{R}_{STE}(t_l) \succ \boldsymbol{R}_\pi(t_l)$ for $l \leq i$. We now show that it also holds for $i+1$. There are three cases according to the type of event.

Case 1 ($\sigma_{i+1} = \mathcal{E}_0$): The relationship can be shown to hold as a consequence of property 1 of Lemma 2.

Case 2 ($\sigma_{i+1} = \mathcal{E}_1$): There are two subcases according to whether the completion is under STE or both policies. (Note: according to the inductive hypothesis and the server assignment rule, a completion under π implies a completion under STE.) If the completion is under STE only, then $\boldsymbol{E}_\pi(t_{i+1}) = \emptyset$ which implies that $\boldsymbol{E}_{STE}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$. Because of the way that customers are assigned to servers, the deadline of the completed customer cannot reside in $Large(\boldsymbol{R}_{STE}(t_i^-), |\boldsymbol{R}_\pi(t_i^-)|)$. Consequently $\boldsymbol{R}_{STE}(t_{i+1}) \succ \boldsymbol{R}_\pi(t_{i+1})$. If the completion is under both policies, then Lemma 3 and the inductive hypothesis ensure that $\boldsymbol{R}_{STE}(t_{i+1}) \succ \boldsymbol{R}_\pi(t_{i+1})$. The inductive hypothesis and the fact that STE will schedule the customer with the smallest deadline from $C_{STE}(t_i^-)$ ensures that property 5 of Lemma 2 can be applied to show that $\boldsymbol{E}_{STE}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$.

Case 3 ($\sigma_{i+1} = \mathcal{E}_2$): Again there are three subcases according to whether the customer misses his deadline under π, STE, or both policies. If under π, property 3 of Lemma 2 is applicable. If under STE, property 2 of Lemma 2 is applicable. If under both STE and π, then we have further subcases according to whether the customers were in service or in the queue. In all of these cases, the result is obtained by using either property 4 or 5 from Lemma 2.

It follows that $E[V_N(STE)|\boldsymbol{S} = s] \geq E[V_N(\pi)|\boldsymbol{S} = s]$ and $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ for $N = 1, 2, \cdots$ and $\overline{V}(STE) \geq \overline{V}(\pi)$. Arguments similar to those given in previous theorems can be made to show $\overline{V}_t(STE) \geq \overline{V}_t(\pi)$, for $0 < t$.

QED

Let us consider now policies that may permit idle processors. We state and prove the following result.

Theorem 8 *For any arbitrary policy π, there exists an STEI policy π^* such that $\overline{V}_N(\pi^*) \geq \overline{V}_N(\pi)$, $\overline{V}(\pi^*) \geq \overline{V}(\pi)$, and $\overline{V}_t(\pi^*) \geq \overline{V}_t(\pi)$, $t > 0$ for the $G/M/c+G$*

queue with no preemptions when the deadline is to end of service.

Proof. Consider any policy π not in the class of STEI policies. We construct an STEI policy π^* that exhibits equal or better performance as that of π. The rules for constructing π^* differ from the rules given in Theorem 1 in the following way.

- In addition to maintaining an ordered list of customers at time t, $\mathcal{A}(t)$, that would be eligible under π, policy π^* also maintains an ordered list $\mathcal{R}(t)$ of all customers in the system at time t under π.
- Rule 3(a) is modified to read: If at time t $|\mathcal{R}(t)| - |\mathcal{A}(t)| = |\boldsymbol{R}_{\pi^*}(t)| - |\mathcal{C}_{\pi^*}(t)|$ then π^* schedules the customer closest to its deadline with probability $1 - p_0(\pi, t, H_t')$.
- Add a new rule to account for changes to $\mathcal{R}(t)$. It is

    6. π^* modifies $\mathcal{R}(t)$ as follows,

        (a) customer c is removed from $\mathcal{R}(t)$ either when its deadline expires or it corresponds to a customer in $\boldsymbol{R}_{\pi^*}(t)$ that completes service.

        (b) customer c is added to $\mathcal{R}(t)$ when it arrives to the system.

We focus on the behavior of π and π^* given $\boldsymbol{S} = s$. We can show that $\boldsymbol{E}_{\pi^*}(t) \succ \boldsymbol{E}_{\pi}(t)$ and $\boldsymbol{R}_{\pi^*}(t) \succ \boldsymbol{R}_{\pi}(t)$ for $0 \leq t$ using the same method of proof as used in Theorem 7. This has as its consequence that $\overline{V}_N(\pi^*) \geq \overline{V}_N(\pi)$ for $N \geq 0$ and $\overline{V}(\pi^*) \geq \overline{V}(\pi)$.
QED

Similar results can be proven for discrete time counterparts. They are

Theorem 9 *STE provides the best performance of all non-preemptive, non-idling policies for the discrete time G/M/c+G queue when the deadlines are to the end of service, i.e.,* $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ $(1 \leq N)$, $\overline{V}(STE) \geq \overline{V}(\pi)$, *and* $\overline{V}_t(STE) \geq \overline{V}_t(\pi)$ $(t = 1, 2, \cdots)$.

Theorem 10 *For any arbitrary policy π, there exists an STEI policy π^* such that* $\overline{V}_N(\pi^*) \geq \overline{V}_N(\pi)$, $\overline{V}(\pi^*) \geq \overline{V}(\pi)$ *and* $\overline{V}_t(\pi^*) \geq \overline{V}_t(\pi)$, $t = 1, 2, \cdots$ *for the discrete time G/M/c+G queue with no preemptions when the deadline is to end of service.*

6 Summary

We have shown that the best scheduling policies for minimizing the fraction of customers missing a deadline in many multiple server queues belong to the class of STEI policies. Furthermore, STE is the optimal policy out of the class of policies that do not allow inserted idle times for the nonpreemptive G/M/c queue. Last, if deadlines are to the end of service and preemptions are allowed, then the best policy for the G/M/c queue is STE. This result hold in the case that servers take vacations.

References

[1] F. Baccelli, P. Boyer, and G. Hebuterne, "Single-Server Queues with Impatient Customers" *Adv. Appl. Prob.*, Vol. 16, pp. 887-905, 1984.

[2] P. Bhattacharya and A. Ephremides, "Scheduling of Time-Critical Messages," *Proc. of 27-th CDC Conf.*, Dec. 1987.

[3] E.G. Coffman (Ed.), *Computer and Job-Shop Scheduling Theory*, John Wiley, New York, 1976.

[4] J.W. Cohen, "Single Server Queues with Restricted Accessibility," *J. Eng. Math.*, Vol. 3, 4, pp. 265-284, Oct. 1969.

[5] R.W. Conway, W.L. Maxwell and L.W. Miller, *Theory of Scheduling*, Addison-Wesley, Reading, Mass., 1967.

[6] M. Dertouzos, "Control Robotics: The Procedural Control of Physical Processes," *Proceedings of the IFIP Congress*, pp. 807-813, 1974.

[7] B. Gavish and P.J. Schweitzer, "The Markovian Queue with Bounded Waiting Time," *Management Sci.*, Vo. 21, 12, pp. 1349-1357, Aug. 1972.

[8] H. Heffes and D.M. Lucantoni, "A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance," *IEEE J. Selected Areas of Communications*, Vol. 4, 6, pp. 856-868, 1986.

[9] J.R. Jackson, "Scheduling a Production Line to Minimize Maximum Tardiness," Research Report 43, Management Science Research Report, UCLA, 1955.

[10] C. Liu and J. Layland, "Scheduling Algorithms for Multi-Programming in a Hard-Real-Time Environment," *J. ACM*, Vol. 20, pp. 46-61, 1973.

[11] A. Mok and M. Dertouzos, "Multiprocessor Scheduling in a Hard Real-Time Environment," *Proc. 7th Texas Conf. on Comp. Syst.*, Nov. 1978.

[12] J.M. Moore, "An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," *Management Science*, vol. 15, pp. 102-109, 1968.

[13] S.S. Panwar, *Time Constrained and Multi-access Communications*, Ph.D. Thesis, Department of Electrical and Computer Electrical Engineering, University of Massachusetts, Amherst, Feb. 1986.

[14] S.S. Panwar, D. Towsley, and J.K. Wolf, "Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service," to appear in *JACM*, October 1988.

[15] W.P. Pierskalla and C. Roach, "Optimal Issuing Policies for Perishable Inventory," *Management Science*, vol. 18, pp. 603-614, 1972.

[16] M. Pinedo, "Stochastic Scheduling with Release Dates and Due Dates," *Operations Research*, vol. 31, pp. 559-572, 1983.

[17] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison Wesley, 1987.

[18] Zaw-Sing Su and Kenneth C. Sevcik, "A Combinatorial Approach to Scheduling Problems," *Operations Research*, vol. 26, pp. 836-844, 1978.