数理解析研究所講究録
第596巻 1986年 194-211

Realizable Performance Requirements of Finite-Source Queues

Hisao Kameda

亀田壽夫

Department of Computer Science
The University of Electro-Communications
Chofu, Tokyo, Japan

1. Introduction

To understand the waiting time performance of time-shared systems, we would like to analyze suitable queueing network models that embody various scheduling strategies and that accurately reflect the structure of real systems. With the current state of knowledge, however, it appears that such queueing network models are difficult to solve analytically, since they usually do not allow product-form solutions. (On the contrary, there exists an accumulation of results concerning the waiting time performance in infinite-source queueing models [6, 10, 11]; however, the infinite-source assumption is said to be sometimes unrealistic [2].) Therefore, as a second best alternative, we study a finite-source queueing model (or sometimes called 'finite population' [11], machine interference' [5], or 'machine repairman' model); we study the model with the hope that solid understanding of the properties of the model will provide a basis for the comprehension of more general environments.

2. The model and the basic properties

The model studied here is a closed cyclic queueing network which contains two service stations and an arbitrary but finite number, N, of jobs, 1, 2, ..., N. The one service station has only one server ('processor') where jobs may suffer queueing delays. The other service station has multiple servers ('terminals'), each of which has one-to-one correspondence with a job, where no job suffers queueing delays. Assume that the mean service time for job j are $1/u_j$ on the processor and $1/v$ on the terminals, for j = 1, 2, ..., N. The service time distributions are assumed to be exponential in Sections 2, 3 and 4.1.

Performance vectors Let T_j denote the average response time for job j, j = 1, 2, ..., N; the average response time means the steady-state average of the time period between the instant job j arrives at the processor and the instant job j leaves the processor. Let **T** denote response time vector $(T_1, T_2, \ldots, T_N)$.

Let U_j denote the utilization factor of the processor for job j, j = 1, 2, ..., N ($\Sigma_{j=1}^{N} U_j = U$, where U denotes the utilization factor of the processor). Let **U** denote utilization vector $(U_1, U_2, \ldots, U_N)$. Naturally we have

$$U_j = 1/[u_j(T_j + 1/v)], \quad j = 1, 2, \ldots, N. \tag{2.1}$$

Both **T** and **U** are performance vectors. Hereafter, however, we mostly refer to **U** as the performance vector of the system.

We are interested in studying the system performance as a function of the processor scheduling strategy. If, for a given scheduling strategy S, the value of performance vector is **U**, we say that S achieves **U**. A given performance vector **U** is said to be achievable if there exists a

scheduling strategy S that achieves **U**.

Scheduling strategies considered We now define the class of scheduling strategies to be used at the processor. We impose the following two restrictions on the scheduling strategies (see Kameda [9]):

(i) The scheduling strategies are work conserving.

(ii) The scheduling strategies use only information about the current state and the past of the queueing process in making scheduling decisions.

These conditions are satisfied by most scheduling strategies used in practice, e.g. first come first served, preemptive and nonpreemptive priority, longest and shortest expected remaining processing time first, preemptive and nonpreemptive last come first served, processor sharing, generalized processor sharing, etc.

Mixing strategies Suppose we are given a number of strategies S_1, S_2, ..., S_k. Consider a new strategy which, at the beginning of each busy period of the processor, decides with probability p_i that sequencing decisions at the processor during that busy period are to be made according to S_i, for $i = 1, 2, \ldots, k$ ($\Sigma_{i=1}^{k} p_i = 1$). We call such a strategy the mixing strategy for S_1, S_2, ..., S_k with parameters p_1, p_2, ..., p_{k-1}.

Note that the class of scheduling strategies we have already defined above is closed under mixing operation. Hence, the mixing strategies possess all of the properties implied by the assumptions on scheduling strategies.

Basic properties of the model Now, we present basic properties of the

model. We define $U(\mathbf{M})$ as follows:

$$U(\mathbf{M}) = 1 - 1/(\Sigma_{n=0}^{M} n! \; \Sigma_{\mathbf{I}\subset\mathbf{M}, I=n} \Pi_{j\in\mathbf{I}} r_j) \tag{2.2}$$

where $\mathbf{M}$ and $\mathbf{I}$ denote arbitrary sets of job indices, M denotes the size of $\mathbf{M}$, I denotes the size of $\mathbf{I}$, and

$$r_j = v/u_j. \tag{2.3}$$

Furthermore, note that the set of indices of all jobs in the model is $\mathbf{N}$. The following three properties concerning the model have already been derived by Kameda [7, 9].

Property A1. In the finite-source queueing model, a given utilization vector $\mathbf{U}$ is achivable by some scheduling strategy, if and only if $\mathbf{U}$ satisfies the following condition:

$$\Sigma_{j\in\mathbf{N}} U_j = U(\mathbf{N}), \text{ and} \tag{2.4}$$

$$\Sigma_{j\in\mathbf{Z}} U_j \leq U(\mathbf{Z}) \quad \text{for any non-empty proper subset } \mathbf{Z} \text{ of } \mathbf{N}, \tag{2.5}$$

where $U(\mathbf{N})$ and $U(\mathbf{Z})$ are given by (2.2).

Property A2. Assume that job j is associated with a weight factor C_j, for j = 1, 2, ..., N. The sum $C = \Sigma_{j\in\mathbf{N}} C_j U_j$ is maximum, if and only if job j has preemptive priority over all other jobs that have the weight factor lower than C_j, for j = 1, 2, ..., N.

Property A3. Consider an arbitrary utilization vector $\mathbf{U}$ which satisfies (2.4) and, for all non-empty proper subsets of $\mathbf{Z}$, (2.5). $\mathbf{U}$ can be achieved by a mixing preemptive priority strategy. The condition for a response time vector $\mathbf{T}$ to be achievable is obtained directly from the above and (2.1).

Having determined, for example, that $\mathbf{U}$ is achievable, there remains

the question of what scheduling strategy should be used to achieve it. One answer is provided by mixing strategies. The strategies to be mixed and the parameters of the mix can be determined using standard linear programming methods just like those shown in Coffman and Mitrani [4], as follows.

Let us name the performance vectors of the preemptive priority disciplines $\mathbf{P}_1$, $\mathbf{P}_2$, ..., $\mathbf{P}_{N!}$ and reformulate the problem as follows: Find N! non-negative numbers p_1, p_2, ..., $p_{N!}$ all but N of which are equal to zero, such that

$$\Sigma_{i=1}^{N!} p_i = 1 \text{ and } \Sigma_{i=1}^{N!} p_i \mathbf{P}_i = \mathbf{U}. \tag{2.6}$$

We have N + 1 linear constraints, N of which are independent (the vectors $\mathbf{P}_i$ and $\mathbf{U}$ have only N - 1 independent elements because of the utilization conservation law (2.4)), to which we wish to find a non-negative solution such that at most N of the variables are nonzero. This can be solved by introducing N artificial variables q_0 and $\mathbf{q} = (q_1, q_2, \ldots, q_{N-1})$ and solving the linear program

$$\max \Sigma_{i=1}^{N!} p_i$$

subject to the constraints $p_i \geq 0$ (i = 1, 2, ..., N!), $q_0 \geq 0$, $\mathbf{q} \geq 0$, $q_0 + \Sigma_{i=1}^{N!} p_i = 1$ and $\mathbf{q} + \Sigma_{i=1}^{N!} p_i \mathbf{P}_i = \mathbf{U}$ (using only the first N - 1 elements of $\mathbf{P}_i$ and $\mathbf{U}$) by the simplex method with the initial basis $p_i = 0$ (i = 1, 2, ..., N!), $q_0 = 1$ and $\mathbf{q} = \mathbf{U}$. When an objective value of 1 is reached (as we know it will be, if $\mathbf{U}$ is achievable), the corresponding p_i's define a mixing strategy which achieves $\mathbf{U}$.

In the case where it is known that the equality holds in some of the constraints, the total number of necessary and sufficient constraints is greatly decreased as will be shown in Theorem 1.

Theorem 1. Suppose that the equality in (2.5) is known to hold for an arbitrary number, p, of distinct non-empty proper subsets of set **N**. Then we can describe the subsets as $\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_p$ such that $\mathbf{Z}_0 \subset \mathbf{Z}_1 \subset \ldots \subset \mathbf{Z}_p \subset \mathbf{Z}_{p+1}$ where $\mathbf{Z}_0$ and $\mathbf{Z}_{p+1}$ denote an empty set and set **N**, respectively. Then, the performance vector $\mathbf{U} = (U_1, U_2, \ldots, U_N)$ is realizable if and only if

$$\Sigma_{j \in \mathbf{Z}_i - \mathbf{Z}_{i-1}} U_j = U(\mathbf{Z}_i) - U(\mathbf{Z}_{i-1}), \quad i = 1, 2, \ldots, p+1. \tag{2.7}$$

$$\Sigma_{j \in \mathbf{Z}} U_j \leq U(\mathbf{Z}_{i-1} \cup \mathbf{Z}) - U(\mathbf{Z}_{i-1}), \quad \text{for any non-empty proper subset } \mathbf{Z} \text{ of } \mathbf{Z}_i - \mathbf{Z}_{i-1}, \; i = 1, 2, \ldots, p+1. \tag{2.8}$$

(Proof. Omitted.)

Remark Let n_i be the size of set $\mathbf{Z}_i - \mathbf{Z}_{i-1}$ in the above proposition, for $i = 1, 2, \ldots, p+1$. Then the total number of constraints, in (2.7) and (2.8), is $\Sigma_{i=1}^{p+1}(2^{n_i} - 2) + p + 1 = \Sigma_{i=1}^{p+1} 2^{n_i} - p - 1$ whereas the number of the original contraints (2.4) and (2.5) is $2^{\Sigma_{i=1}^{p+1} n_i} - 1$. Thus we see that the total number of constraints in (2.7) and (2.8) is much less than that of constraints in (2.4) and (2.5).

3. Optimization of performance measures

We usually think of performance measures each of which is a function of a performance vector, e.g. the (overall) mean response time, T^* [as we will see later, $T^* = N/(\Sigma_{j=1}^{N} u_j U_j) - 1/v$]; we often wish one of the performance measures to be optimized. Here we consider some performance measures, each of which is expressed as $f(U_1, U_2, \ldots, U_N)$. Therefore, the optimization problem of the performance measure is expressed as

$$\min A = f(U_1, U_2, \ldots, U_N)$$

subject to the constraints:

$$\Sigma_{j \in \mathbf{N}} U_j = U(\mathbf{N}),$$

$$\Sigma_{j \in \mathbf{Z}} U_j \leq U(\mathbf{Z}) \text{ for every non-empty proper subset } \mathbf{Z} \text{ of } \mathbf{N}.$$

Note that the above is a linearly constrained mathematical programming problem (see, for example, Shapiro [14]).

3.1 Linear performance measures

Consider the case where the function $f(U_1, U_2, \ldots, U_N)$ is linear in each U_j, that is, of the form $\Sigma_{j \in \mathbf{N}} w_i U_i$ with w_i independent of U_i. Then the above optimization problem is reduced to a linear programming problem. We can easily see that the optimal measure is achieved by a preemptive priority discipline as we see directly from property A2.

Some examples are given in the following. Note that interaction means an event that begins when a job arrives at the processor, and continues until the job leaves the processor.

(I-a) Average number of interactions per unit time interval (i.e. throughput), $\tilde{T}$:

$$\tilde{T} = \Sigma_{j=1}^{N} u_j U_j.$$

$\tilde{T}$ is maximized by the discipline whereby a higher preemptive priority is given to a job that has a greater amount of u_j. Note that $\tilde{T}$ is a special case of the following measure.

(I-b) Total average production value per unit time interval, V:

We consider that each interaction of job j is associated with a production value C_j. Then, V is defined as:

$$V = \Sigma_{j=1}^{N} C_j u_j U_j.$$

V is maximized by the discipline whereby a higher preemptive priority is given to a job that has a greater amount of $C_j u_j$.

(I-c) Mean response time, T*:

Job j experiences average response time T_j, and the average frequency of interactions for job j is $u_j U_j$.

Therefore, T* is expressed as:

$$T^* = \Sigma_{j=1}^{N} u_j U_j T_j / \Sigma_{j=1}^{N} u_j U_j = N/\tilde{T} - 1/v.$$

Thus, T* is minimized by the discipline which maximizes $\tilde{T}$ as in (I-a).

(I-d) Overall resource utilization, U*:

Consider the case where the cost of each terminal (including human labor associated with it) is not identical. In that case we may wish terminals of higher cost to be more utilized so that the overall resource utilization U*, may be maximum. Let w_0 denote the cost of the processor; let w_j denote the cost of the terminal associated with job j, j = 1, 2, ..., N. Then, U* is defined as:

$$U^* = \Sigma_{j=1}^{N} U_j (w_0 + w_j u_j / v).$$

U* is maximized by the discipline wherby a higher preemptive priority is given to job that has a greater amount of $w_j u_j$. Note that the discipline reduces to what optimizes $\tilde{T}$ and T* if $w_i = w_j$ for all i, j.

3.2 Nonlinear performance measures

Consider the case where the function $f(U_1, U_2, \ldots, U_N)$ is not linear in each U_j. Then the optimization problem is that of a linearly constrained nonlinear programming. (The schedule that optimizes the measure is not necessarily a preemptive priority discipline.) Once we have found a performance vector which is a solution of the nonlinear programming problem, we then can find a scheduling strategy which optimizes the performance measure, for example, by the method mentioned in Section 2.

Consider the case where the function $f(U_1, U_2, \ldots, U_N)$ is convex. Then the problem is a convex programming and it is known that a local optimum solution is the global optimum solution. Futhermore, if f is differentiable with respect to all U_j's, the optimal solution is characterized by the following Kuhn-Tucker condition (see, for example, Shapiro [14]):

$$\partial f(U_1, U_2, \ldots, U_N)/\partial U_j + y_{\mathbf{N}} + \Sigma_{\mathbf{Z}\subset\mathbf{N},\ Z\geq 1,\ j\in\mathbf{Z}}\ y_{\mathbf{Z}} = 0,\quad j\in\mathbf{N},$$

$$\Sigma_{j\in\mathbf{N}}\ U_j = U(\mathbf{N}),$$

$$y_{\mathbf{Z}} \geq 0,\ \Sigma_{j\in\mathbf{Z}}\ U_j \leq U(\mathbf{Z}),\ y_{\mathbf{Z}}[\Sigma_{j\in\mathbf{Z}}\ U_j - U(\mathbf{Z})] = 0$$

for every non-empty proper subset $\mathbf{Z}$ of $\mathbf{N}$,

where $y_{\mathbf{N}}$ and $y_{\mathbf{Z}}$ denote Lagrange multipliers.

Some examples are given in the following.

(II-a) Mean of the average response time for each job, $\bar{T}$:

$$\bar{T} = \Sigma_{j=1}^{N}\ T_j/N.$$

$\bar{T}$ is a special case of the following measure.

(II-b) Average response cost, C*:

$$C^* = \Sigma_{j=1}^{N}\ c_j T_j$$

where c_j is the cost associated with the unit length of the average

response time of job j, j = 1, 2, ..., N. By using (2.1),

$$C^* = \Sigma_{j=1}^{N} c_j/u_j U_j - \Sigma_{j=1}^{N} c_j/v.$$

Let a_j denote c_j/u_j for the sake of simplicity. Thus we have a problem of minimizing:

$$A = \Sigma_{j=1}^{N} a_j/U_j$$

under the constraints (2.4) and (2.6). Note that this is a separable, convex, linearly constrained non-linear programming problem.

The solution of this is characterized as follows: Consider that the equality holds in some of the constraints (2.6) for $\mathbf{Z} = \mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_p$ ($\mathbf{Z}_1 \subset \mathbf{Z}_2 \subset \ldots \subset \mathbf{Z}_p$) when $\mathbf{U}$ is an optimal solution. (see Lemma 3 of Kameda [9].) For an arbitrary $j \in \mathbf{Z}_k - \mathbf{Z}_{k-1}$ we have from (3.1) and (3.2),

$$-a_j/U_j^2 + y_{\mathbf{N}} + \Sigma_{i=k}^{p} y_{\mathbf{Z}_i} = 0.$$

Therefore

$$U_j = [a_j/(y_{\mathbf{N}} + \Sigma_{i=k}^{p} y_{\mathbf{Z}_i})]^{1/2}.$$

Thus from (2.8) in Theorem 1,

$$(y_{\mathbf{N}} + \Sigma_{i=k}^{p} y_{\mathbf{Z}_i})^{-1/2} = [U(\mathbf{Z}_k) - U(\mathbf{Z}_{k-1})]/(\Sigma_{j\in\mathbf{Z}_k-\mathbf{Z}_{k-1}} (a_j)^{1/2}). \quad (3.3)$$

Therefore

$$U_j = (U(\mathbf{Z}_k) - U(\mathbf{Z}_{k-1}))(a_j)^{1/2}/[\Sigma_{j\in\mathbf{Z}_k-\mathbf{Z}_{k-1}} (a_j)^{1/2}]. \quad (3.4)$$

By noting, from (3.2), that $y_{\mathbf{Z}_i}$, i = 1, 2, ..., p, are nonnegative, we see that A is minimized by such U_j's as satisfy (3.4) if

$$(U(\mathbf{Z}_k) - U(\mathbf{Z}_{k-1}))/[\Sigma_{j\in\mathbf{Z}_k-\mathbf{Z}_{k-1}} (a_i)^{1/2}]$$
$$\leq (U(\mathbf{Z}_{k+1}) - U(\mathbf{Z}_k))/[\Sigma_{j\in\mathbf{Z}_{k+1}-\mathbf{Z}_k} (a_i)^{1/2}], \quad (3.5)$$

for k = 1, 2, ..., p (by nothing (3.3)),

and (2.9) hold. Therefore, if we find such a set of $\mathbf{Z}_i$'s as satisty (3.5), we can obtain an optimal solution for each U_j as (3.4). By letting c_j be 1/N, from the above, we can obtain $\mathbf{U}$ which optimizes $\bar{T}$.

(II-c) Average response ratio, R*:

$$R^* = (\Sigma_{j=1}^{N} u_j T_j u_j U_j)/(\Sigma_{j=1}^{N} u_j U_j).$$

Note that $u_j T_j$ is the response ratio for job j, and that R* is a special case of the following performance measure, $\bar{C}$.

(II-d) Average cost per interaction, $\bar{C}$:

$$\bar{C} = (\Sigma_{j=1}^{N} c_j T_j u_j U_j)/(\Sigma_{j=1}^{N} u_j U_j),$$

where c_j is defined as in (II-b).

From (2.1),

$$\bar{C} = [\Sigma_{j=1}^{N} c_j(1 - U_j u_j/v)]/(\Sigma_{j=1}^{N} u_j U_j).$$

In this case, we have not obtained the closed form solution for U_j as (3.4).

In any case, the schedules that optimize these performance measures seem hard to implement as they now stand.

4. Equal response ratio

We consider here the ratio of the average response time for a job, to the average service time for the job; we call it response ratio for the job. We often wish the response ratio for each job to be equal. That is, we wish to have such a response time vector $(T_1, T_2, \ldots, T_N)$ as satisfies:

$$u_j T_j = K, \qquad j = 1, 2, \ldots, N. \tag{4.1}$$

It is already known that, in the infinite-source single-server queueing model (M/G/1), the processor sharing discipline and the preemptive-resume last-come-first-served (LCFSPR) discipline make the response ratio for each job class equal (see, for example, Kleinrock [11]). The response ratio means the ratio of the expected queueing time of a job class to the expected service time of the job class. (The model has often been used in gaining insight into the performance of time-shared computer systems. However, the infinite-source assumption may sometimes be unrealistic in the sense that, in a usual time-shared computer system, only a finite number of jobs can be in the system.) In this section, we shall see the effect of changing the above infinite-source assumption to the finite-source assumption.

We use the following notation in the subsequent subsections. Let us define $G(k, \mathbf{Z})$ for an arbitrary non-negative integer k and an arbitrary set of natural numbers $\mathbf{Z}$:

$$G(k, \mathbf{Z}) = \Sigma_{n=0}^{Z} (n + k)! \Sigma_{I \subseteq \mathbf{Z}, I=n} \Pi_{i \in I} r_i. \tag{4.2}$$

Note that Z denotes the size of set $\mathbf{Z}$. We assume that $r_i > 0$ for any i unless otherwise specified. Then from (2.2) we have $U(\mathbf{Z})$ as follows:

$$U(\mathbf{Z}) = 1 - 1/G(0, \mathbf{Z}). \tag{4.3}$$

Note that we have, for any j, k, $\mathbf{Z}$ (j not in $\mathbf{Z}$),

$$G(k, \mathbf{Z} \cup \{j\}) = G(k, \mathbf{Z}) + r_j G(k + 1, \mathbf{Z}). \tag{4.4}$$

4.1. Response ratios achieved by processor sharing and LCFSPR

In this subsection, we assume that the service time distributions are general (see Chandy et al. [3]). Assume that the processor-sharing (or LCFSPR) discipline is used at the processor. Let $P(\mathbf{Z})$ denote the probability that jobs of set $\mathbf{Z}$ stay at the processor in the finite-source model: Let $P(0)$ denote the probability that no job stays at the processor. Then by the product-form theorem (Baskett et al. [1], Chandy et al. [3])

$$P(\mathbf{Z}) = P(0)Z! \prod_{j \in \mathbf{Z}} r_j$$

where $P(0) = 1/G(0, \mathbf{N})$, $r_j = v/u_j$, $j = 1, 2, \ldots, N$, and Z denotes the size of set $\mathbf{Z}$. Then, the fraction of time job j is at the terminal is

$$G(0, \mathbf{N} - \{j\})/G(0, \mathbf{N}).$$

And the fraction of time job j is at the processor (i.e., U_j) is

$$1 - G(0, \mathbf{N} - \{j\})/G(0, \mathbf{N}).$$

Thus, the response ratio $u_j T_j$ for job j is, from (2.1) and (4.4),

$$u_j T_j = G(1, \mathbf{N} - \{j\})/G(0, \mathbf{N} - \{j\}), \quad j = 1, 2, \ldots, N. \tag{4.5}$$

Theorem 2. If the processor sharing (or LCFSPR) discipline is used in the finite-source queueing model, the longer the mean service time $1/u_j$ for job j, the smaller the response ratio $u_j T_j$ for job j, for $j = 1, 2, \ldots, N$: That is,

$$u_i T_i \lessgtr u_j T_j \quad \text{if} \quad u_i \lessgtr u_j, \text{ for all } i, j \in \mathbf{N}.$$

(Proof. Omitted.)

Intuitively, we can understand the above result from the following: Under the scheduling disciplines considered, the processing of job j is

interrupted or delayed by the arrival of jobs in the set **N** - {j}, and if job j has a mean service time (on the processor) longer than that of job i, the set of jobs in **N** - {j} consists of shorter jobs than the set of jobs in **N** - {i}.

4.2 Achievability of the equal response ratio under Markovian assumptions

Now we shall examine whether the equal response ratio is achievable or not. In this section, we assume that the service time distributions are exponential for each job. Under the assumption, we already have Property A.1 on the achievability of performance vectors by scheduling disciplines.

If we have an achievable performance vector which provides the equal response ratio for each job, then from (2.1), (4.1), and (2.4),

$$\Sigma_{j\in\mathbf{N}} 1/(K + 1/r_j) = U(\mathbf{N}). \tag{4.6}$$

Since the left side of (4.6) is monotonically decreasing with the increase of K, (4.6) must have a unique solution for K. Then we have the problem of whether the response time vector $(K/u_1, K/u_2, \ldots, K/u_N)$ is achievable or not. For this we can have a positive answer as shown in the following theorem.

Theorem 3. The performance vector **T** which realizes the equal response ratio (i.e., satisfies $u_1T_1 = u_2T_2 = \ldots = u_NT_N$) is achievable by some scheduling discipline in the finite-source queueing model.
(Proof. Omitted.)

Having proved that the equal response ratio is achievable, there remains the question of what scheduling discipline should be used to

achieve it. One answer is provided by mixing strategies as considered in Kameda [9]. Naturally, there remains the possibility that other scheduling disciplines achieve the equal response ratio (cf. Mitrani and Hine [12]).

5. Summary

We have considered finite-source queueing models where each job has a distinct mean service time at the processor and the same mean service time at the terminals.

First, we have studied the optimization problems of various performance measures under Markovian assumptions. Each of the performace measures considered is expressed as a function of the utilization factor of the processor (or the average response time) for each job in the model. We have observed that linear performance measures can be optimized by preemptive priority disciplines and the implementation of them seems to have little difficulties. However, nonlinear performace measures are optimized by scheduling strategies which can be implemented through a certain amount of computation of a nonlinear program (plus a linear program) as given in this paper; at least, we can say that they cannot be simple preemptive fixed priority disciplines; thus, we conjecture that it may be difficult to find and implement the scheduling strategies which optimize such performance measures, also in more general situations.

Then we have shown that the processor-sharing (or LCFSPR) discipline, although it attains the equal response ratio for each job class in infinite-source M/G/1, gives a smaller response ratio to a job having a longer mean service time at the processor, in the model where service time distributions are general. We have also shown that there exists a scheduling discipline whereby the response ratio for each job is equal, in the model in which service time distributions are exponential for each job.

References

[1] Baskett, F., Chandy, K.M., Muntz, R.R., and Palacios, F.G.: "Open, closed, and mixed networks of queues with different classes of customers", J.ACM 22, 2, pp. 248-260 (1975).

[2] Buzen, J.P., and Goldberg, P.S.: "Guidelines for the use of infinite source queueing models in the analysis of computer system performance", Proc. AFIPS 1974 NCC pp. 371-374 (1974).

[3] Chandy, K.M., Howard, J.H.,Jr., and Towsley, D.F.: "Product form and local balance in queueing networks", J.ACM 24, 2, pp. 250-263 (1977).

[4] Coffman, E.G.,Jr., and Mitrani, I.: "A characterization of waiting time performance realizable by single-server queues", Operations Research 28, 3-Part II, pp. 810-821 (1980).

[5] Cox, D.R., and Smith, W.L.: Queues, Methuen, London (1961).

[6] Gelenbe, E., and Mitrani, I.: Analysis and Synthesis of Computer Systems, Academic Press, London (1980).

[7] Kameda, H.: "A finite-source queue with different customers", J.ACM 29, 2, pp. 478-491 (1982).

[8] Kameda, H.: "Appraisals of scheduling decisions in computing systems using a finite-source queueing model", Proc. 8th Int. Symp. on Computer Performance Modelling, Measurement and Evaluation, Amsterdam (North-Holland, Amsterdam), pp. 455-468 (1981).

[9] Kameda, H.: "Realizable performance vectors of a finite-source queue" Operations Research 32, 6, pp. 1358-1367 (1984).

[10] Kleinrock, L.: "A conservation law for a wide class of queueing disciplines", Nav. Res. Log. Quart. 12, pp. 181-192 (1965).

[11] Kleinrock, L.: Queueing Systems, Volume II: Computer Applications. John Wiley, New York (1976).

[12] Mitrani, I., and Hine, J.H.: "Complete parameterized families of job scheduling strategies", Acta Inform. 8, pp. 61-73 (1977).

[13] Scherr, A.L.: An Analysis of Time-Shared Computer Systems. MIT Press, Cambridge, Mass. (1967).

[14] Shapiro, J.F.: Mathematical Programming: Structures and Algorithms. Wiley, New York (1979).