OPTIMAL CONTROL OF FINITE SOURCE PRIORITY QUEUES WITH COMPUTER SYSTEM APPLICATIONS

D. ASZTALOS
Computer Centre of National Planning Office, Budapest, Hungary

(Received November 1979)

Communicated by M. Arato

Abstract—The paper deals with the derivation of optimal control rules for finite source queueing systems with preemptive resume service discipline. The three performance measures considered are: server utilization, mean queue length and throughput.

It is indicated how the results on optimal control can be applied to multiprogrammed computer systems and some numerical examples are given.

1. INTRODUCTION

The closed queueing network models may be successfully applied to the analysis of behaviour of multiprogrammed computer systems consisting of one central processor unit (CPU) and many peripherals. Each job corresponds to one customer and the CPU and the peripherals are represented by the servers. In large configurations it frequently may be assumed that all jobs of a given mix access separate peripherals. In this case all of the peripherals may be represented by one server with infinite capacity. We use that each closed queueing network consisting of two servers, one of which has infinite capacity, is equivalent to a finite source queueing system (FSQ). The server of the equivalent system represents the CPU and the finite source corresponds to the separately used peripherals. In computer system applications the number of customers in the above system is referred as level of multiprogramming. In a given computer system the number of jobs actually executed in the system is constant for a relatively long time-interval. The definition of the optimal priority allocation for each time-interval, when a preemptive-resume discipline is used, has a great practical importance. The optimality should be defined with respect of a well defined performance measure. The performance measures discussed in this paper are: server utilization, average queue length and throughput.

The paper deals with finite source models with exponential structure. This means that the service times and the residence times at the source are mutually independent, exponentially distributed random variables. Let the number of customers be \( N \) and designate them by the integers \( 1, 2, \ldots, N \). Let the service time parameter be \( \mu_i \), the residence time parameter be \( \lambda_i \), for customer \( i \), \( i = 1, 2, \ldots, N \). We consider only the case when for every \( i \) the \( i \)th customer has priority over customers of index higher than \( i \), applying the preemptive resume service discipline. Such systems are referred as exponential finite source priority queueing systems (EFSPQ).

The FSQs are dealt with in general context in [1] considering mainly the priority disciplines. Tomko gives in [2] a detailed analysis of the busy period of EFSPQ model. He shows in the case \( N = 2 \), that the server utilization in steady state case is maximised when \( \lambda_1 > \lambda_2 \). For the case \( N > 2 \), there are some particular results in [3, 4] with respect of optimization of the server utilization. The special conditions of optimality in [3, 4] are: \( \mu_1 > \mu_2 > \cdots > \mu_N \) and \( \lambda_1 > \lambda_2 > \cdots > \lambda_N \). These assumptions are not very realistic in computer systems. Smith proves in [5] the optimality of the \( \mu_1 < \mu_2 < \cdots < \mu_N \) control, minimising the mean queue length in steady-state case when \( \mu_i = \mu_1 \), \( i = 2, \ldots, N \).

In this paper the following new results are proved for mean values in steady-state case. The maximal mean value of server utilization in an EFSPQ is achieved when \( \lambda_1 > \lambda_2 > \cdots > \lambda_N \). An immediate consequence of this fact is that if \( \lambda_1 = \lambda_i \), \( i = 2, \ldots, N \), then the mean value of server utilization is the same for any priority allocation. But in this case, the throughput is maximal and the mean queue length is minimal if \( \mu_1 > \mu_2 > \cdots > \mu_N \).
\(i = 2, \ldots, N\) in EFSPQ. Then the mean value of utilization and throughput are maximal if \(\lambda_1 > \lambda_2 > \cdots > \lambda_N\). There are no similar simple rules of optimal control for mean queue length and throughput of EFSPQ when \(\lambda_i\) and \(\lambda_j\), \(i = i, \ldots, N\) are different.

We note that our results cannot be applied to finite source queueing systems with general service time distributions and nonpreemptive priority discipline unlike to the infinite source models (see e.g. [1]).

2. THE PERFORMANCE MEASURES

Let us consider an EFSPQ model described in the introduction. We consider three performance measures of an EFSPQ model: mean values of server utilization, queue length and throughput. All these values are examined in the steady-state of the system.

Let \(E8^{(j)}\) denote the mean value of the busy period when there are \(j\) customers in the system. It follows from the exponential structure that the mean value of the idle period equals \(m_{N-1} = (\lambda_1 + \lambda_2 + \cdots + \lambda_N)^{-1}\) when there are \(N\) customers in the system. The mean value of server utilization can be calculated as:

\[
\rho = \frac{E8^{(N)}}{E8^{(N)} + m_N^{-1}}. \tag{1}
\]

To calculate \(E8^{(N)}\) we can use the recursive formulas of [2], which are the following:

\[
E8^{(j)} = \frac{m_{j-1}}{m_j} \left[ E8^{(j-1)} + \frac{1}{\mu_j} (1 - \varphi_{j-1}(\lambda_j)) \left( 1 + m_{j-1} E8^{(j-1)} \right) \right] + \frac{\lambda_j}{m_j} \cdot \frac{1}{\mu_j} \left( 1 + m_{j-1} E8^{(j-1)} \right), \tag{2}
\]

\[
\varphi_j(s) = \frac{m_{j-1}}{m_j} \left[ \varphi_{j-1}(s + \lambda_j) + \frac{\mu_j [\varphi_{j-1}(s) - \varphi_{j-1}(s + \lambda_j)]}{\mu_j + s + m_{j-1}(1 - \varphi_{j-1}(s))} \right] + \frac{\lambda_j}{m_j \mu_j + s + m_{j-1}(1 - \varphi_{j-1}(s))}, \tag{3}
\]

where \(j = 1, \ldots, N\); and \(m_j = \lambda_1 + \cdots + \lambda_j\) and \(\varphi_j(s)\) is the Laplace-transform of the busy period variable when there are \(j\) customers in the system. These formulas will be used in a slightly different form in the sequel. They are:

\[
E8^{(j)} = E8^{(j-1)} \frac{m_{j-1}}{m_j \mu_j} (\mu_j + m_j - m_{j-1} \varphi_{j-1}(\lambda_j)) + \frac{1}{m \mu_j} (m_j - m_{j-1} \varphi_{j-1}(\lambda_j)). \tag{4}
\]

\[
\varphi_j(s) = \frac{m_{j-1} [\mu_j \varphi_{j-1}(s) + (s + m_{j-1}) \varphi_{j-1}(s + \lambda_j) - m_{j-1} \varphi_{j-1}(s) \varphi_{j-1}(s + \lambda_j)] + \lambda_j \mu_j}{m_j (s + \mu_j + m_{j-1}(1 - \varphi_{j-1}(s)))}. \tag{5}
\]

One should maximise \(E8^{(N)}\) to maximise \(\rho\).

Let \(E_0\) be the mean value of the queue length in steady-state, and \(E_{0j}\) be that part of \(E_0\) which corresponds to the customer of priority \(j\). \(E_{0j}\) is given by the formula (see [1] Ch. IV.):

\[
E_{0j} = 1 - \frac{\mu_j}{\lambda_j} (e_{j-1} - e_j), \tag{6}
\]

where

\[
e_j = (1 + m_j E8^{(j)})^{-1} \tag{7}
\]

is the steady-state probability of the idle server when there are \(j\) customers in the system. Then \(E_0 = \Sigma_{j=1}^N E_{0j}\).

Let \(T\) be the throughput of the EFSPQ model, which is defined as the mean value of number of customers serviced in unit time in the steady-state case.

Let \(r_j\) be the steady-state probability that the \(j\)-th customer is found at the source. Obviously

\[
r_j = 1 - E_{0j} = \frac{\mu_j}{\lambda_j} (e_{j-1} - e_j).
\]
can be interpreted as the average time spent by customer \( j \) in the source in each time unit. Thus, the throughput of customer \( j \) can be calculated as

\[
T_j = \lambda_i r_j = \mu_i (e_{j-1} - e_j).
\]

Thus, we get

\[
T = \sum_{j=1}^{N} T_j = \sum_{j=1}^{N} \mu_i (e_{j-1} - e_j). \tag{8}
\]

3. OPTIMAL CONTROL PROBLEMS

In this section we will prove some results with respect of optimal control of EFSPQ systems.

THEOREM 1

Let us consider an EFSPQ system. \( E\theta^{(N)} \)—the mean value of busy period in steady state case—is maximal when \( \lambda_1 > \lambda_2 > \cdots > \lambda_N \).

Proof. It is sufficient to show for any \( j, 1 \leq j-1 \) and \( j \leq N \), that if \( \lambda_{j-1} > \lambda_j \), then

\[
E\theta^{(j)}(\lambda_1, \lambda_2, \ldots, \lambda_{j-1}, \lambda_j) > E\theta^{(j)}(\lambda_1, \lambda_2, \ldots, \lambda_j, \lambda_{j-1}).
\]

if \( \lambda_{j-1} < \lambda_j \), then

\[
E\theta^{(j)}(\lambda_1, \lambda_2, \ldots, \lambda_{j-1}, \lambda_j) < E\theta^{(j)}(\lambda_1, \lambda_2, \ldots, \lambda_j, \lambda_{j-1})
\]

and if \( \lambda_{j-1} = \lambda_j \), then

\[
E\theta^{(j)}(\lambda_1, \lambda_2, \ldots, \lambda_{j-1}, \lambda_j) = E\theta^{(j)}(\lambda_1, \lambda_2, \ldots, \lambda_j, \lambda_{j-1}).
\]

In this proof the priority of customer with parameter \( \lambda_i, i = 1, \ldots, j \) is defined by its position number from left to right in the row vector \( (\lambda_1, \lambda_2, \ldots, \lambda_j) \).

For the sake of simplicity we use the notations:

\[
E\theta_1^{(j)} = E\theta^{(j)}(\lambda_1, \lambda_2, \ldots, \lambda_{j-1}, \lambda_j); \quad E\theta_2^{(j)} = E\theta^{(j)}(\lambda_1, \lambda_2, \ldots, \lambda_j, \lambda_{j-1}).
\]

Then, using (4) repeatedly we get:

\[
E\theta_1^{(j)} - E\theta_2^{(j)} = \frac{1}{m \mu_{j-1} \mu_j} (a \beta - \gamma e)(1 + m_{j-2} E\theta^{(j-2)}) \tag{9}
\]

where

\[
\alpha = \mu_{j+1} m_j (m_{j-2} \lambda_{j-1}) \varphi_{j-1}(\lambda_{j-1}),
\]

\[
\beta = \mu_{j-1} m_{j-2} + \lambda_{j-1} - m_{j-2} \varphi_{j-2}(\lambda_{j-1}),
\]

\[
\gamma = \mu_{j+1} m_j - (m_{j-2} + \lambda_j) \varphi_{j-1}(\lambda_{j-1}),
\]

\[
\epsilon = \mu_j + m_{j-2} + \lambda_j - m_{j-2} \varphi_{j-2}(\lambda_{j-1}).
\]

Using (5) we get:

\[
\varphi_{j-1}(\lambda_j) = \frac{\mu_{j-1} (\lambda_{j-1} + m_{j-2} \varphi_{j-2}(\lambda_{j-1})) + m_{j-2} \varphi_{j-2}(\lambda_{j-1} + \lambda_j) \cdot D}{(m_{j-2} + \lambda_{j-1}) (\mu_{j-1} + \lambda_j + m_{j-2} - m_{j-2} \varphi_{j-2}(\lambda_{j-1}))}, \tag{11}
\]
where \( D = m_{i-2} + \lambda_{i-1} - m_{i-2} \phi_{i-2}(\lambda_{i-1}), \)

\[
\phi_{i-1}(\lambda_{i-1}) = \frac{\mu_i [\lambda_i + m_{i-2} \phi_{i-2}(\lambda_{i-1})] + m_{i-2} \phi_{i-2}(\lambda_{i-1} + \lambda_i) \cdot C}{(m_{i-2} + \lambda_i)(\mu_i + \lambda_{i-1} + m_{i-2} - m_{i-2} \phi_{i-2}(\lambda_{i-1}))},
\]

where \( C = m_{i-2} + \lambda_i - m_{i-2} \phi_{i-2}(\lambda_i). \)

The sign of (9) is defined by the sign of \( \alpha \beta - \gamma \varepsilon. \)

We show that \( \alpha \beta - \gamma \varepsilon \) is a symmetric expression with respect of \( \mu_{i-1} \) and \( \mu_i, \)

\[
\alpha \beta - \gamma \varepsilon = \begin{cases} 
> 0 & \text{if } \lambda_{i-1} > \lambda_i \\
= 0 & \text{if } \lambda_{i-1} = \lambda_i \\
< 0 & \text{if } \lambda_{i-1} < \lambda_i. 
\end{cases}
\] (13)

This will prove our theorem.

Using (10)–(12) we get:

\[
\alpha \beta - \gamma \varepsilon = \left[ \frac{\mu_i + \mu_j - (m_{i-2} + \lambda_{i-1}) \phi_{i-1}(\lambda_i)}{(\mu_i + D)} \right] ((\mu_i + D) + C) = \left[ \frac{\mu_i + \mu_j - (m_{i-2} + \lambda_{i-1}) \phi_{i-1}(\lambda_i)}{(\mu_i + D)} \right] (\mu_i + D) + C
\]

\[
= \frac{D}{\mu_i + D} \left[ \mu_i + m_{i-2} \phi_{i-2}(\lambda_i) \right] (\mu_i + D) + C
\]

\[
= \frac{(\mu_i + m_{i-2} \phi_{i-2}(\lambda_i)) (\mu_i + D) + C}{(\mu_i + D) + C}
\]

\[
= \frac{(\mu_i + m_{i-2} \phi_{i-2}(\lambda_i)) (\mu_i + D) + C}{(\mu_i + D) + C}
\]

where

\[
X = \frac{m_{i-2} \phi_{i-2}(\lambda_{i-1})}{\mu_{i-1} + \lambda_i},
\]

\[
Y = \frac{m_{i-2} \phi_{i-2}(\lambda_{i-1})}{\mu_{i-1} + \lambda_i},
\]

\[
E = \frac{m_{i-2} \phi_{i-2}(\lambda_{i-1})}{\mu_{i-1} + \lambda_i}.
\]

It is easy to check, that \( (\mu_{i-1} + C)(\mu_i + D) \) is positive and is the same for both priority orders \( (\lambda_1, \lambda_2, \ldots, \lambda_{i-1}, \lambda_i) \) and \( (\lambda_1, \lambda_2, \ldots, \lambda_i, \lambda_{i-1}) \). Thus, it is sufficient to prove (13) for

\[
(a \beta - \gamma \varepsilon)^* = (a \beta - \gamma \varepsilon)(\mu_{i-1} + C)(\mu_i + D).
\]

Then

\[
(a \beta - \gamma \varepsilon)^* = \mu_{i-1} (\mu_i (C - D)(X + Y - m_i - m_{i-2} E) + \mu_j (D(C - D)(Y - m_i) + \mu \mu(C - D)(X - m_i) + CD(C - D)(m_{i-2} E - m_i).
\]

It is easy to check, that \( D(Y - m_i) = C(X - m_i) = -CD. \)

Thus, with replacements we get:

\[
(a \beta - \gamma \varepsilon)^* = (D - C)(\mu_{i-1} \mu_m m_{i-2} + \phi_{i-2}(\lambda_{i-1} + \lambda_j) - \phi_{i-2}(\lambda_{i-2} + \lambda_i)) + (\mu_{i-1} + \mu_j) CD + CD(\lambda_{i-1} + \lambda_j + m_{i-2}(1 - \phi_{i-2}(\lambda_{i-1} + \lambda_j))).
\]

Thus, the \( (a \beta - \gamma \varepsilon)^* \) is a symmetric expression with respect of \( \mu_{i-1} \) and \( \mu_i. \) It follows from the properties of the Laplace-transform that

\[
1 + \phi_{i-2}(\lambda_{i-1} + \lambda_j) - \phi_{i-2}(\lambda_{i-1} - \phi_{i-2}(\lambda_i)) > 0, \quad C > 0, \quad D > 0 \quad \text{and} \quad 0 < 1 - \phi_{i-2}(\lambda_{i-1} + \lambda_j) < 1.
\]
Furthermore
\[ D - C = \lambda_{j-1} - \lambda_j - m_{j-2}(\varphi_{j-2}(\lambda_{j-1}) - \varphi_{j-2}(\lambda_j)) = \begin{cases} > 0 & \lambda_{j-1} > \lambda_j \\ = 0 & \lambda_{j-1} = \lambda_j \\ < 0 & \lambda_{j-1} < \lambda_j \end{cases} \]  \tag{15} \]

Thus the sign of \((\alpha \beta - \gamma \varepsilon)\) is defined by \(D - C\).

Equation (13) follows from (14) and (15), and this proves our theorem.

**Corollary 1**

Let \(\lambda_1 = \lambda_2 = \cdots = \lambda_N\). Then \(E\delta^{(N)}\) is the same for any priority allocation.

We now consider the EFSPQ systems with homogeneous source, i.e., \(\lambda_1 = \cdots = \lambda_N = \lambda\).

**Theorem 2**

Let us consider an EFSPQ system with homogeneous source. The \(Eq\) is minimal and \(T\) is maximal when 
\[ \mu_1 > \mu_2 > \cdots > \mu_N. \]

**Proof.** For EFSPQ with homogeneous source \(\lambda_i = \lambda, \ i = 1, \ldots, N\). Using (6) and (8) we get for the mean queue length and throughput the following expressions:
\[ Eq = \sum_{j=1}^{N} Eq_j = N - \frac{1}{\lambda} \sum_{j=1}^{N} \mu_j (e_{j-1} - e_j) \]
and
\[ T = \sum_{j=1}^{N} \mu_j (e_{j-1} - e_j). \]

Thus, \(Eq = N - T/\lambda\) and if \(T\) is maximised, at the same time \(Eq\) is minimised.

It is sufficient to show that
\[ (T_j + T_{j+1})_{\mu_j > \mu_{j+1}} > (T_j + T_{j+1})_{\mu_j < \mu_{j+1}}. \]  \tag{16} \]

Let us consider the sum
\[ T_j + T_{j+1} = \mu_j e_{j-1} - \mu_{j+1} e_{j+1} - (\mu_j - \mu_{j+1}) e_j. \]  \tag{17} \]
Replacing \(E\delta^{(N)}\) in (7) by the r.h.s. of (4) and repeatedly using (7) we get for \(e_j\):
\[ e_j = \frac{\mu e_{j-1}}{\mu_j + \lambda [j - (j - 1) \varphi_{j-1}(\lambda)]}. \]  \tag{18} \]

Let \(C = \lambda [j - (j - 1) \varphi_{j-1}(\lambda)]\). It is obvious that \(C \geq 0\) for \(\lambda \leq 0\). Using (18) the expression (17) may be rewritten as:
\[ T_j + T_{j+1} = \mu_j e_{j-1} - \frac{\mu_{j+1} + C}{\mu_j + C} - \left( e_{j-1} + \frac{1}{\mu_j} \right). \]  \tag{19} \]

It follows from (7) that \(e_{j-1} > e_{j+1}, e_{j-1}\) does not depend on \(\mu_j, \mu_{j+1}\) and \(e_{j+1}\) would not change its value when the customers of priorities \(j\) and \(j+1\) are mutually replaced by each other (see Corollary 1). At the same time, if \(\mu_j > \mu_{j+1}\), then \((\mu_{j+1} + C)/(\mu_j + C) > 1/\mu_j\) and if \(\mu_j < \mu_{j+1}\), then \((\mu_{j+1} + C)/(\mu_j + C) < 1/\mu_j\). Thus, (16) and the theorem are proved.

We now consider the EFSPQ systems with homogeneous service, i.e., \(\mu_1 = \mu_2 = \cdots = \mu_N = \mu\).

**Theorem 3.** Let us consider an EFSPQ system with homogeneous service. The \(E\delta^{(N)}\) and \(T\) are maximal when \(\lambda_1 > \lambda_2 > \cdots > \lambda_N\).
Proof. If the service is homogeneous, we get for $T$:

$$T = \mu \sum_{i=1}^{N} (e_{i-1} - e_i) = \mu (e_0 - e_N) = \mu (1 - e_N).$$  \hspace{1cm} (20)$$

It follows from Theorem 1 that $E_0^{(N)}$ is maximal when $\lambda_1 > \lambda_2 > \cdots > \lambda_N$. It follows from (7) that actually in this case $e_N$ would be minimal, which means that $T$ is maximal when $\lambda_1 > \lambda_2 > \cdots > \lambda_N$. The theorem is proved.

Some earlier results, concerning the homogeneous source case, are contained in [6]. It is proved, that in an exponential, finite homogeneous source queueing system $E_0^{(N)}$ is the same for a wide class of service disciplines including the processor sharing one, too. Using the results of [7], a closed form solution can be derived for $E_0^{(N)}$, both in the homogeneous source and the inhomogeneous source systems with processor sharing discipline. The last result is a significant improvement of the algorithm in [2], calculating $E_0^{(N)}$.

4. APPLICATIONS AND NUMERICAL RESULTS

We have defined some optimal control rules for $M|M|1/N$ (or EFSPQ) queueing systems in the previous section. For the case, when the source is homogeneous, the optimal control rule, minimising $E_q$ and maximising $T$ is the same as for the infinite source $M|M|1$ preemptive and $M/G|l$ nonpreemptive systems. This is the $\mu_1 > \mu_2 > \cdots > \mu_N$ rule where the index is the priority number.

For the general $M|M|1/N$ system we have an optimal control rule only with respect of the utilization factor $\rho$, which is the $\lambda_1 > \lambda_2 > \cdots > \lambda_N$ rule.

It would be important for applications to define that set of parameters $S = \{[\mu_1, \lambda_1], \ldots, (\mu_N, \lambda_N)\}$ for which the $\mu_1 > \mu_2 > \cdots > \mu_N$ rule is optimal with respect of the throughput $T$. Numerical examples (see Table 1), measurements and simulation experiments [5] imply that set $S$ is sufficiently large.

Applying the results of Section 3 to multiprogrammed computer systems, the most important deduction is that under exponentiality assumptions the throughput of the system and the utilization of the CPU cannot be optimised in all cases with the same control rule, in contrast to the widely used heuristic control rule $\mu_1 > \mu_2 > \cdots > \mu_N$, used in practice to improve both performance measures.

**Table 1. Numerical examples**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Priority allocation</th>
<th>$\rho$</th>
<th>$T$</th>
<th>$E_q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_0 = 1.0$</td>
<td>$\lambda_i = 0.2$</td>
<td>(1, 2, 3)</td>
<td>0.690$^*$</td>
<td>0.690$^*$</td>
</tr>
<tr>
<td>$i = 1, 2, 3$</td>
<td>$\lambda_i = 0.4$</td>
<td>(3, 2, 1)</td>
<td>0.720$^*$</td>
<td>0.720$^*$</td>
</tr>
<tr>
<td>$\mu_1 = 10.0$</td>
<td>$\lambda_i = 0.25$</td>
<td>(1, 2, 3)</td>
<td>0.810</td>
<td>0.501$^*$</td>
</tr>
<tr>
<td>$\mu_2 = 1.0$</td>
<td>$i = 1.2.3$</td>
<td>(3, 7, 1)</td>
<td>0.810</td>
<td>0.255</td>
</tr>
<tr>
<td>$\mu_3 = 0.1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu_4 = 1.0$</td>
<td>$\lambda_i = 0.1$</td>
<td>(2, 3, 1)</td>
<td>0.856$^*$</td>
<td>1.182</td>
</tr>
<tr>
<td>$\mu_5 = 2.0$</td>
<td>$\lambda_i = 3.0$</td>
<td>(3, 2, 1)</td>
<td>0.835</td>
<td>1.936$^*$</td>
</tr>
<tr>
<td>$\mu_6 = 3.0$</td>
<td>$\lambda_i = 1.5$</td>
<td>(1, 3, 2)</td>
<td>0.795</td>
<td>1.810</td>
</tr>
<tr>
<td>$\mu_7 = 4.0$</td>
<td>$\lambda_i = 2.0$</td>
<td>(1, 2, 3)</td>
<td>0.975</td>
<td>1.803$^*$</td>
</tr>
<tr>
<td>$\mu_8 = 3.0$</td>
<td>$\lambda_i = 0.2$</td>
<td>(3, 1, 2)</td>
<td>0.996$^*$</td>
<td>0.640</td>
</tr>
<tr>
<td>$\mu_9 = 0.3$</td>
<td>$\lambda_i = 10.0$</td>
<td>(2, 1, 3)</td>
<td>0.975</td>
<td>1.796</td>
</tr>
<tr>
<td>$\mu_{10} = 2.0$</td>
<td>$\lambda_i = 0.5$</td>
<td>(1, 2, 3)</td>
<td>0.506$^*$</td>
<td>0.924$^*$</td>
</tr>
<tr>
<td>$\mu_{11} = 1.8$</td>
<td>$\lambda_i = 0.4$</td>
<td>(3, 1, 2)</td>
<td>0.501</td>
<td>0.908</td>
</tr>
<tr>
<td>$\mu_{12} = 1.6$</td>
<td>$\lambda_i = 0.3$</td>
<td>(3, 2, 1)</td>
<td>0.499$^*$</td>
<td>0.900$^*$</td>
</tr>
<tr>
<td>$\mu_{13} = 2.0$</td>
<td>$\lambda_i = 0.5$</td>
<td>(2, 3, 1)</td>
<td>0.533</td>
<td>1.182</td>
</tr>
<tr>
<td>$\mu_{14} = 2.2$</td>
<td>$\lambda_i = 0.4$</td>
<td>(3, 1, 2)</td>
<td>0.539$^*$</td>
<td>1.194$^*$</td>
</tr>
<tr>
<td>$\mu_{15} = 2.4$</td>
<td>$\lambda_i = 0.7$</td>
<td>(3, 2, 1)</td>
<td>0.537</td>
<td>1.191</td>
</tr>
<tr>
<td>$\mu_{16} = 1.0$</td>
<td>$\lambda_i = 1.0$</td>
<td>(1, 2, 3)</td>
<td>0.987</td>
<td>1.506$^*$</td>
</tr>
<tr>
<td>$\mu_{17} = 1.0$</td>
<td>$\lambda_i = 1.5$</td>
<td>(2, 3, 1)</td>
<td>0.987$^*$</td>
<td>0.681</td>
</tr>
<tr>
<td>$\mu_{18} = 0.1$</td>
<td>$\lambda_i = 1.4$</td>
<td>(3, 2, 1)</td>
<td>0.987</td>
<td>0.187$^*$</td>
</tr>
</tbody>
</table>

$^*$ Denotes the optimal value of the measure.
$-$ Denotes the worst value of the measure.
In [8] is described how the homogeneous source EFSPQ system was applied to model a multiprogrammed computer system. However, at that time the $\mu_1 > \mu_2 > \cdots > \mu_N$ control rule was used only heuristically in [8].

REFERENCES