□ 研究論文 □

韓國軍事運營分析 學會誌
第17卷, 第1號, 1991.6.30.

On The Performance of A Suboptimal Assignment Policy in N-Queue m-Server System

Ko, Soon-Ju*

Abstract

Consider N queues without arrivals and with m identical servers. All jobs are independent and service requirements of jobs in a queue are i.i.d. random variables. At any time only one server may be assigned to a queue and switching between queues are allowed. A unit cost is imposed per job per unit time. The objective is to minimize the expected total cost.

An flow approximation model is considered and an upperbound for the percentage error of best nonswitching policies to an optimal policy is found. It is shown that the best nonswitching policy is not worse than 11% of an optimal policy

For the stochastic model, we consider the case in which the service requirements of all jobs are i.i.d. with an exponential distribution. A longest first policy is shown to be optimal and a worst case analysis shows that the nonswitching policy which starts with the longest queues is not worse than 11% of the optimal policy.

* National Defense College

1. Introduction

Many situations in resource sharing environments may be modeled as following : There are N classes of jobs with x_i jobs in class i and the processing requirements of jobs are statistically independent with class dependent distributions. m ($<N$) identical processors in parallel are to be used to process the jobs and only one processor may be assigned to a class at each time. Pre-emptions are allowed at any time. The problem is then how to assign processors to minimize the expected flowtime. This model differs from the Weber's model in the pre-emptive scheduling of n independent parallel jobs on m identical processors by the assignment constraint and allowing different distributions. In the queueing system context, by interpreting classes as queues, this model corresponds to a multi-server system of N-competing queues with no arrivals in which a unit cost per customer per unit time is imposed, where the problem is to minimize expected total cost over infinite horizon.

Considerable research has been done on single server systems of competing queues (e.g. [1] through [5]), while very few results are available for multi-server systems[6,7]. Notice that the problem can be formulated as optimal server allocation problem in multi-armed bandit process where it is well is known that an index rule is optimal for single server problems, while it is no more optimal for multi-server problems[8,9,20]

An important priority rule in server allocation policies is so called c μ-rule. This policy gives a higher priority to the classes with higher product of the waiting cost per customer per unit time (C_i) and the mean service rate (μ_i).

For a single server system of many competing queues, the cμ-rule is optimal to minimize the average cost among nonpre-emptive policies[1], and it behaves near optimal even in finite buffer systems as the buffer lengths increases[4]. It is also shown in [2] and [3] by formulating the system as a multiarmed bandit process that cμ-rule is still optimal for discounted total expected cost over infinite horizon for the models with geometric service requirements. Notice that cμ-rule is an index rule for that multiarmed bandit process model and independent of arrival patterns. But for the multiserver system of competing queues cμ-rule is no more optimal[7]. This breakdown would share a reason why the index rules fail to be optimal in multiserver multiarmed bandit problems. But there dose not seem to be any quantitative analysis of the performance of index rule in multiserver multiarmed bandit processes.

Determination of optimal assignment may be solved using linear programming formulation as in[6] for finite buffer length systems, but it would hardly give a structural

information on the optimal policy and have restrictive application to systems with small buffer lengths. Nevertheless $c\mu$-rule being the policy that allocate services to the nonempty queues with largest expected cost decrease per unit time, it is tempting to conjecture that it is a good heuristic schedule for multiserver system.

In this paper we consider the model described above and try to get an upper bound for the relative performance of a priorty policy, $c\mu$-rule. A flow approximation analysis is made and it is shown that the performance of the best priority policies are within 11% of the optimal policy, while showing that $c\mu$-rule is not necessarily best among priority policies. For a stochastic model, we consider the case in which the service requirements of all jobs are i.i.d with an exponential distribution. A longest queues first policy is shown to be optimal and a worst case analysis shows that the priority which starts with the longest queues is not worse than 11% of the optimal policy. This may support the result obtained by the flow approximation analysis.

2. A flow approximation

2.1. The model and problem formulation

Let x_i and μ_i be the number of jobs and the service rate of jobs in queue i respectively. A flow approximation model to be considered is

$$\dot{x}_i(t) = -\mu_i I\{x_i(t)>0\} u_i(t), \quad x_i(0)=x_i, \quad i=1,2,3,\cdots\cdots,N \tag{2.1}$$
$$u_i(t)=\{0,1\}, \quad \sum_i u_i(t)\le m, \text{ for all } t,$$

where $x_i(t)$ represents the number of jobs in queue i at t and I represents the indicator function. At each time only one server is assigned to a queue and pre-emptions are allowed at any time. That is, the assignments may be switched from a queue to the other at any time while pre-emptions are allowed. Hence the queues to which a server is not assigned are freezed. The objective is to minimize the total cost

$$J=\int_0^{t_f}\sum_i x_i(t)\,dt, \tag{2.2}$$

where t_f is the time at which all the jobs are completed. Throughout we assume that $\mu_1 \ge \mu_2 \ge \cdots\cdots \ge \mu_N$.

we first derive some preliminary results that will be used later in the next section.

2.2. Preliminaries

Here we consider a system of three queues and two servers. A priority(list) policy is defined to be the policy in which a priority list of queues is given a priori, and whenever a server completes all the jobs in a quese, the first unscheduled queue in the list is assigned to that server. Hence once two servers are assigned to queues i and j initially then a server will complete all the jobs of one of queues at $t=\min\{\frac{x_i}{\mu_i}, \frac{x_j}{\mu_j}\}$ and start to process jobs in the third queue. We call this (i,j)-*rule* and hence $c\mu$-*rule* corresponds to (1,2)-*rule*. For any policy, we note that the total cost can be decomposed into two parts, processing cost and the cost which incurred by freezing queues(call this extra cost). The processing cost C_x is independent of policy and equal to $\frac{1}{2}\sum_i (\frac{x_i}{\mu_i})x_i$ (total area under three triangulars. See Figure 1.). Thus minimizing the total cost is equivalent to minimizing the extra cost. Let V^{π} and C^{π} be the total cost and the extra cost respectively under policy π. Notice that the total cost under (i,j)-*rule* is

$$V_{ij} = x_k \min\{\frac{x_i}{\mu_i}, \frac{x_j}{\mu_j}\} + C_x, \quad k \neq i, j.$$

Fact 1 : If $x_3 \geq x_1, x_2$, then $c\mu$-*rule* is never optimal.

Proof :

Let i be such that $\frac{x_i}{\mu_i} = \max\{\frac{x_1}{\mu_1}, \frac{x_2}{\mu_2}\}$ and j be the other, $i, j=1$ or 2.

Then

$C_{i3} = x_3(\frac{x_i}{\mu_j})$ and $C_{j3} = x_i(\frac{x_i}{\mu_j})$ by $\mu_3 \leq \mu_i$, $x_3 \geq x_i$.

Thus $(j, 3)$-*rule* is better than $c\mu$-rule.

Fact2 : (2,3)-*rule* is never optimal among priority policies.

Proof :

In order for (2,3)-*rule* to be the best among priority policies, we should have

i) $x_1 \min\{\frac{x_2}{\mu_2}, \frac{x_3}{\mu_3}\} \geq x_3 \min\{\frac{x_1}{\mu_1}, \frac{x_2}{\mu_2}\}$ *ii)* $x_1 \min\{\frac{x_2}{\mu_2}, \frac{x_3}{\mu_3}\} \geq x_2 \min\{\frac{x_1}{\mu_1}, \frac{x_3}{\mu_3}\}$

We consider the following possible cases and show that *i) ii)* can be achieved in no cases.

1) $\frac{x_1}{\mu_1} = \max\frac{x_i}{\mu_i}$: this implies $x_1 \geq x_2, x_3$. Thus either *i)* or *ii)* can not be achieved depending on whether $\frac{x_2}{\mu_2} \geq \frac{x_3}{\mu_3}$ or not.

2) $\frac{x_2}{\mu_2} = \max\frac{x_i}{\mu_i}$: For *i)*, $x_1(\frac{x_3}{\mu_3}) \leq x_3(\frac{x_1}{\mu_1})$ is required. But $\mu_1 \geq \mu_3$.

3) $\frac{x_3}{\mu_3} = \max\frac{x_i}{\mu_i}$: For *ii)*, $x_1(\frac{x_2}{\mu_2}) \leq x_2(\frac{x_1}{\mu_2})$ is required. But $\mu_2 \geq \mu_3$.

Fact 3 : (1,3)-*rule* is optimal among priority policies if and only if

i) $\frac{x_3}{\mu_3} > \frac{x_1}{\mu_1} > \frac{x_2}{\mu_2}$ or *ii)* $\frac{x_3}{\mu_3} > \frac{x_2}{\mu_2} > \frac{x_1}{\mu_1}$, $x_3 \geq x_2$.

Proof : Similar arguments as in Fact 2 proves.

2.3. A sufficient condition for the optimality of $c\mu$-*rule*.

Sufficient conditions for the optimality of $c\mu$-*rule* and (1,3)-*rule* are given here.

Lemma 2.1 : If $\max\left(\frac{x_1}{\mu_1}, \frac{x_2}{\mu_2}\right) \geq \min\left(\frac{x_1}{\mu_1}, \frac{x_2}{\mu_2}\right) + \frac{x_3}{\mu_3}$, $c\mu$-*rule* is optimal.

Proof :

Let $l, m \mid c\mu$-*rule* be the policy which serves queues l and m for ε units of time and then follows $c\mu$-*rule* thereafter. Let i be such that $\frac{x_i}{\mu_i} = \max\left(\frac{x_1}{\mu_1}, \frac{x_2}{\mu_2}\right)$ and j be the other, $i, j = 1$ or 2. We show that for $(l, m) = (i, 3)$ or $(j, 3)$

$$V_{c\mu}(x) \leq V_{\cdot,\cdot\,|\,c\mu}(x) \text{ for all } x.$$

Then we have

$$C_{c\mu}(x) = x_3\left(\frac{x_j}{\mu_j}\right)$$

$$C_{i3\,|\,c\mu}(x) = x_i\varepsilon + \left(\frac{x_j}{\mu_j}\right)(x_3 - \mu_3\varepsilon)$$

$$= \mu_i\left(\frac{x_j}{\mu_j}\right)\varepsilon + \left(\frac{x_j}{\mu_j}\right)(x_3 - \mu_3\varepsilon)$$

$$C_{j3\,|\,c\mu}(x) = (x_i - x_3 + \mu_3\varepsilon)\varepsilon + (x_3 - \mu_3\varepsilon)\left(\frac{x_j}{\mu_j}\right)$$

$$= (x_i - x_3)\varepsilon + (x_3 - \mu_3\varepsilon)\left(\frac{x_j}{\mu_j}\right) + o(\varepsilon)$$

This we see that $C_{c\mu} \leq C_{i3\,|\,c\mu}$ by $\mu_j \geq \mu_3$, and $C_{c\mu} \leq C_{j3\,|\,c\mu}$ since the assumption implies $\frac{x_i}{\mu_3} \geq \frac{x_j}{\mu_j} + \frac{x_3}{\mu_3}$

In addition to Lemma 1, we give here a sufficient condition under which (1,3)-*rule* is optimal without proof. Similiar arguments proves the following.

Lemma 2.2. : If $\frac{x_3}{\mu_3} \geq \frac{x_1}{\mu_1} + \frac{x_2}{\mu_2}$ with $x_3 >$ then (1,3)-*rule* is optimal.

2.4. Performance of the best priority policies.

Let be a priority rule for a system of N queues and m servers $(N > m)$. We let V^* and C^* denote the total cost and extra cost under an optimal policies respectively. We are interested in an upper bound for $\frac{V - V^*}{V^*}$ for the performance of the policy against the optimal policy.

We first consider symmetric models, and the general case will be investigated later.

2.4.1. Symmetric case

We consider here a model with $\mu_i = \mu$ and $x_i = n$, i=1,2,3,....N. In this model processor sharing policy is optimal (see Theorem 2.4 below) and all the priority rules are equivalent. Let be any priority rule.

Theorem 2.3 : If $x_i = n$, $\mu_i = \mu$, $i = 1,2,....N$ with $N = mq + r$, q integer, then

$$\frac{T_2 - T_*}{T_*} = \frac{r(m-r)}{N^2}$$

Proof : It can be easily obtained that

$$C_2 = (N-m)n^2 + (N-2m)n^2 + ...(N-qm)n^2$$

$$= (Nq - \frac{q(q+1)}{2}m)n^2 \qquad (2.3)$$

$$C_* = T_* - C_2$$

$$= \frac{1}{2}(Nn)(\frac{Nn}{m}) - Nn^2. \qquad (2.4)$$

Hence we get using (2.3), (2.4) and $N = mq + r$,

$$\frac{T_2 - T_*}{T_*} = \frac{C_2 - C_*}{T_*} = \frac{(2q+1)m}{N} - \frac{q(q+1)m^2}{N^2} - 1$$

$$= \frac{r(m-r)}{N^2} \qquad (2.5)$$

We notice that (2.5) can be maximized by $m = 2$, $N = 3$ and is optimal for $N = mq$.

2.4.2. The Longest Queues First policy

We consider here a system in which $x_1 \geq x_2 \geq .....\geq x_N$, $\mu_i = \mu$, $i = 1, 2,.....N$. We show first that a Longest Queues first policy, the one which assign servers at each time to the longest queues (call this *), is optimal and investigate the performance of (1,2)-rule for N=3, m=2. Notice that * is the processor sharing policy for the symmetric case in 2.4.1.

Theorem 2.4 : For a system of N queues and m servers with $\mu_i = \mu$, $i = 1,2,.....N$, the policy * is optimal.

Proof : We only prove for $N=3$ and $m=2$. Similar argument proves for the case that $N > m \geq 2$.

We assume that $\mu=1$ and $x_1 \geq x_2 \geq x_3$ without loss of generality. Assume $x_1 \leq x_2 + x_3$, otherwise $c\mu$-rule is optimal by Lemma 2.1 but it can be shown that $C_\mu = C_*$, where C_*

is the extra cost under *. Let $C_{13,*}(x)$ be extra cost under the policy which process queue 1 and 3 for ε units of time and follows * thereafter. Direct calculation shows that (see Figure 2. for illustration)

$$C_* = x_3 l + (x_2+x_3-x_1)\tau + \tfrac{1}{2}(x_1-x_2)\tau + \tfrac{3}{4}(x_2+x_3-x_1)^2$$

$$C_{13,*} = x_2\varepsilon + (x_3-\varepsilon)l^* + (x_2+x_3-x_1-\varepsilon)\tau^* + \tfrac{1}{2}(x_1-x_2)\tau^* + \tfrac{3}{4}(x_2+x_3-x_1-\varepsilon)^2$$

$$C_{23,*} = x_1\varepsilon + (x_3-\varepsilon)\tilde{l} + (x_2+x_3-x_1)\tilde{\tau} + \tfrac{1}{2}(x_1-x_2-\varepsilon)\tilde{\tau} + \tfrac{3}{4}(x_2+x_3-x_1)^2$$

$$\text{where } l=\tilde{l} = x_2-x_3,\ \tau=\tau^*=2(x_1-x_2)$$

$$l^*=l+\varepsilon,\ \tilde{\tau} = \tau+2\varepsilon \qquad (2.6)$$

Thus using (2.6) and neglecting $o(\varepsilon)$ terms, we get by assumption

$$C_{13,*} - C_* = \tfrac{1}{2}(x_2+x_3+x_1)\varepsilon \geq 0$$

$$C_{23,*} - C_* = (x_2+3x_3-x_1)\varepsilon \geq 0$$

This proves the optimality of π^*.

Lemma 2.5 : Assume that $x_1 \geq x_2 \geq x_3$, $\mu_i = \mu$, $i=1,2,3$ and be $(1,2)$-rule, then

$$\frac{V_\pi - V_*}{V_*} \leq \frac{1}{9}$$

Proof :

Assume that $\mu = 1$. Both policies and * process queues 1 and 2 until $x_2(t) = x_3(t) = x_3$. Let this time be τ. The same cost incurred up to this time and extra cost is equal to $x_3(x_2-x_3)$. And

$$C_\pi = x_2x_3 = (x_2-x_3)x_3 + x_3^2$$

$$C_* = (x_2-x_3)x_3 + C_*^{\tilde{\pi}}$$

where $C_*^{\tilde{\pi}}$ is the extra cost under * starting with $(x_1(\tau), x_3, x_3)$. Let $C_\pi^{\tilde{\pi}}$ be the extra cost under * starting with (x_3, x_3, x_3). Then we have (see theorem 2.6 below).

$$C_*^{\tilde{\pi}} \geq C_\pi^{\tilde{\pi}} = \tfrac{3}{4}x_3^2, \qquad (2.7)$$

where the equality comes from (2.5). Hence from (2.7),

$$V_\pi - V_* = C_\pi - C_* \leq \tfrac{1}{4}x_3^2 \qquad (2.8)$$

Also note that

$$V_* = C_\pi + C_*^{\tilde{\pi}} \geq C_\pi + C_\pi^{\tilde{\pi}} = \tfrac{1}{2}\sum x_i^2 + \tfrac{3}{4}x_3^2 \qquad (2.9)$$

Thus from (2.8) and (2.9), it can be easily seen that

$$\frac{V_\pi - V_*}{V_*} \leq \frac{C_\pi - C_*}{C_\pi + C_\pi^{\tilde{\pi}}} \leq \frac{1}{9}$$

2.4.3. Relative performance of the best priority policies

We consider a general case in which $N=3$, $m=2$, and μ_i's and π_i's may be different. We will denote the problem by P : (x_i, μ_i), $i=1,2,3$, representing that the initial queue length and the service rate of the jobs in queues i are x_i and μ_i respectively. Let $L(x_i, \mu_i)$ be the line that decreases linearly from x_i to zero with slope μ_i. We say that $L(x_i, \mu_i) \le L(x_j, \mu_j)$ if $L(x_i, \mu_i)$ lies below or equal to $L(x_j, \mu_j)$ for all t.

Consider an another problem $\tilde{P}$: $(\tilde{x}_i, \tilde{\mu}_i)$, $i=1,2,3$. We define a relation between two problems P and $\tilde{P}$.

Definition : we say that $\tilde{P} \le P$ if and only if for each i of $\tilde{P}$ there exists one to one correspondence j in P such that $L(\tilde{x}_i, \tilde{\mu}_i) \le L(x_j, \mu_j)$.

For any policy , we will append superscripts P and $\tilde{P}$ to C_π, C_S and I_π to denote that these quantities are for P or $\tilde{P}$, while I_*^P, C_*^P and $I_\circ^{\tilde{P}}$, $C_\circ^{\tilde{P}}$, will denote the total cost and the extra cost under optimal policies for P and $\tilde{P}$ respectively.

Theorem 2.6 : If $\tilde{P} \le P$, then $C_*^P \ge C_\circ^{\tilde{P}}$.

Proof :

Any policy may be described by a sequence (i_n, t_n, τ_n), $n=1,2,3,\ldots\ldots$ where t_n represents the n th switching time of assignments, and i_n (resp. τ_n) is the queue being freezed at time t_n (resp. duration of the freezing). Then applying the same policy both on P and $\tilde{P}$, that is, if queue i of P is freezed at t_n with duration τ_n under then the corresponding j of $\tilde{P}$ will be freezed at the same time will the same duration, we get $x_i(t) \ge \tilde{x}_j(t)$ for all t for each (i, j) correspondences and hence $C_\pi^P \ge C_\pi^{\tilde{P}}$. (see Figure 3. for illustration). In particular, this holds for the optimal policy for P. Thus we have $C_\circ^P \ge C_\circ^{\tilde{P}}$. where inequality comes from the definition of optimal policy on $\tilde{P}$

We now use Theorem 2.6 and Lemma 2.5 to get an upper bound for the relative performance of the best priority to the optimal one. Let be the best priority policy for a given P. Consider a priority policy π_I which process the longest queues first for $\tilde{P}$.

The idea is then to construct a $\tilde{P}$ such that $\tilde{P} \le P$, $C_\pi^P = C_{\pi I}^{\tilde{P}}$ and $\tilde{\mu}_i = \tilde{\mu}$, $i = 1,2,3$. Accordingly, using theorem 2.6 and noting that $C_S^P \ge C_S^{\tilde{P}}$.

$$C_\pi^P - C_*^P \le C_\pi^P - C_\circ^{\tilde{P}} = C_{\pi I}^{\tilde{P}} - C_\circ^{\tilde{P}} \tag{2.10}$$

$$I_*^P = C_S^P + C_*^P \ge C_S^P + C_\circ^{\tilde{P}} \ge C_S^{\tilde{P}} + C_\circ^{\tilde{P}} = I_\circ^{\tilde{P}}. \tag{2.11}$$

Then by Lemma 2.5, it is obtained from (2.10), (2.11)

$$\frac{C_\pi^P - C_*^P}{C_*^P} \le \frac{C_\pi^P - C_*^P}{C_*^P} \le \frac{C_{\pi 1}^P - C_0^P}{C_0^P} \le \frac{1}{9}.$$

Thus the following theorem will be proved dy constructing a $\widetilde{P}$ such that the above properties holds for a given P.

Theorem 2.7 : Let be the best priority policy. Then

$$\frac{C_\pi^P - C_*^P}{C_*^P} \le \frac{1}{9}.$$

Proof :

By Fact 2, (2,3)-*rule* is never optimal among priority policies. Hence we only need to consider cμ-*rule* and (1,3)-*rule*.

First consider the case π= (1,3)-*rule*.

By Fact 3 we have

$i)\frac{x_3}{\mu_2} \ge \frac{x_1}{\mu_1} \ge \frac{x_2}{\mu_2}$ or $ii)\frac{x_3}{\mu_3} \ge \frac{x_2}{\mu_2} \ge \frac{x_1}{\mu_1}$, $x_3 \ge x_2$.

For both cases, we have $x_3 \ge x_2$. We now construct $\widetilde{P}$ as follows :

1) If $x_3 \ge x_1$, then $\widetilde{\mu}_i = \mu_i$, $i=1,2,3$ and $\widetilde{x}_1 = x_3$, $\widetilde{x}_2 = x_1$, $\widetilde{x}_3 = x_2$.

2) If $i)$ holds with $x_1 \ge x_3$, then $\widetilde{\mu}_i = \mu_i$, $i=1,2,3$ and $\widetilde{x}_1 = \widetilde{x}_2 = x_3$, $\widetilde{x}_3 = x_2$.

3) If $ii)$ holds with $x_1 \ge x_3$, then $\widetilde{\mu}_i = \mu$, where $(\frac{x_3}{\mu})$ = $(\frac{x_1}{\mu_1})$ and $\widetilde{x}_1 = \widetilde{x}_2 = x_3$, $\widetilde{x}_3 = x_2$.

Then it is easy to check that

$C_\pi^P = x_2(\frac{x_1}{\mu_1})$, $C_{\pi 1}^P = x_2(\frac{x_1}{\mu_1})$ and $\widetilde{P} \le P$, $\widetilde{\mu}_i$'s are same.

Secondly, for the case = (1,2)-*rule*, construct $\widetilde{P}$ as follows :

1) If $\frac{x_2}{\mu_2} \le \frac{x_1}{\mu_1}$, then $\widetilde{\mu}_i = \mu_2$, $i=1,2,3$, $\widetilde{x}_1 = x^*$, where $(\frac{x^*}{\mu_2}) = (\frac{x_3}{\mu_3})$, $\widetilde{x}_2 = \max\{x_1, x_3\}$, $\widetilde{x}_3 = \min\{x_1, x_3\}$.

2) If $\frac{x_1}{\mu_1} \le \frac{x_2}{\mu_2}$, $x_2 \ge x_1$, then $\widetilde{\mu}_i = \mu_1$, $i=1,2,3$, and $\widetilde{x}_1 = x_2$, $\widetilde{x}_2 = \max\{x_1, x_3\}$, $\widetilde{x}_3 = \min\{x_1, x_3\}$.

3) If $\frac{x_1}{\mu_1} \le \frac{x_2}{\mu_2}$, $x_1 \ge x_2$, then $\widetilde{\mu}_i = \mu$, $i=1,2,3$, $(\frac{x_1}{\mu_1}) = (\frac{x_2}{\mu})$ and $\widetilde{x}_1 = \widetilde{x}_2 = x_2$, $\widetilde{x}_3 = x_3$.

Then it is easy to check that

$C_\pi^P = x_3 \min\{\frac{x_1}{\mu_1}, \frac{x_2}{\mu_2}\}$ = $C_{\pi 1}^{\widetilde{P}}$ and $\widetilde{P} \le P$, μ_i's are same.

3. A stochastic model

3.1. The model and the problem

We consider a system of three queues and two servers. the service requirement of all jobs are independent and identically distributed by an exponential distribution with parameter μ. We will assume that $\mu=1$ throughout. The problem is how to allocate two servers in order to minimize the expected total cost

$$J = E\{\int_0^{t_f} \sum_i x_i(t)\,dt\}. \tag{3.1}$$

where $x_i(t)$ denote the number of jobs in queue i at time t, $x_i(0) = x_i$ and t_f is the time at which all the jobs are completed. At any time only one serve may be assigned to a queue but pre-emptions are allowed.

We will show that Theorem 2.4 and Lemma 2.5 extends to the stochastic model above.

3.2. Optimality of "Longest Queues First" policy.

Let $x=(x_1, x_2, x_3)$ be the state. A Longest Queues First policy (call this *) is the one which, whenever a server becomes free, assign to it a job in the longest queue among the queues not yet assigned. Let $V^*(x)$ be the total cost under this policy starting with state x.

Definition : We say that $x \sim \tilde{x}$ if and only if $\Pi x = \tilde{x}$. (Π permutation).

Then clearly $V^*(x)=V^*(\tilde{x})$ if $x\sim\tilde{x}$.

Let x^i denote the state in which a job in queue i is removed from x and let $V^*_{i,j,*}(x)$ be the total expected cost starting with x under the policy which serve queues i and j until the next completion and then follows the longest queues first policy thereafter. Then we have

$$V^*_{i,j,*}(x) = \tfrac{1}{2} + \tfrac{1}{2}V^*(x^i) + \tfrac{1}{2}V^*(x^j) \tag{3.2}$$

$V^*(x) = 1+\frac{1}{2}V^*(x^i)+\frac{1}{2}V^*(x^j)$ for x with $x_j=x_k=0$.

Let $|x|$ represent the total number of jobs in state x and let $d(x)$ be defined by $d(x)$ = max x_i- min x_j.

Definition : We say that $x \langle \tilde{x}$ if and only if $x = \tilde{x}$ and $d(x) \le d(\tilde{x})$.

Lemma 3.1 : If $x \langle \tilde{x}$, then $V(x) \le V(\tilde{x})$.

Proof :

Assume that it is true for n-1 jobs. Let i_1, i_2 (j_1, j_2) be the queues selected by * for x ($\tilde{x}$) with $x_{i_1} \ge x_{i_2}$ ($\tilde{x}_{j_1} \ge \tilde{x}_{j_2}$). Then $x^{i_1} \langle \tilde{x}^{j_1}$, $x^{i_2} \langle \tilde{x}^{j_2}$. For the case that $\tilde{x}$ has only one nonempty queue, say, j, we have $x^{i_1} \langle \tilde{x}^{j}$ and $x^{i_2} < \tilde{x}^{j}$. Thus by assumption and (3.2), $V(x) \le V(\tilde{x})$.

Let $\mathcal{Z}$ be the set of states such that $x_i = 0$ for some i. We give an obvious fact without proof in the following corollary.

Corollary : Assume x, $\tilde{x} \in \mathcal{Z}$ and $x \sim \tilde{x}$ or $x \langle \tilde{x}$. Then

$V_\pi(x) = V_\pi(\tilde{x})$ for any π, π.

Theorem 3.2 : The longest Queues First policy is optimal.

Proof :

Assume that $x_1 \ge x_2 \ge x_3$. Then it suffice to show that

$V(x) \le V_{(i,j),*}(x)$ for all x for $(i,j) = (1,3)$ or $(2,3)$.

This is equivalent by (3.2) to showing that

$$V(x^1) \le V(x^3) \text{ and } V(x^2) \le V(x^3) \tag{3.3}$$

Now $x^1 \langle x^3$ and $x^2 \langle x^3$. Thus by Lemma 3.1, (3.3) holds

3.3 Relative performance of the best priority policy to the optimal one.

We first show that a priority rule that starts to process the initially longest queues, is the best policy among priority rules and then determine an upper bound for $\frac{V_\pi(x) - V_{\pi^*}(x)}{V_{\pi^*}(x)}$ applicable for all x, where $V_\pi(x)$ and $V_{\pi^*}(x)$ be the total expected cost under the best priority rule and π^* (Longest Queues First) respectively. Let (i, j)-*rule* denote the priority rule that process the queues i and j first.

3.3.1. Preliminaries

Lemma 3.3 : Assume that $x_1 \ge x_2 \ge x_3$. Then $(1,2)$-*rule* is the best among priority rules.

Proof :

Let $x(t)$ be the state at time t under (1,2)-*rule* and $\tilde{x}(t)$ represent the state under (1,3)-*rule* or (2,3)-*rule* starting with (x_1, x_2, x_3)

First we compare (1,2)-*rule* with (1,3)-*rule*. We couple the completion of a job in queue 1 of $x(t)$ to the completion of a job in queue 1 of $\tilde{x}(t)$ and couple a completion in queue 2 of $x(t)$ with that in queue 2 or 3 of $\tilde{x}(t)$. Let τ be the first time at which $\tilde{x}_1(t)=0$ or $\tilde{x}_3(t)=0$ and τ_2 be the first time such that $x_1(t)=0$ or $x_2(t)=0$ after $\tilde{x}_3(t)$ hit zero first.

If $\tilde{x}_1(t)$ hits zero first at τ, then $x(\tau)=(0, x_2-y, x_3)$ and $\tilde{x}(\tau)=(0, x_2, x_3-y)$ for some $y<x_3$, and hence $x(\tau)<\tilde{x}(\tau)$. If $\tilde{x}_3(t)$ hits zero at τ, we have at τ_2, $x(\tau_2)=(0, x_2-x_3-k, x_3)<\tilde{x}(\tau_2)=(0, x_2-k, 0)$ for some k or $x(\tau_2)=(l-k, 0, x_2)\sim\tilde{x}(\tau_2)=(l-k, x_3, 0)$ for some l, k with $l>k$ depending on whether $x_1(t)$ hits zero first or not.

Hence in any event, by Lemma 3.1 and Corollary, (1,2)-*rule* is shown to be better then (1,3)-*rule*.

Secondly, to compare (1,2)-*rule* with (2,3)-*rule* we couple the completion of a job in queue 2 of $x(t)$ to the completion of a job in queue 2 of $\tilde{x}(t)$ and couple a completion in queue 1 or 3 of $x(t)$ to that queue $\tilde{x}(t)$. Similarly let τ be the first time at which $\tilde{x}_2(t)=0$ or $\tilde{x}_3(t)=0$ and τ_2 be the first time such that $x_1(t)=0$ or $x_2(t)=0$ after $\tilde{x}_3(t)$ hit zero first. Then it can be easily seen that if $\tilde{x}_2(t)$ hits zero at τ, $x(\tau)<\tilde{x}(\tau)$ and otherwise, $x(\tau_2)<\tilde{x}(\tau_2)$ or $x(\tau_2)\sim\tilde{x}(\tau_2)$ depending on whether $x_1(t)$ hit zero first or not.

Hence in any event, again by Lemma 3.1 and Corollary, (1,2)-*rule* is better than (1,3)-*rule*.

Let S be the set of states such that $x_3\geq x_2, x_1$.

Definition : We say that $x \ll \tilde{x}$ if and only if $x, \tilde{x} \in S$, $x_3=\tilde{x}_3$, $x=\tilde{x}$, and $|x|>|\tilde{x}|$, where $|x|=|x_1-x_2|$.

Lemma 3.4 :

Let π be (1,2)-*rule*. Then if $x\ll\tilde{x}$, then $V_\pi(x)\leq V_\pi(\tilde{x})$.

Proof :

It suffice to consider for $\tilde{x}=(x_1, x_2, x_3)$, $x=(x_1+1, x_2-1, x_3)$. Assume it is true for $n-1$ jobs. Note that if $x_2>1$, then $x^1\ll\tilde{x}^1$, $x^2\ll\tilde{x}^2$ and if $x_2=1$, then $d(x^3)<d(\tilde{x}^2)$, $x^1\ll\tilde{x}^1$ and $V_\pi(x^3)=V_\pi(x^3)$, $V_\pi(\tilde{x}^2)=V_\pi(\tilde{x}^2)$. Thus by assumption and Lemma 3.1, we have $V_\pi(x)\leq V_\pi(\tilde{x})$.

Consider a state (x_1, x_2, x_3) and assume that $x_1\geq x_2\geq x_3$. Observe that π and π^* will process same jobs until the time $x(t)$ hits a stste of the form (x'_1, x_2, x_3) for some

$x'_1 \geq x_3$. Thus we only need to calculate $V_\pi(x'_1, x_2, x_3) - V_{\pi^*}(x'_1, x_2, x_3)$ for some $x_1' \geq x_3$ for $V_\pi(x) - V_{\pi^*}(x)$. We will show that for a fixed x_3, this is decreasing in x_1 for $x'_1 \geq x_3$.

To show that $V_\pi(x,n,n) - V_{\pi^*}(x, n, n)$ is decreasing in x for $x \geq n$, we will show that for $x \geq n$,

$$V_\pi(x+1,n,n) - V_{\pi^*}(x+1,n,n) \leq V_\pi(x,n,n) - V_{\pi^*}(x,n,n).$$

From (3.2), it is easy to see that it suffice to show

$$V_\pi(x+1,n-1,n) - V_{\pi^*}(x+1,n-1,n) \leq V_\pi(x,n,n) - V_{\pi^*}(x,n,n).$$

This will be shown true by (P) for $k = 0$ in Lemma 3.5 below.

For convenience, let x denote (x,n,n) and Bx be defined by $Bx = (x_1+1, x_2-1, x_3)$ for a fixed n. Let $(x)_k^\pi$ [resp. $(x)_k^{\pi^*}$], $k=0,1,2,\ldots,(2n-2)$, represent the sequence of states resulted by the consecutive completions of jobs in the second longest queue among the queues chosen by the policy π[resp. π^*] starting with (x,n,n). We define $(Bx)_k^\pi$, $(Bx)_k^{\pi^*}$, $(Bx)_k^\pi$, $(Bx)_k^{\pi^*}$ similiarly for the sequences of states starting with Bx.

Lemma 3.5 :

Let $x = (x,n,n)$, $Bx = (x+1, n-1, n)$. Then for all $x \geq n$

$V_\pi[(Bx)_k^\pi] - V_{\pi^*}[(Bx)_k^{\pi^*}] \leq V_\pi[(x)_k^\pi] - V_{\pi^*}[(x)_k^{\pi^*}]$, $k=0,1,2,\ldots,(2n-2)$ (P)

Proof :

First we show that (P) is true for $x=n$. For this, we use induction on k. For $k=(2n-2)$, (P) can be written as

$$(P) : V_\pi(n+1,0,1) - V_{\pi^*}(n+1,0,1) \leq V_\pi(n,0,2) - V_{\pi^*}(n,1,1)$$

Thus, (P) holds by $V_\pi(n,0,2) = V_{\pi^*}(n,0,2)$ and Lemma 3.1.

Assume that (P) is ture for $k+1$ and show (P) for k. Let $(x)_k^\pi$ [reep, $(Bx)_k^\pi$] $\eta=\pi$, o, $k=0,1,2,\ldots,(2n-2)$ denote the sequences of states resulted by the completion of a job in the longest queue among the queues chosen by π and π^* for $(x)_k^\pi$ [resp. $(Bx)_k^\pi$]. Then we have for $\eta=\pi$, o,

$$V_\eta[(x)_k^\pi] = \tfrac{1}{2} + \tfrac{1}{2} V_\eta[(x)_{k+1}^\pi] + \tfrac{1}{2} V_\eta[(x)_{k+1}^{\pi^*}] \qquad (3.5)$$

$$V_\eta[(Bx)_k^\pi] = \tfrac{1}{2} + \tfrac{1}{2} V_\eta[(Bx)_{k+1}^\pi] + \tfrac{1}{2} V_\eta[(Bx)_{k+1}^{\pi^*}] \qquad (3.5)$$

Now it can be easily seen that for all $k=0,1,2,\ldots,(2n-2)$,

$$(Bn)_k^\pi \langle\langle (n)_k^\pi \text{ and } (n)_k^{\pi^*} \langle\langle (Bn)_k^{\pi^*}.$$

Hence by Lemma 3.1 and 3.4,

$$V_1[(Bn)_k^{k+1}] \le V_1[(n)_k^{k+1}] \text{ and } V_0[(n)_k^{k+1}] \le V_0[(Bn)_k^{k+1}]. \tag{3.6}$$

Then using (3.5), by induction hypothesis and (3.6), (P) can be easily established.

Next we show (P) for $x>n$. We use induction on x.

Assume that (P) are true for $x-1\ge n$ and show that they are true for x. It can be seen easily that for $\eta=\pi$, o and $x>n$,

$$V_\eta[(x)_k] = \tfrac{1}{2} + \tfrac{1}{2}V_\eta[(x)_{k+1}] + \tfrac{1}{2}V_\eta[(x-1)_k] \tag{3.7}$$

$$V_\eta[(Bx)_k] = \tfrac{1}{2} + \tfrac{1}{2}V_\eta[(Bx)_{k+1}] + \tfrac{1}{2}V_\eta[(B(x-1))_k], k=0,1,2,\ldots,(2n-2).$$

By repeated use of (3.7) for $V_\eta[(x)_{k+1}]$ and $V_\eta[(Bx)_{k+1}]$ and induction hypothesis, i.e., (P) is true for $(x-1)$ for all $k=0,1,2\ldots(2n-2)$, it can be easily seen that it suffices to show (P) for k=2n-2, i.e.,

$$V_\pi(x+1,0,1) - V_o(x+1,0,1) \le V_\pi(x,0,2) - V_o(x,1,1). \tag{3.7}$$

Now, (3.7) trivially holds since $V_\pi(x,0,2) = V_o(x,0,2) \ge V_o(x,1,1)$ by Lemma 3.2.

3.3.2 An upper bound

Assume that $x_1\ge x_2\ge x_3$ and let π be (1,2)-rule. Then by the observation and Lemma 3.4,

$$\frac{V_\pi(x) - V_o(x)}{V_o(x)} \le \frac{V_\pi(x_3,x_3,x_3) - V_o(x_3,x_3,x_3)}{V_o(x_3,x_3,x_3)}, \tag{3.8}$$

where $V_o(x) \ge V_o(x_3,x_3,x_3)$ is used.

Consider the evolutions of states under the policy π and π^* starting with (n,n,n). Let $x(l)$, $x^*(l)$ be the state at time l under π and π^* respectively. Let $(i(l), j(l))$, $(i(l), j(l))$ denote the sequences of queues to which servers are assigned at l under π and π^* respectively, where we assume that $x_i(l)\ge x_j(l)$ and $x^*_i(l)\ge x^*_j(l)$ at each l. At each time couple the completion of a job in $i(l)$ to the completion of a job in $i(l)$ and similarly to $j(l)$ until the time of 2n completions. Let this time be τ. Consider a worst realization ω that will give maximum value of $V_\pi(x(\tau,\omega)) - V_o(x^*(\tau,\omega))$ and let $\tilde{x}, \tilde{x}^*$ denote the state $x(\tau,\omega)$ and $x_*(\tau,\omega)$ respectively. Then it is easy to see that

$$V_\pi(\tilde{x}) \le V_\pi(n,0,0) \text{ and } V_o(\tilde{x}^*) \ge V_o(\tilde{x}). \tag{3.9}$$

where $\tilde{x}$ is such that $d(\tilde{x}) \le d(x)$ for all x with $|x| = n$.

Let F(m) be the expected flow time for m parallel jobs on two servers.

Then clearly.

$$F(n) \le F_0(\tilde{x}) \text{ and } F(3n) \le F_0(n,n,n). \tag{3.10}$$

And from (3.8) ~ (3.10),

$$\frac{F_2(x)-F_0(x)}{F_0(x)} \le \frac{F_2(x) - F_0(x^*)}{F_0(n,n,n)} \le \frac{F_2(n,0,0)-F(n)}{F(3n)} \tag{3.11}$$

Now $F(n)$, $F(3n)$ and $F_2(n,0,0)$ can be calculated as

$$F(n) = \frac{1}{2}\frac{n(n+1)}{2} + \frac{1}{2} \tag{3.12}$$

$$F(3n) = \frac{1}{2}\frac{3n(3n+1)}{2} + \frac{1}{2} \tag{3.12}$$

$$F_2(n,0,0) = \frac{n(n+1)}{2}$$

Thus by (3.11) and (3.12), we get

$$\frac{F_2(x) - F_0(x)}{F_0(x)} \le \frac{1}{9}$$

We summarize this in the following theorem.

Theorem 3.6 :

Assume $x_1 \ge x_2 \ge x_3$ and let π be (1,2)-rule. Then

$$\frac{F_2(x) - F_0(x)}{F_0(x)} \le \frac{1}{9}$$

4. Conclusion and Discussion

We have consider a m server system of N competing queues (N>m) without arrivals and investigated the performance of priority rules. A flow approximation analysis shows that a priority rule, $c\mu$-rule is not necessarily best among priority rules and the performance of the best priority rule is within 11% of optimal one for the total cost.

For a stochastic model, we considered a model in which the job processing times are i.i.d. with an exponential distribution. The best priority policy has been determined and the performance of that policy is investigated. If turns out that the same bound is shown to be applicable as well to this model. This support the results obtained by the flow approximation analysis.

Also the model considered here can be viewed as a variant to Weber's problem of

scheduling n jobs on m identical processors by an assignment constraints (only one processor may be assigned to a class, class being the collection of independent jobs with same distribution of processing requirement). Hence it find itself interest.

Suggestions for further works are naturally to

i) consider a model with arrivals and investigate the performance of $c\mu$-rule using similar flow approximation analysis.

ii) consider a model in which the job processing times are independent random variables of exponential distributions with queue dependent parameters and investigate the performance of $c\mu$-rule.

5. References

[1] D.R. Cox and Walter L.Smith, *Queues*, Methuen:New York, Wiley.1961.

[2] C.Buyukkoc,J.Walrand and P.Varaiya,"The $c\mu$-rule Revisited", *J. Appl. Prob.* (To appear)

[3] J.S. Baras, D.J. Ma and A.M. Makowski, "K Comperting Queues with Geometric Requirements and Linear Costs : The $c\mu$-rule is always optimal",Dept of Electrical Engineering, Univ.of Maryland, College park, MD 20742.(Submittied to System and Control Letters)

[4] S.Tu and j. Walrand, "The $c\mu$-rule is Near Optimal", Dept of EECS, Univ.of California, Berkeley. (Preprint)

[5] J.M. Harrison ,"A Priority Queue with Discounted Linear Costs", *Op. Res.* Vol.23,No.2, March-April 1975.

[6] V.V. Movaq and L.A. Ponomaremko,"On the Optimal Assignment of Priorities Depending on the State of a Servicing System With a Finite Number of Waiting Spaces ", *Eng. Cybern.* vol.12,1974,pp.66-72.

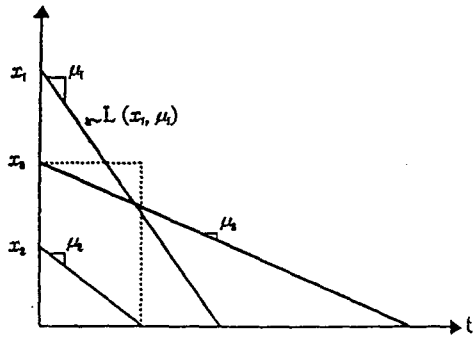
[7] Unknown

[8] S. Ross, *Introduction To Stochastic Dynamic Programming*, Academic Press, 1983.

[9] P.Varaiya,J.Walrand and C. Buyukkoc,"Extensions of the Multi-armed Bandit Problem ", To appear *in IEEE Trans. -AC.*

[10] J.C.Firrin,"Bandit Processes and Dynamic Allocation Indices", *Journal of the Royal Sratistical Society*, Vol.41(1979),pp.556-565.

6. Figures



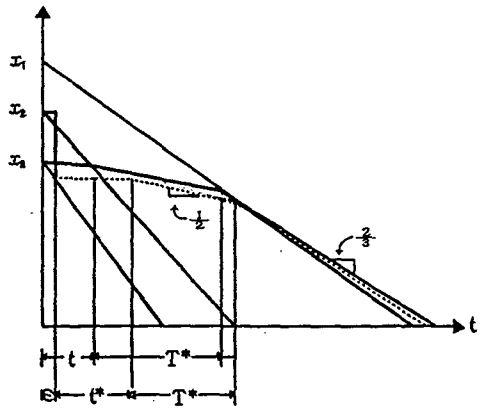
T_i : *Triangular area formed by* $L(x_i, \mu_i)$

$C_i = Processing\ cost = \sum_{i=1} T_i = \frac{1}{2} \sum_{i=1} \left(\frac{x_i}{\mu_i}\right)$

$C_{12} = Extra\ cost\ for\ (1,2)\text{-}rule$

$= Area\ below$ ———

Figure 1 : Illustration of C_3, C_{12}, $L(x_i, \mu_i)$



$l = x_2 - x_3$

$l^* = l + \varepsilon$

$\tau = \tau^* = 2(x_1 - x_2)$

$C_{13,*}$: *Area below* ————————

C_* : *Area bleow* - - - - - - - - - - - -

Figure 2 : Illustration of $C_{13,*}$, C_* for Theorem 4

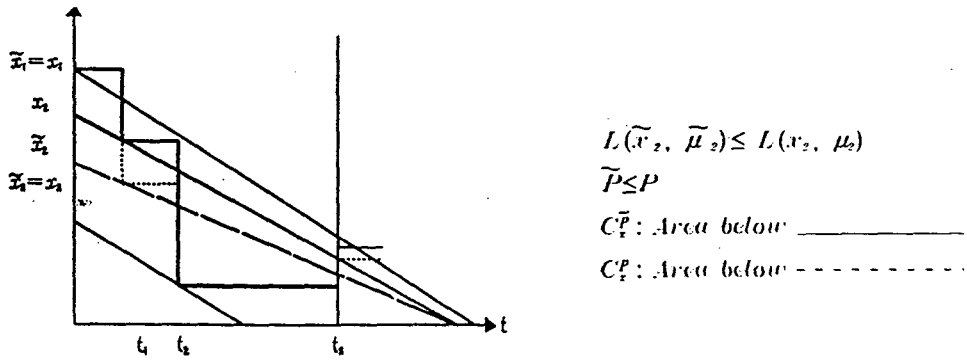


Figure 3 : Illustration of $P \leq \tilde{P} \Rightarrow C_2^{\tilde{P}} \leq C_2^P$.