Scheduling Two Classes of Exponential Jobs on Parallel Processors: Structural Results and Worst Case Analysis

Cheng-Shang Chang, Randolph Nelson*, and Michael Pinedo***

Feb., 1990

Revised Oct., 1990

ABSTRACT

In this paper, we consider scheduling problems with m machines in parallel and two classes of jobs. We assume that all jobs are present at time 0 and there are no further arrivals. The service times of class 1 (2) jobs are independent and exponentially distributed with mean ${\mu_1}^{-1}$ (${\mu_2}^{-1}$). Each class 1 (2) job incurs a cost c_1 (c_2) per unit of time until it leaves the system. The objective is to minimize the expected total cost that is the expected weighted sum of completion times. We show that the optimal policy among all preemptive policies is of threshold type. Based on these structural results, we also show that the ratio of the expected weighted sum of completion times under the $c\mu$-rule to that under the optimal rule is less than 1.71.

Keywords: threshold control, $c\mu$-rule, worst case analysis

* *IBM Research Division, T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, N.Y. 10598.*

** *Dept. of Industrial Engineering and Operations Research, Columbia University, New York, N.Y. 10027.*

I. INTRODUCTION

In this paper, we consider scheduling problems with m machines in parallel and two classes of jobs. We assume that there are x_1 class 1 jobs and x_2 class 2 jobs present at time 0 and there are no further arrivals. The service times of class 1 (2) jobs are independent and exponentially distributed with mean $\mu_1{}^{-1}$ ($\mu_2{}^{-1}$). Each class 1 (2) job incurs a cost c_1 (c_2) per unit of time until it leaves the system. The objective is to minimize the expected total cost that is the expected weighted sum of completion times, i.e.,

$$E \int_0^\infty \sum_{j=1}^{2} c_j Q_j(t) dt,$$

where $Q_j(t)$, $j = 1$ and 2, are the numbers of class j jobs in the system at time t. Clearly, $Q_j(0) = x_j$, $j = 1$ and 2. We show that the optimal policy among all preemptive policies is of threshold type. In other words, the decisions depend on the number of jobs in each class at time t. If $Q_1(t)$ and $Q_2(t)$ are larger than the thresholds, then the $c\mu$-rule, which schedules the m jobs with the largest $c\mu$-value on the m processors, is optimal. Otherwise, the Longest Expected Processing Time first (LEPT) rule, which schedules the m jobs with the largest expected processing times on the m processors, is optimal. The intuition behind this is that the system should first process the most "valuable" jobs when there are a sufficient number in the system whereas it becomes more important to balance load among the machines when there are not enough jobs. Based on these structural results, we show that the ratio of the expected weighted sum of completion times under the $c\mu$-rule (used exclusively) to that under the optimal rule is less than $\frac{2+\sqrt{2}}{2} \approx 1.71$. We also provide an example showing that the ratio is at least $\frac{1+\sqrt{2}}{2} \approx 1.21$ which is the worst case of the scheduling problems with deterministic processing times (Kawaguchi and Kyan (1986)).

In the literature the $c\mu$-rule and the LEPT rule have received a great deal of attention. It has been shown by various authors that for a single processor the $c\mu$-rule minimizes the same objective functions as the one considered in this paper, see Pinedo (1983), Baras, Dorsey and Makowski (1985), and Buyukkoc, Varaiya and Walrand (1985). For parallel processors, however, it is known that the $c\mu$-rule is not optimal. It has been shown that scheduling n jobs according to the LEPT rule on m processors in parallel minimizes the makespan stochastically, even when subject to an arbitrary arrival process, see Van Der Heyden (1981), Frederickson, Bruno and Downey (1981), Weiss (1982) and Weber (1982, 1983). Ross (1983) considered a system consisting of a fixed number of jobs at time 0 that are to be scheduled on two processors. For this setup he showed that, under the agreeability condition where the LEPT rule and the $c\mu$-rule results in the same order of jobs, the $c\mu$-rule minimizes the expected weighted sum of completion times. Kampke (1987) (1989) showed that the model considered by Ross can be extended to m processors with the weaker agreeability condition, if $c_j \geq c_k$, then $c_j\mu_j \geq c_k\mu_k$. Chang, Chao and Pinedo (1989) generalized Ross's result to m processors allowing for a general arrival process and considered various other objectives as well.

This paper is organized as follows: in Section II, we prove the threshold type result by backward induction. Based on these results, we show in Section III that the worst case ratio is bounded by 1.71. Finally, we discuss some generalizations of our results in Section IV.

II. STRUCTURAL RESULTS

The state of the system, denoted by (x_1, x_2), is the number of class 1 and class 2 jobs, x_1 and x_2, respectively, that are in the system at a given time. Since service times are assumed to be memoryless, the optimal policy makes decisions only when a job is completed. We say a schedule follows the $c\mu$-rule at state (x_1, x_2) if it schedules jobs with the maximal value of $c_i\mu_i$ and say a schedule follows the LEPT (Longest Expected Processing Time) rule if it schedules jobs with the maximal value of ${\mu_i}^{-1}$. Without loss of generality, we assume $\mu_1 c_1 \geq \mu_2 c_2$ throughout the paper. We present our main result of this section in the following theorem:

Theorem 2.1. For each x_1 ($x_1 + x_2 > m$), there exists a threshold $\Delta(x_1)$ such that the $c\mu$-rule is optimal for all $x_2 \geq \Delta(x_1)$ and the LEPT rule is optimal for all $x_2 < \Delta(x_1)$. In other words, if the number of class 2 jobs is at least as large as the threshold $\Delta(x_1)$, the $c\mu$-rule is used. If the number of class 2 jobs does not exceed the threshold, then the LEPT rule is used. Similarly, for each x_2 ($x_1 + x_2 > m$), there exists a threshold $\Delta(x_2)$ such that the $c\mu$-rule is optimal for all $x_1 \geq \Delta(x_2)$ and the LEPT rule is optimal for all $x_1 < \Delta(x_2)$.

Remark 2.2. The cases where $\Delta(x_1) = \Delta(x_2) = 0$ are shown by Pinedo (1983), Baras, Dorsey and Makowski (1985) and Buyukkoc, Varaiya and Walrand (1985) for single processor problems. For parallel processor problems, Kampke (1987)(1989) showed that the $c\mu$-rule is optimal if $\mu_1 c_1 \geq \mu_2 c_2$ and $c_1 \geq c_2$. In the case with arrivals, Chang, Chao and Pinedo (1989) showed that the $c\mu$-rule is also optimal if it coincides with the LEPT rule. This is the case where $\mu_1 c_1 \geq \mu_2 c_2$ and ${\mu_1}^{-1} \geq {\mu_2}^{-1}$.

Our proof of Theorem 2.1 is based on dynamic programming and backward induction. The rest of this section is devoted to the proof of this theorem. Let $J(x_1, x_2)$ be the expected weighted sum of completion times under the optimal policy given x_1 class 1 jobs and x_2 class 2 jobs present at time 0. Using a dynamic programming formulation, the value of $J(x_1, x_2)$ can be written as

$$J(x_1, x_2) = \frac{c_1 x_1}{\mu_1} + \frac{c_2 x_2}{\mu_2}$$

if $x_1 + x_2 \leq m$ and

$$J(x_1, x_2) = \min_{l \in A(x_1, x_2)} \{\gamma_l(x_1, x_2) + \alpha_l J(x_1 - 1, x_2) + (1 - \alpha_l) J(x_1, x_2 - 1)\}, \tag{1}$$

if $x_1 + x_2 > m$, where we define

- $A(x_1, x_2) \stackrel{\text{def}}{=} \{l : 0 \leq l \leq m, l \leq x_1, (m - l) \leq x_2\}$
- $q(l) \stackrel{\text{def}}{=} l\mu_1 + (m - l)\mu_2$

- $\gamma_l(x_1, x_2) \stackrel{\text{def}}{=} \frac{c_1 x_1 + c_2 x_2}{q(l)}$
- $\alpha_l \stackrel{\text{def}}{=} \frac{l\mu_1}{q(l)}$

The interpretation of the first case, $x_1 + x_2 \leq m$, is clear since all jobs present in the system can be scheduled on processors. To interpret the second case, assume that the optimal scheduler assigns l processors to class 1 jobs and $m - l$ processors to class 2 jobs in state (x_1, x_2). Then the cost incurred until the next departure from the system is given by $\gamma_l(x_1, x_2)$. With probability α_l the next departure is a class 1 job which leads to state $(x_1 - 1, x_2)$ and with probability $1 - \alpha_l$ the departure is from a class 2 job leading to state $(x_1, x_2 - 1)$.

We let $\pi(x_1, x_2)$ be the number of processors allocated to class 1 jobs in the optimal schedule for state (x_1, x_2). A value of $\pi(x_1, x_2) = l$ thus implies that l class 1 jobs are scheduled and $m-l$ class 2 jobs are scheduled when the system state is (x_1, x_2), $x_1+x_2 > m$. We will say that the optimal policy for state (x_1, x_2) is rule 1 if it schedules as many as class 1 jobs as is possible, i.e. $\pi(x_1, x_2) = min(x_1, m)$ and that the policy rule 2 if it schedules an many class 2 jobs as possible, i.e. $\pi(x_1, x_2) = max(0, m - x_2)$. We note that, since we assume $c_1\mu_1 \geq c_2\mu_2$, the $c\mu$-rule and rule 1 are identical whereas the LEPT rule is the same as rule 1 if ${\mu_1}^{-1} \geq {\mu_2}^{-1}$ and the same as rule 2 if the inequality is reversed. Note that the $c\mu$-rule and the LEPT rule coincide if ${\mu_1}^{-1} \geq {\mu_2}^{-1}$. Thus, in order to conclude that the optimal policy is of the threshold type in Theorem 2.1, we show that (i) for $c_1 \geq c_2$, rule 1 (the $c\mu$-rule) is optimal for all states in Lemma 2.8 and (ii) for $c_1 < c_2$, there exists thresholds which form a switching curve such that rule 1 is optimal if the state is above the switching curve and rule 2 which is the LEPT rule since $c_1 < c_2$ and $\mu_1 c_1 \geq \mu_2 c_2$ implies ${\mu_1}^{-1} \leq {\mu_2}^{-1}$, is optimal if the state is below the switching curve in Corollaries 2.14 and 2.18.

In order to show that the optimal policy is a threshold type policy, we establish monotonic properties of the sign of a certain function, $f(x_1, x_2)$, based on (1). To define

$f(x_1, x_2)$, we first define

$$F_1(x_1, x_2) \stackrel{\text{def}}{=} \gamma_m(x_1, x_2) + J(x_1 - 1, x_2) \tag{2}$$

and

$$F_2(x_1, x_2) \stackrel{\text{def}}{=} \gamma_0(x_1, x_2) + J(x_1, x_2 - 1) \tag{3}$$

for $x_1 + x_2 \geq m + 1, x_1 \geq 1, x_2 \geq 1$. The function $f(x_1, x_2)$ is then given as the difference in these functions, i.e.

$$f(x_1, x_2) \stackrel{\text{def}}{=} F_1(x_1, x_2) - F_2(x_1, x_2). \tag{4}$$

for $x_1 + x_2 \geq m + 1, x_1 \geq 1, x_2 \geq 1$. It is not necessary to define $f(x_1, x_2)$ at the states $(x_1, 0)$, $(0, x_2)$ and (x_1, x_2) with $x_1 + x_2 \leq m$ since there is only one choice at these states. However, in order to be consistent with the following statements in this paper, $f(x_1, x_2)$ at these states are defined to be 0. For the case where $x_1, x_2 \geq m$, equation (4) permits an interpretation as the difference in the incurred costs obtained from using rule 1 and rule 2 policies at state (x_1, x_2). From (4), it is plausible that, for $x_1, x_2 \geq m$, rule 1 (2) at state (x_1, x_2) is optimal if $f(x_1, x_2) \leq 0$ ($f(x_1, x_2) > 0$) and thus the sign of $f(x_1, x_2)$ determines the optimal policy. Lemma 2.3 proves the statement for all states.

Lemma 2.3. For all states (x_1, x_2), rule 1 (2) is optimal if $f(x_1, x_2) \leq 0$ ($f(x_1, x_2) > 0$).

Proof: Let $J_l(x_1, x_2)$ be the incurred cost for assigning $l \in A(x_1, x_2)$ processors to class 1 jobs and then using the optimal policy after the first completion of a job. We can write

$$J_l(x_1, x_2) \stackrel{\text{def}}{=} \gamma_l(x_1, x_2) + \alpha_l J(x_1 - 1, x_2) + (1 - \alpha_l) J(x_1, x_2 - 1). \tag{5}$$

Without loss of generality, assume that $l + 1 \in A(x_1, x_2)$. Some algebra shows that

$$J_{l+1}(x_1, x_2) - J_l(x_1, x_2) = \frac{m\mu_1\mu_2}{q(l+1)q(l)} f(x_1, x_2). \tag{6}$$

The lemma then follows from the fact that the sign of $J_{l+1}(x_1, x_2) - J_l(x_1, x_2)$ is equal to the sign of $f(x_1, x_2)$ and is independent of l. ■

We now derive a fundamental identity of the function $f(x_1, x_2)$.

Lemma 2.4. Let $l_1 \stackrel{\text{def}}{=} \pi(x_1-1, x_2)$ and $l_2 \stackrel{\text{def}}{=} \pi(x_1, x_2-1)$. Then for $x_1+x_2 \geq m+2, x_1 \geq 1, x_2 \geq 1$,

$$f(x_1, x_2) = \alpha_{l_1} f(x_1 - 1, x_2) + (1 - \alpha_{l_2}) f(x_1, x_2 - 1) + B, \tag{7}$$

where $B \stackrel{\text{def}}{=} \frac{\mu_2 c_2 - \mu_1 c_1}{m \mu_1 \mu_2} \leq 0$.

Note that $\alpha_{l_1} = 0$ at the states $(0, x_2)$ and $\alpha_{l_2} = 1$ at the states $(x_1, 0)$. In both cases, (7) is valid no matter what values of $f(x_1, x_2)$ at the states $(0, x_2)$ and $(x_1, 0)$ are defined. The details of the algebra used to prove Lemma 2.4 are found in the Appendix.

Our main results in this section are obtained by establishing monotonic properties of $f(x_1, x_2)$. More precisely, we say that $f(x_1, x_2)$ is **conditionally right-monotonic** (resp. **up-monotonic**) at state (x_1, x_2) if $f(x_1 - 1, x_2) > 0$ or $f(x_1, x_2 - 1) > 0$ implies that $f(x_1, x_2) \leq f(x_1-1, x_2)$ (resp. $f(x_1, x_2) \leq f(x_1, x_2-1)$). We argue in Corollary 2.6 below that the conditionally right-monotonic (resp. up-monotonic) property of $f(x_1, x_2)$ at the state (x_1, x_2) implies that $f(x_1, x_2) \leq 0$ if $f(x_1 - 1, x_2) \leq 0$ (resp. $f(x_1, x_2 - 1) \leq 0$). As a consequence of Lemma 2.3, rule 1 is optimal at (x_1, x_2) if rule 1 is optimal at $(x_1 - 1, x_2)$ (resp. $(x_1, x_2 - 1)$) and $f(x_1, x_2)$ is conditionally right monotone (resp. up-monotone). In Theorem 2.13 we establish that, for all states, $f(x_1, x_2)$ is conditionally right-monotonic and in Theorem 2.17 that it is conditionally up-monotonic (for states (x_1, x_2), $x_2 \leq m$). Our proofs of these monotonic properties are inductive and relied heavily on properties of (7). Properties that we use later in these proofs are stated in the following corollaries.

Corollary 2.5. If $f(x_1 - 1, x_2) \leq 0$ and $f(x_1, x_2 - 1) \leq 0$, then $f(x_1, x_2) \leq 0$ for $x_1 + x_2 \geq m + 2$, $x_1 \geq 1$, $x_2 \geq 1$.

Corollary 2.5 follows from (7) taking into account $0 \leq \alpha_{l_1}, \alpha_{l_2} \leq 1$ and $B \leq 0$.

Corollary 2.6. (i) For a given state (x_1, x_2), $x_1 + x_2 \geq m + 2$, $x_1 \geq 1$, $x_2 \geq 1$, suppose that $f(x_1, x_2 - 1) > 0$ implies that $f(x_1, x_2) \leq f(x_1 - 1, x_2)$. Then $f(x_1, x_2) > 0$ implies

$f(x_1-1,x_2)>0$ or equivalently $f(x_1-1,x_2)\le 0$ implies $f(x_1,x_2)\le 0$. (ii) Similarly, suppose that $f(x_1-1,x_2)>0$ implies that $f(x_1,x_2)\le f(x_1,x_2-1)$. Then $f(x_1,x_2)>0$ implies $f(x_1,x_2-1)>0$ or equivalently $f(x_1,x_2-1)\le 0$ implies $f(x_1,x_2)\le 0$.

Proof: Since the proofs for both cases are similar, we only prove (i). If $f(x_1-1,x_2)\le 0$ and $f(x_1,x_2-1)\le 0$, then $f(x_1,x_2)\le 0$ by Corollary 2.5. If $f(x_1-1,x_2)\le 0$ and $f(x_1,x_2-1)>0$, then it follows from the assumption that $0\ge f(x_1-1,x_2)\ge f(x_1,x_2)$. ■

Corollary 2.6 implies that if $f(x_1,x_2)$ is conditionally right-monotonic (up-monotonic) at state (x_1,x_2), then $f(x_1,x_2)>0$ implies $f(x_1-1,x_2)>0$ ($f(x_1,x_2-1)>0$).

Corollary 2.7. If $f(x_1,x_2-1)\le 0$ and $x_2\ge m$, then $f(x_1,x_2)\le 0$.

Proof: Since $f(x_1,x_2-1)\le 0$ and $B\le 0$, it follows from (7) that we only have to show $\alpha_{l_1}f(x_1-1,x_2)\le 0$. If $f(x_1-1,x_2)\le 0$, then the corollary follows trivially. If $f(x_1-1,x_2)>0$, then rule 2 is used when x_1-1 class 1 jobs and x_2 class 2 jobs are present. In such a case, $l_1=0$ and $\alpha_{l_1}=0$ since $x_2\ge m$. Thus, $\alpha_{l_1}f(x_1-1,x_2)\le 0$. ■

Lemma 2.8. (Agreeability Condition) If $c_1\ge c_2$, then the $c\mu$-rule (rule 1) is optimal for all (x_1,x_2).

Proof: According to Corollaries 2.5, it suffices to show that $f(l+1,m-l)\le 0$ for $l=0,1,\dots,m-1$. Observe that

$$f(l+1,m-l)=\frac{(c_2-c_1)(l\mu_1+(m-l-1)\mu_2)}{m\mu_1\mu_2}+B. \tag{8}$$

Since we assume $\mu_1 c_1\ge\mu_2 c_2$ and $c_1\ge c_2$, $f(l+1,m-l)\le 0$. ■

Corollary 2.9. For state (x_1,x_2) either the $c\mu$-rule or the LEPT rule are optimal.

Proof: *Case* 1. (${\mu_1}^{-1}\ge{\mu_2}^{-1}$) In this case the $c\mu$-rule and the LEPT rule coincide. Since $c_1\mu_1\ge c_2\mu_2$, it follows that $c_1\ge c_2$ and thus from Lemma 2.8 the case is proved.

Case 2. ($\mu_1{}^{-1} < \mu_2{}^{-1}$) In this case the $c\mu$-rule is the same as rule 1 and the LEPT rule is the same as rule 2. The proof follows from Lemma 2.3. ∎

Clearly, the thresholds are simply zero if $c_1 \geq c_2$. Thus, we only have to consider the case $c_1 < c_2$. We now prove three lemmas that form the basis step for the induction of Theorem 2.13 which proves that $f(x_1, x_2)$ is conditionally right-monotonic. Lemma 2.10 establishes that $f(x_1, x_2)$ is conditionally right-monotonic for states where $x_1+x_2 = m+2$, $x_1 \geq 2$ and $x_2 \geq 2$. Lemma 2.11 establishes it for states $(2, x_2)$, $x_2 \geq m$ and Lemma 2.12. for states $(x_1, 1)$, $x_1 \geq m + 1$.

Lemma 2.10. If $c_1 < c_2$, and $f(l + 1, m - l) > 0$ or $f(l + 2, m - l - 1) > 0$, then $f(l + 2, m - l) \leq f(l + 1, m - l)$ for $l = 0, 1, \ldots, m - 2$.

Proof: By (7), we have

$$f(l+2, m-l) = \alpha_{l'_1} f(l+1, m-l) + (1 - \alpha_{l'_2}) f(l+2, m-l-1) + B, \tag{9}$$

where $l'_1 \stackrel{\text{def}}{=} \pi(l+1, m-l)$ and $l'_2 \stackrel{\text{def}}{=} \pi(l+2, m-l-1)$. Consider the following two cases.

Case 1. $f(l + 1, m - l) > 0$.

It follows from (8) that

$$\begin{aligned} f(l+2, m-l-1) &= \frac{(c_2 - c_1)((l+1)\mu_1 + (m-l-2)\mu_2)}{m\mu_1\mu_2} + B \\ &= f(l+1, m-l) + \frac{(c_2 - c_1)(\mu_1 - \mu_2)}{m\mu_1\mu_2}. \end{aligned} \tag{10}$$

Since we assume $c_1 < c_2$ and $\mu_1 c_1 \geq \mu_2 c_2$, we have $\mu_1 \geq \mu_2$ and $f(l + 2, m - l - 1) \geq f(l + 1, m - l) \geq 0$. Thus, $l'_1 = l$ and $l'_2 = l + 1$. Inserting (8) and (10) into (9) yields

$$\begin{aligned} f(l+2, m-l) &= \alpha_l\Big(\frac{(c_2 - c_1)q(l)}{m\mu_1\mu_2} + (B - \frac{(c_2 - c_1)\mu_2}{m\mu_1\mu_2})\Big) \\ &\quad + (1 - \alpha_{l+1})\Big(\frac{(c_2 - c_1)q(l+1)}{m\mu_1\mu_2} + (B - \frac{(c_2 - c_1)\mu_2}{m\mu_1\mu_2})\Big) + B \\ &= f(l+1, m-l) + \alpha_l\Big(\frac{c_1(\mu_2 - \mu_1)}{m\mu_1\mu_2}\Big) + (1 - \alpha_{l+1})\Big(\frac{c_1(\mu_2 - \mu_1)}{m\mu_1\mu_2}\Big) \\ &\leq f(l+1, m-l). \end{aligned}$$

Case 2. $f(l+1, m-l) \leq 0$ and $f(l+2, m-l-1) > 0$.

In this case, $l'_1 = l'_2 = l+1$. Inserting (10) into (9) yields

$$f(l+2, m-l) = f(l+1, m-l) - \left(\frac{(\mu_1 c_1 - \mu_2 c_2)q(l+1) - (m-l-1)\mu_2(c_2 - c_1)(\mu_1 - \mu_2)}{m\mu_1\mu_2 q(l+1)}\right).$$

Since $f(l+1, m-l) \leq 0$, we have $\mu_1 c_1 - \mu_2 c_2 \geq (c_2 - c_1)(m-l-1)\mu_2$ from (8). In conjunction with $q(l+1) \geq (l+1)\mu_1 \geq \mu_1 - \mu_2$, we have $f(l+2, m-l) \leq f(l+1, m-l)$. ∎

Lemma 2.11. If $c_1 < c_2$, $f(1, x_2) > 0$ or $f(2, x_2 - 1) > 0$, then $f(2, x_2) \leq f(1, x_2)$ for all $x_2 \geq m$.

Proof: We will prove this by induction on x_2. We have proved the case when $x_2 = m$ in Lemma 2.10. Assume that the lemma holds for $x_2 - 1$ ($x_2 - 1 \geq m$). From (7), it follows that

$$f(1, x_2) = (1 - \alpha_{l_a}) f(1, x_2 - 1) + B, \tag{11}$$

and

$$f(2, x_2) = \alpha_{l_b} f(1, x_2) + (1 - \alpha_{l_c}) f(2, x_2 - 1) + B, \tag{12}$$

where $l_a \stackrel{\text{def}}{=} \pi(1, x_2 - 1)$, $l_b \stackrel{\text{def}}{=} \pi(1, x_2)$ and $l_c \stackrel{\text{def}}{=} \pi(2, x_2 - 1)$. Consider the following two cases.

Case 1. $f(2, x_2 - 1) > 0$.

From Corollary 2.6 (i) and the induction hypothesis, it follows that $f(1, x_2 - 1) > 0$. In this case, $l_a = l_c = 0$ since $x_2 - 1 \geq m$. Subtracting (11) from (12) yields

$$f(2, x_2) - f(1, x_2) = \alpha_{l_b} f(1, x_2) + (f(2, x_2 - 1) - f(1, x_2 - 1)).$$

Observe that $\alpha_{l_b} f(1, x_2) \leq 0$ since $l_b = 0$ and $\alpha_{l_b} = 0$ if $f(1, x_2) > 0$. The inequality $f(2, x_2) \leq f(1, x_2)$ follows from the induction hypothesis.

Case 2. $f(2, x_2 - 1) \leq 0$ and $f(1, x_2) > 0$.

In this case, $l_b = 0$. From (12), it follows that

$$f(2, x_2) \leq 0 < f(1, x_2). \blacksquare$$

Lemma 2.12 If $c_1 < c_2$, $f(x_1 - 1, 1) > 0$, then $f(x_1, 1) \leq f(x_1 - 1, 1)$ for all $x_1 \geq m + 1$.

Proof: Since $\pi(x_1, 0) = m$, it follows from (7) that

$$f(x_1, 1) = \alpha_{l_1''} f(x_1 - 1, 1) + B,$$

where $l_1'' \stackrel{\text{def}}{=} \pi(x_1 - 1, 1)$. It then follows that $f(x_1, 1) \leq \alpha_{l_1''} f(x_1 - 1, 1) \leq f(x_1 - 1, 1)$. $\blacksquare$

Theorem 2.13. If $c_1 < c_2$, $f(x_1 - 1, x_2) > 0$ or $f(x_1, x_2 - 1) > 0$, then $f(x_1, x_2) \leq f(x_1 - 1, x_2)$ for all $x_1 + x_2 \geq m + 2$, $x_1 \geq 2$ and $x_2 \geq 1$.

Proof: We will prove this by induction on $x_1 + x_2$. Lemma 2.10 provides the basis step for $x_1 + x_2 = m + 2$ and Lemma 2.11 and 2.12 provides proofs for all the boundary steps for all values of $x_1 + x_2 \geq m + 2$. Assume that the lemma holds for $(x_1 - 1, x_2)$ and $(x_1, x_2 - 1)$ as our induction hypothesis. From (7), it follows that

$$f(x_1, x_2) = \alpha_{l_1} f(x_1 - 1, x_2) + (1 - \alpha_{l_2}) f(x_1, x_2 - 1) + B$$

and

$$f(x_1 - 1, x_2) = \alpha_{l_3} f(x_1 - 2, x_2) + (1 - \alpha_{l_4}) f(x_1 - 1, x_2 - 1) + B, \tag{13}$$

where $l_1 \stackrel{\text{def}}{=} \pi(x_1 - 1, x_2)$, $l_2 \stackrel{\text{def}}{=} \pi(x_1, x_2 - 1)$, $l_3 \stackrel{\text{def}}{=} \pi(x_1 - 2, x_2)$ and $l_4 \stackrel{\text{def}}{=} \pi(x_1 - 1, x_2 - 1)$. Again, consider the following two cases.

Case 1. $f(x_1, x_2 - 1) > 0$.

By Corollary 2.6 (i), we have $f(x_1 - 1, x_2 - 1) > 0$. In this case, $l_2 = l_4$. Now we argue that $\alpha_{l_1} f(x_1 - 1, x_2) \leq \alpha_{l_3} f(x_1 - 1, x_2)$. If $f(x_1 - 1, x_2) > 0$, then it follows from Corollary 2.6 (i) that $f(x_1 - 2, x_2) > 0$. Thus, we have $l_1 = l_3$ and $\alpha_{l_1} = \alpha_{l_3}$ if $f(x_1 - 1, x_2) > 0$.

If $f(x_1-1,x_2)\le 0$, then $l_1\ge l_3$ since there are more class 1 jobs in (x_1-1,x_2) than in (x_1-2,x_2). Observe that

$$\alpha_{l_1}-\alpha_{l_3}=\frac{m\mu_1\mu_2(l_1-l_3)}{q(l_1)q(l_3)}\ge 0.$$

Therefore, we have

$$f(x_1,x_2)\le \alpha_{l_3}f(x_1-1,x_2)+(1-\alpha_{l_4})f(x_1,x_2-1)+B. \tag{14}$$

Subtracting (13) from (14) yields

$$\begin{aligned}&f(x_1,x_2)-f(x_1-1,x_2)\\&\le \alpha_{l_3}(f(x_1-1,x_2)-f(x_1-2,x_2))+(1-\alpha_{l_4})(f(x_1,x_2-1)-f(x_1-1,x_2-1)).\end{aligned}$$

Thus, $f(x_1,x_2)\le f(x_1-1,x_2)$ follows from the induction hypothesis and $f(x_1-1,x_2-1)>0$.

Case 2. $f(x_1,x_2-1)\le 0$ and $f(x_1-1,x_2)>0$.

From (7), it follows that

$$f(x_1,x_2)\le \alpha_{l_1}f(x_1-1,x_2)\le f(x_1-1,x_2).\ \blacksquare$$

Corollary 2.14. For each x_2 $(x_1+x_2>m)$, there exists a threshold $\Delta(x_2)$ such that $f(x_1,x_2)\le 0$ if $x_1\ge\Delta(x_2)$ and $f(x_1,x_2)>0$ if $x_1<\Delta(x_2)$.

Proof: This is equivalent to that if $f(x_1-1,x_2)\le 0$, then $f(x_1,x_2)\le 0$. If $f(x_1,x_2-1)\le 0$, then the corollary follows from Corollary 2.5. If $f(x_1,x_2-1)>0$ and $c_1<c_2$, then it follows from Theorem 2.13 that $f(x_1,x_2)\le f(x_1-1,x_2)\le 0$. The case that $c_1\ge c_2$ follows trivially from Lemma 2.8. $\blacksquare$

Lemma 2.15 establishes that $f(x_1,x_2)$ is conditionally up-monotonic for states (x_1,x_2) where $x_1+x_2=m+2$, $x_1\ge 2$ and $x_2\ge 2$. Lemma 2.16 establishes it for states $(x_1,2)$,

$x_1 \geq m$. These two lemmas form the basis step for the induction of Theorem 2.17 which proves that $f(x_1, x_2)$ is conditionally up-monotonic for the states (x_1, x_2), $2 \leq x_2 \leq m$, $x_1 + x_2 \geq m + 2$.

Lemma 2.15. If $c_1 < c_2$, $f(l+1, m-l) > 0$ or $f(l+2, m-l-1) > 0$, then $f(l+2, m-l) \leq f(l+2, m-l-1)$ for $l = 0, 1, \ldots, m-2$.

Proof: From (10), we have $f(l+1, m-l) \leq f(l+2, m-l-1)$. The lemma then follows from Lemma 2.10. ■

Lemma 2.16. If $c_1 < c_2$, $f(x_1, 1) > 0$ or $f(x_1 - 1, 2) > 0$, then $f(x_1, 2) \leq f(x_1, 1)$ for all $x_1 \geq m$.

Proof: We will prove the lemma by induction on x_1. By Lemma 2.15, we have $f(m, 2) \leq f(m, 1)$ if $c_1 < c_2$, $f(m, 1) > 0$ or $f(m-1, 2) > 0$. Assume that the lemma holds for $x_1 - 1$ $(x_1 - 1 \geq m)$ as the induction hypothesis. Consider the following three cases.

Case 1. $f(x_1 - 1, 2) \leq 0$ and $f(x_1, 1) > 0$.

It follows from Theorem 2.13 that $f(x_1, 2) \leq f(x_1 - 1, 2) \leq 0$. Thus, we have $f(x_1, 2) \leq 0 < f(x_1, 1)$.

Case 2. $f(x_1 - 1, 2) > 0$ and $f(x_1, 1) > 0$.

Again, by Theorem 2.13, we have $f(x_1 - 1, 1) > 0$. From (7), it follows that

$$f(x_1, 2) = \alpha_{m-2} f(x_1 - 1, 2) + (1 - \alpha_{m-1}) f(x_1, 1) + B, \tag{15}$$

and

$$f(x_1, 1) = \alpha_{m-1} f(x_1 - 1, 1) + B. \tag{16}$$

Subtracting (16) from (15) yields

$$\begin{aligned} f(x_1, 2) - f(x_1, 1) &= \alpha_{m-2}(f(x_1 - 1, 2) - f(x_1 - 1, 1)) \\ &+ (1 - \alpha_{m-1})(f(x_1, 1) - f(x_1 - 1, 1)) - \frac{\mu_2}{q(m-1)} \frac{2(\mu_1 - \mu_2)}{q(m-2)} f(x_1 - 1, 1). \end{aligned}$$

Since $c_1 < c_2$ and $\mu_1 c_1 \geq \mu_2 c_2$, we have $\mu_1 \geq \mu_2$. It follows from the induction hypothesis and Theorem 2.13 that $f(x_1, 2) \leq f(x_1, 1)$ in this case.

Case 3. $f(x_1 - 1, 2) > 0$ and $f(x_1, 1) \leq 0$

By Corollary 2.6 (ii), we still have $f(x_1 - 1, 1) > 0$. Since $f(x_1, 1) \leq 0$, we rewrite (15) as the inequality

$$f(x_1, 2) \leq \alpha_{m-2} f(x_1 - 1, 2) + B, \tag{17}$$

Subtracting (16) from (17) yields

$$\begin{aligned}
&f(x_1, 2) - f(x_1, 1) \\
&\leq \alpha_{m-2}(f(x_1 - 1, 2) - f(x_1 - 1, 1)) - \frac{m\mu_1\mu_2}{q(m-1)q(m-2)} f(x_1 - 1, 1) \\
&\leq 0. \blacksquare
\end{aligned}$$

Theorem 2.17. If $c_1 < c_2$, $f(x_1 - 1, x_2) > 0$ or $f(x_1, x_2 - 1) > 0$, then $f(x_1, x_2) \leq f(x_1, x_2 - 1)$ for all $2 \leq x_2 \leq m, x_1 + x_2 \geq m + 2$.

Proof: We will prove this by induction on $x_1 + x_2$. Since we have proved that the lemma holds for all the boundary states in Lemmas 2.15 and 2.16, we assume that the lemma holds for $x_1 + x_2 - 1$ as the induction hypothesis. From (7), it follows that

$$f(x_1, x_2) = \alpha_{l_1} f(x_1 - 1, x_2) + (1 - \alpha_{l_2}) f(x_1, x_2 - 1) + B$$

and

$$f(x_1, x_2 - 1) = \alpha_{l_4} f(x_1 - 1, x_2 - 1) + (1 - \alpha_{l_5}) f(x_1, x_2 - 2) + B, \tag{18}$$

where $l_1 \stackrel{\text{def}}{=} \pi(x_1 - 1, x_2)$, $l_2 \stackrel{\text{def}}{=} \pi(x_1, x_2 - 1)$, $l_4 \stackrel{\text{def}}{=} \pi(x_1 - 1, x_2 - 1)$ and $l_5 \stackrel{\text{def}}{=} \pi(x_1, x_2 - 2)$. We need to consider the following three cases.

Case 1. $f(x_1 - 1, x_2) \leq 0$ and $f(x_1, x_2 - 1) > 0$.

This follows the same argument as in the proof of Lemma 2.16, Case 1.

Case 2. $f(x_1 - 1, x_2) > 0$ and $f(x_1, x_2 - 1) > 0$.

From Theorem 2.13, it follows that $f(x_1-1,x_2-1)>0$. By Corollary 2.6 (ii), we have $f(x_1,x_2-2)>0$. Since $x_2\le m$, we have $l_1=m-x_2$, $l_2=l_4=m-(x_2-1)$ and $l_5=m-(x_2-2)$. Subtracting (18) from (7) yields

$$
\begin{aligned}
&f(x_1,x_2)-f(x_1,x_2-1)\\
&=\alpha_{m-x_2}(f(x_1-1,x_2)-f(x_1-1,x_2-1))\\
&+(1-\alpha_{m-x_2+2})(f(x_1,x_2-1)-f(x_1,x_2-2))\\
&+\frac{m\mu_1\mu_2}{q(m-x_2+1)q(m-x_2)}(f(x_1,x_2-1)-f(x_1-1,x_2-1))\\
&-\frac{m\mu_1\mu_2}{q(m-x_2+1)}\frac{2(\mu_1-\mu_2)}{q(m-x_2+2)q(m-x_2)}f(x_1,x_2-1)\\
&\le 0.
\end{aligned}
$$

Case 3. $f(x_1-1,x_2)>0$ and $f(x_1,x_2-1)\le 0$.

By the same argument as in the proof of Theorem 2.13, Case 1, rewrite (7) as the inequality

$$
f(x_1,x_2)\le\alpha_{l_4}f(x_1-1,x_2)+(1-\alpha_{l_2})f(x_1,x_2-1)+B. \tag{19}
$$

Subtracting (18) from (19) yields

$$
\begin{aligned}
&f(x_1,x_2)-f(x_1,x_2-1)\le\alpha_{l_4}(f(x_1-1,x_2)-f(x_1-1,x_2-1))\\
&+(1-\alpha_{l_2})f(x_1,x_2-1)-(1-\alpha_{l_5})f(x_1,x_2-2).
\end{aligned}
$$

Clearly, if $f(x_1,x_2-2)>0$, then the lemma holds by the induction hypothesis and the assumption that $f(x_1,x_2-1)\le 0$. If $f(x_1,x_2-2)\le 0$, we have $l_2=l_5$. Since $f(x_1-1,x_2)>0$, we have $f(x_1-1,x_2-1)>0$ by Corollary 2.6 (ii). Thus, the lemma also holds by the induction hypothesis. ∎

Corollary 2.18. For each x_1 $(x_1+x_2>m)$, there exists a threshold $\Delta(x_1)$ such that $f(x_1,x_2)\le 0$ if $x_2\ge\Delta(x_1)$ and $f(x_1,x_2)>0$ if $x_2<\Delta(x_1)$.

Proof: It follows from Theorem 2.17 and Corollary 2.5 that $f(x_1,x_2)\le 0$ if $f(x_1,x_2-1)\le 0$ for all $x_2\le m$ and $c_1<c_2$. In conjunction with Corollary 2.7, we complete the proof for the case $c_1<c_2$. Again, the case $c_1\ge c_2$ follows from Lemma 2.8. ∎

III. WORST CASE ANALYSIS

From the threshold type result in the previous section, it follows that the $c\mu$-rule is optimal if x_1 and x_2 are sufficiently large. Let $J_{c\mu}(x_1, x_2)$ be the expected weighted sum of completion times using the $c\mu$-rule for all states. Let $R(x_1, x_2, \mu_1, \mu_2, c_1, c_2, m)$ be the ratio of the expected weighted sum of completion times under the $c\mu$-rule to that under the optimal rule and R be the supremum of $R(x_1, x_2, \mu_1, \mu_2, c_1, c_2, m)$ over all its arguments. We show in the following theorem that R is bounded by $\frac{2+\sqrt{2}}{2} \approx 1.71$. We also provide an example showing that $R \geq \frac{1+\sqrt{2}}{2} \approx 1.21$ which is the worst case of the scheduling problems with deterministic service times. We note that if the LEPT rule is optimal for (x_1, x_2) then it is optimal for all $(x_1', x_2') \leq (x_1, x_2)$.

Theorem 3.1. $R \leq \frac{2+\sqrt{2}}{2} \approx 1.71$.

The proof of Theorem 3.1 is based on the following lemmas. Since we assume $\mu_1 c_1 \geq \mu_2 c_2$, the $c\mu$-rule corresponds to rule 1 which schedules class 1 jobs before class 2 jobs. When seeking the worst case it suffices to consider certain cases of $(x_1, x_2, \mu_1, \mu_2, c_1, c_2, m)$. Since the $c\mu$-rule is optimal if $c_1 \geq c_2$ (Lemma 2.8), the worst case occurs when $c_1 < c_2$. In conjunction with $\mu_1 c_1 \geq \mu_2 c_2$, we only have to consider the case $\mu_1 \geq \mu_2$.

Lemma 3.2. The worst case cannot occur at a state where $f(x_1, x_2) \leq 0$. In other words, worst cases occur when (x_1, x_2) is under the switching curve and the LEPT rule (rule 2) is optimal for all $(x_1', x_2') \leq (x_1, x_2)$.

Proof: If $f(x_1, x_2) \leq 0$, then it follows from (1) that

$$J(x_1, x_2) = \gamma_{l_0}(x_1, x_2) + \alpha_{l_0} J(x_1 - 1, x_2) + (1 - \alpha_{l_0}) J(x_1, x_2 - 1) \tag{20}$$

and

$$J_{c\mu}(x_1, x_2) = \gamma_{l_0}(x_1, x_2) + \alpha_{l_0} J_{c\mu}(x_1 - 1, x_2) + (1 - \alpha_{l_0}) J_{c\mu}(x_1, x_2 - 1), \tag{21}$$

where $l_0 = min(x_1, m)$. Observe that

$$\frac{J_{c\mu}(x_1,x_2)}{J(x_1,x_2)} \le \frac{\alpha_{l_0} J_{c\mu}(x_1-1,x_2) + (1-\alpha_{l_0}) J_{c\mu}(x_1,x_2-1)}{\alpha_{l_0} J(x_1-1,x_2) + (1-\alpha_{l_0}) J(x_1,x_2-1)}$$
$$\le max\left(\frac{J_{c\mu}(x_1-1,x_2)}{J(x_1-1,x_2)}, \frac{J_{c\mu}(x_1,x_2-1)}{J(x_1,x_2-1)}\right).$$

Thus, the worst case cannot occur at a state where $f(x_1, x_2) \le 0$. From the threshold type result, it follows that worst case occurs only when the optimal rule is the LEPT rule. ∎

By Lemma 3.2, it suffices to consider such pairs (x_1, x_2) for which the LEPT rule is optimal (at the beginning and at all subsequent steps). Let $Q_{c\mu,i}(t)$ $(Q_{L,i}(t))$ be the number of class i jobs in the system at time t using the $c\mu$-rule (LEPT rule) for all states. Let $z_{c\mu,i} \stackrel{\text{def}}{=} E\int_0^\infty Q_{c\mu,i}(t)dt$ and $z_{L,i} \stackrel{\text{def}}{=} E\int_0^\infty Q_{L,i}(t)dt$. Then

$$R(x_1, x_2, \mu_1, \mu_2, c_1, c_2, m) = \frac{c_1 z_{c\mu,1} + c_2 z_{c\mu,2}}{c_1 z_{L,1} + c_2 z_{L,2}}.$$

Lemma 3.3. The worst case occurs when $\mu_1 c_1 = \mu_2 c_2$.

Proof: Consider two sample paths generated by the $c\mu$-rule and the LEPT rule. Since the class 1 jobs are executed first under the $c\mu$-rule, we can construct the sample paths such that the completion time of the j-th class 1 job under the $c\mu$-rule is smaller than that under the LEPT rule for all $j = 1, \ldots, x_1$. Thus, $z_{c\mu,1} \le z_{L,1}$. Similarly, we also have $z_{c\mu,2} \ge z_{L,2}$. Therefore, $\frac{\partial R(x_1,x_2,\mu_1,\mu_2,c_1,c_2,m)}{\partial c_1} \le 0$. Since $c_1 \ge c_2 \frac{\mu_2}{\mu_1}$, we conclude that the worst case occurs when $\mu_1 c_1 = \mu_2 c_2$. ∎

Let $V_{c\mu,1}(t)$ $(V_{c\mu,2}(t))$ be the amount of work for class 1 (2) jobs at time t under the $c\mu$-rule and $V_{c\mu}(t) \stackrel{\text{def}}{=} V_{c\mu,1}(t) + V_{c\mu,2}(t)$ be the total amount of work in the system at time t under the $c\mu$-rule. Define similarly the quantities, $V_L(t), V_{L,1}(t)$ and $V_{L,2}(t)$ under the LEPT rule. Based on the memoryless property of exponential random variables, $EV_{c\mu,i}(t) = \frac{1}{\mu_i} EQ_{c\mu,i}(t)$ and $EV_{L,i}(t) = \frac{1}{\mu_i} EQ_{L,i}(t)$, $i = 1$ and 2. Thus,

$$R(x_1, x_2, \mu_1, \mu_2, c_1, c_2, m) = \frac{\mu_1 c_1 E\int_0^\infty V_{c\mu,1}(t)dt + \mu_2 c_2 E\int_0^\infty V_{c\mu,2}(t)dt}{\mu_1 c_1 E\int_0^\infty V_{L,1}(t)dt + \mu_2 c_2 E\int_0^\infty V_{L,2}(t)dt}.$$

Corollary 3.4. By Lemma 3.3, it suffices to consider the case $\mu_1 c_1 = \mu_2 c_2 = 1$ and thus

$$R = \sup_{x_1, x_2, \mu_1 \geq \mu_2, m} \left(\frac{E\int_0^\infty V_{c\mu}(t)dt}{E\int_0^\infty V_L(t)dt}\right).$$

Lemma 3.5. For any x_1, x_2, μ_1, μ_2 and m,

$$E\int_0^\infty V_{c\mu}(t)dt \leq G_m(x_1,\mu_1) + G_m(x_2,\mu_2) + \frac{x_1 x_2}{m\mu_1\mu_2},$$

where

$$G_m(x,\mu) = \begin{cases} \frac{1}{2m}(\frac{1}{\mu})^2(x^2+x) + \frac{(m-1)}{2}(\frac{1}{\mu})^2, & \text{if } x \geq m-1, \\ x(\frac{1}{\mu})^2, & \text{if } x < m-1. \end{cases}$$

Proof: Our proof will concentrate on the case where $x_1 \geq m-1$ and $x_2 \geq m-1$. The other cases can be proved similarly. The amount of work for class 1 jobs at time 0 can be constructed as $m\sum_{j=1}^{x_1+1-m}\tilde{S}_{1,j} + \sum_{j=1}^{m-1} S_{1,j}$, where $\tilde{S}_{1,j}$'s are independent and exponentially distributed r.v.'s with mean $1/m\mu_1$ and $S_{1,j}$'s are independent and exponentially distributed r.v.'s with mean ${\mu_1}^{-1}$. Let $\tau_1 \stackrel{\text{def}}{=} \sum_{j=1}^{x_1+1-m}\tilde{S}_{1,j}$. Since class 1 jobs are executed first under the $c\mu$-rule, clearly, τ_1 is the time to complete the first x_1+1-m class 1 jobs. Thus,

$$\int_0^{\tau_1} V_{c\mu,1}(t)dt = \frac{1}{2}(m\sum_{j=1}^{x_1+1-m}\tilde{S}_{1,j} + 2\sum_{j=1}^{m-1}S_{1,j})(\sum_{j=1}^{x_1+1-m}\tilde{S}_{1,j}). \tag{22}$$

After τ_1, there are $m-1$ class 1 jobs left and they are exponentially distributed with mean ${\mu_1}^{-1}$ by the memoryless property. Thus,

$$\int_{\tau_2}^\infty V_{c\mu,1}(t)dt = \sum_{j=1}^{m-1}\frac{1}{2}(S_{1,j})^2.$$

Therefore,

$$E\int_0^\infty V_{c\mu,1}(t)dt = \frac{1}{2m}(\frac{1}{\mu_1})^2(x_1^2+x_1) + \frac{(m-1)}{2}(\frac{1}{\mu_1})^2. \tag{23}$$

If $x_1 < m-1$, then all the class 1 jobs are served at time 0 and

$$E\int_0^\infty V_{c\mu,1}(t)dt = E\sum_{j=1}^{x_1}\frac{1}{2}(S_{1,j})^2 = x_1(\frac{1}{\mu_1})^2. \tag{24}$$

The case for class 2 jobs is more complicated because not all the processors are available for class 2 jobs at the same time. In general there is not a closed form expression for $E\int_{\tau_1}^{\infty} V_{c\mu,2}(t)dt$. Our approach is to bound this value by considering an auxiliary system where all of the processors are available for class 2 jobs at the same time. To do this, we first consider a lower bound for the time the processors could be available. Let τ_1' be the time to finish all class 1 jobs assuming a job can be processed by more than one processor at the same time, i.e.,

$$\tau_1' = \begin{cases} \tau_1 + \frac{1}{m}\sum_{j=1}^{m-1} S_{1,j}, & \text{if } x_1 \geq m-1, \\ \frac{1}{m}\sum_{j=1}^{x_1} S_{1,j}, & \text{if } x_1 < m-1. \end{cases}$$

Consider an auxiliary system in which class 2 jobs are assumed to start at τ_1'. Under this assumption, the cost (the integral over the amount of work from 0 to infinity) for class 2 jobs can be computed using the same technique as that for class 1 jobs, i.e. the amount of work at τ_1' for class 2 jobs in the auxiliary system is $m\sum_{j=1}^{x_2+1-m}\tilde{S}_{2,j} + \sum_{j=1}^{m-1} S_{2,j}$, where $\tilde{S}_{2,j}$'s are independent and exponentially distributed r.v.'s with mean $1/m\mu_2$ and $S_{2,j}$'s are independent and exponentially distributed r.v.'s with mean ${\mu_2}^{-1}$. The expected cost for class 2 jobs in the auxiliary system is $\frac{x_2}{\mu_2}E\tau_1' + G_m(x_2, \mu_2)$, where $E\tau_1' = \frac{x_1}{m\mu_1}$.

Now consider the relationship between the expected cost in the original system and that in the auxiliary system. Without loss of generality, assume that, in the original system, Processor m is the processor that completes the (x_1+1-m)-th class 1 job and thus Processor m is available for class 2 jobs at τ_1. Processor i, in the original system, is available at time $\tau_1 + S_{1,i}$. We create a mapping between the two systems by assigning the j-th class 2 job to Processor i in the original system if it is assigned to Processor i in the auxiliary system. Let V_i, $i = 1, \ldots, m$, be the amount of work for class 2 jobs assigned to Processor i in the auxiliary system. Clearly, V_i's are exchangeable r.v.'s and are *independent* of class 1 jobs. Moreover, V_i is also the amount of work for class 2 jobs assigned to Processor i in the original system that follows this same processor assignment. It is thus the case that V_i, $i = 1, \ldots, m-1$, in the original system, are shifted in time by

$S_{1,i} - \frac{1}{m}\sum_{j=1}^{m-1} S_{1,j}$. V_m is shifted in time by $\frac{1}{m}\sum_{j=1}^{m-1} S_{1,j}$. The difference of the cost for class 2 jobs between the auxiliary system and the original system that follows this same processor assignment is therefore

$$\begin{aligned}
&\sum_{i=1}^{m-1} V_i(S_{1,i} - \frac{1}{m}\sum_{j=1}^{m-1} S_{1,j}) - V_m \frac{1}{m}\sum_{j=1}^{m-1} S_{1,j} \\
&= \sum_{i=1}^{m-1} S_{1,i}(V_i - \frac{1}{m}\sum_{j=1}^{m} V_j).
\end{aligned} \tag{25}$$

Since the values of the V_i's are exchangeable and independent of $S_{1,i}$'s, the expectation of this difference is 0. Since the processor assignment for class 2 jobs in the original system can be rearranged to further minimize the cost, we have that the expected cost for class 2 jobs under the $c\mu$-rule is smaller than that in the auxiliary system. ∎

Lemma 3.6. For any x_1, x_2, μ_1, μ_2 and m,

$$E\int_0^\infty V_L(t)dt \geq G_m(x_1,\mu_1) + G_m(x_2,\mu_2) + \frac{(x_2+1-m)^+ x_1}{m\mu_1\mu_2},$$

where $x^+ \stackrel{\text{def}}{=} max(0, x)$.

Proof: Since class 2 jobs are executed first under the LEPT rule, the cost for class 2 jobs is $G_m(x_2, \mu_2)$. Let τ_2 be the first time that there is a processor available for class 1 jobs. Clearly, if $x_2 \geq m-1$, τ_2 is the time to complete the first $x_2 + 1 - m$ class 2 jobs. If $x_2 < m-1$, $\tau_2 = 0$. Consider an auxiliary system which starts class 1 jobs at τ_2. By the sample path argument, the cost for class 1 jobs under the LEPT rule is larger than that in the auxiliary system. The expected cost for class 1 jobs in the auxiliary system is $\frac{x_1}{\mu_1}E\tau_2 + G_m(x_1, \mu_1)$, where $E\tau_2 = \frac{(x_2+1-m)^+}{m\mu_2}$. ∎

Corollary 3.7. It follows from Lemmas 3.5 and 3.6 that

$$R \leq \sup_{\mu_1 \geq \mu_2} \{R'(x_1, x_2, \mu_1, \mu_2, m)\},$$

where

$$R'(x_1,x_2,\mu_1,\mu_2,m) \stackrel{\text{def}}{=} \left(\frac{G_m(x_1,\mu_1)+G_m(x_2,\mu_2)+\frac{x_1x_2}{m\mu_1\mu_2}}{G_m(x_1,\mu_1)+G_m(x_2,\mu_2)+\frac{(x_2+1-m)^+x_1}{m\mu_1\mu_2}}\right).$$

Lemma 3.8. $R'(x_1,x_2,\mu_1,\mu_2,m)$ is maximized when $x_2 = m-1$.

Proof: Rewrite

$$R'(x_1,x_2,\mu_1,\mu_2,m) = 1 + \frac{\frac{x_1}{m\mu_1\mu_2}(x_2-(x_2+1-m)^+)}{G_m(x_1,\mu_1)+G_m(x_2,\mu_2)+\frac{(x_2+1-m)^+x_1}{m\mu_1\mu_2}}. \tag{26}$$

If $x_2 \geq m-1$, then the denominator is increasing in x_2 while the numerator is not. Thus, we only have to consider the case $x_2 \leq m-1$. In this case, $G_m(x_2,\mu_2) = x_2(\frac{1}{\mu_2})^2$ and $(x_2+1-m)^+ = 0$. Therefore,

$$R'(x_1,x_2,\mu_1,\mu_2,m) - 1 = \frac{x_1}{m\mu_1\mu_2}\left(\frac{x_2}{G_m(x_1,\mu_1)+x_2(\frac{1}{\mu_2})^2}\right). \tag{27}$$

The proof then follows since the function within the parenthesis in (27) is increasing in x_2.

■

If $x_1 \leq m-1$, then $G_m(x_1,\mu_1) = x_1(\frac{1}{\mu_1})^2$, and again, it follows that the function within the parenthesis in (27) is increasing in x_1 if $x_1 \leq m-1$. Thus, we only have to consider the case $x_1 \geq m-1$. Let $a \stackrel{\text{def}}{=} \frac{\mu_1}{\mu_2}$ and $b \stackrel{\text{def}}{=} \frac{x_1}{a}$. Replacing $x_2 = m-1$ in (27) yields

$$R - 1 \leq \sup_{a\geq 1}\{f(a,b,m)\}, \tag{28}$$

where

$$f(a,b,m) \stackrel{\text{def}}{=} \frac{m-1}{m}\frac{b}{\frac{1}{2m}(b^2+b\frac{1}{a})+\frac{m-1}{2}(\frac{1}{a})^2+(m-1)}.$$

Lemma 3.9. $f(a,b,m) \leq \frac{\sqrt{2}}{2}$ for all a, b and m and thus $R \leq 1+\frac{\sqrt{2}}{2} \approx 1.71$.

Proof: Since $f(a,b,m)$ is increasing in a, keeping b and m fixed and letting $a \to \infty$ yields

$$f(a,b,m) \leq \frac{m-1}{m}\frac{b}{\frac{1}{2m}b^2+m-1}. \tag{29}$$

By direct calculus, it can be shown that the maximum of the right hand side of (29) occurs when $b = \sqrt{2m(m-1)}$. Replacing b by $\sqrt{2m(m-1)}$ in (29) yields

$$f(a,b,m) \leq \sqrt{\frac{m-1}{2m}} \leq \frac{\sqrt{2}}{2}. \ \blacksquare$$

In the following example, we show that the worst case of scheduling exponential jobs is worse than that of scheduling deterministic jobs.

Example 3.10. Consider the case that the processing times of class 1 jobs are infinitesimally small ($\mu_1 \to \infty$) and the total amount of work for class 1 jobs is m ($\frac{x_1}{\mu_1} \to m$). Let $x_2 \leq m-1$. In this case, the inequality in Lemma 3.5 becomes an equality since class 1 jobs fit perfectly. Thus,

$$J_{c\mu} = \frac{1}{2}m + x_2(\frac{1}{\mu_2})^2 + \frac{x_2}{\mu_2}.$$

By assigning class 1 jobs to the processors not used by class 2 jobs, we have in this case that

$$J \leq G_{m-x_2}(x_1,\mu_1) + x_2(\frac{1}{\mu_2})^2 = \frac{1}{2}\frac{m^2}{m-x_2} + x_2(\frac{1}{\mu_2})^2.$$

Replacing $\alpha = \frac{m-x_2}{m}$ and $y = \frac{1}{\mu_2}$ yields

$$R \geq \sup_{\alpha,y}\{1 + \frac{y - \frac{1}{2\alpha}}{y^2 + \frac{1}{2\alpha(1-\alpha)}}\}. \qquad (30)$$

It can be shown by calculus that the maximum of the ratio in (30) is $\frac{1}{2}+\sqrt{\frac{1}{2}} \approx 1.2071$ which is the worst case of the scheduling problem with deterministic service times. Moreover, we solve (1) numerically showing that the ratio could be as high as 1.21228 when $m = 106, x_1 = 30000, x_2 = 75, \mu_1 = \frac{x_1}{m}, \mu_2 = \sqrt{2}-1, c_1 = {\mu_1}^{-1}$ and $c_2 = {\mu_2}^{-1}$.

IV. Discussion

We expect that the threshold type result continues to hold for the scheduling problems with arrivals. Moreover, one expects intuitively that the thresholds obtained by the scheduling problems without arrivals are the upper bounds for the thresholds for the problems with arrivals. For the problem with more than two classes of jobs, the result for the existence of simple thresholds between the $c\mu$-rule and the LEPT rule is no longer valid. Consider a scheduling problem with three jobs and two processors. Assume $\mu_1 > \mu_2 >> \mu_3$ and $\mu_1 c_1 = \mu_2 c_2 + \epsilon_1 = \mu_3 c_3 + \epsilon_1 + \epsilon_2$, where ϵ_1 and ϵ_2 are very small positive numbers. Since the $c\mu$ values of these three jobs are very close, the effect of load balancing would be crucial and one would expect that Job 3 should have the highest priority. Since the service time of Job 3 is much longer than the other two jobs, with high probability Job 1 and Job 2 will be executed on the same processor. In this case, the $c\mu$ values decide the execution sequence of Job 1 and Job 2. Thus, the optimal rule does not follow either the $c\mu$-rule or the LEPT rule. More precisely, let $\mu_1 = 505, c_1 = 2, \mu_2 = 200, c_2 = 5, \mu_3 = 1$ and $c_3 = 999$. The expected weighted sum of completion times under the $c\mu$-rule is $\frac{c_1}{\mu_1} + \frac{c_2}{\mu_2} + \frac{c_3}{\mu_1+\mu_2} + \frac{c_3}{\mu_3} = 1000.446$. The expected weighted sum of completion times under the LEPT rule is $\frac{c_3}{\mu_3} + \frac{c_2}{\mu_2} + \frac{c_1}{\mu_3+\mu_2} + \frac{c_1}{\mu_1} = 999.0389$. However, the expected weighted sum of completion times under the policy that executes Job 2 last is $\frac{c_1}{\mu_1} + \frac{c_3}{\mu_3} + \frac{c_2}{\mu_1+\mu_3} + \frac{c_2}{\mu_2} = 999.0388$ and is smaller than either the $c\mu$-rule or the LEPT rule. However, we still believe that the $c\mu$-rule is optimal if the number of jobs of each class is sufficiently large. Moreover, we expect that the performance of a policy which switches between the $c\mu$-rule and the LEPT rule according to an appropriate switching surface should be very close to the performance of the optimal policy. However, it is still not clear how we set the switching surface.

The bound we obtain for the worst case ratio is not tight. Numerical examples we ran all had ratios less than 1.22. The reason the bound is not tight arises from the fact that the lower bound obtained in Lemma 3.6 is loose. Although there are other bounds tighter than the one in Lemma 3.6, these bounds usually do not have closed-form expressions and the ratio cannot be optimized analytically.

Acknowledgement The authors would like to thank Dr. Steve Lavenberg for his insightful criticisms of this work and for his careful reading of the manuscript.

Appendix. Proof of Lemma 2.4

Let

$$g(x_1, x_2) \stackrel{\text{def}}{=} \gamma_m(x_1, x_2) - \gamma_0(x_1, x_2) = (\frac{1}{\mu_1} - \frac{1}{\mu_2})(\frac{c_1 x_1 + c_2 x_2}{m})$$

From (1), it follows that

$$J(x_1, x_2 - 1) = \frac{c_1 x_1 + c_2(x_2 - 1)}{q(l_2)} + \frac{l_2 \mu_1}{q(l_2)} J(x_1 - 1, x_2 - 1) + \frac{(m - l_2)\mu_2}{q(l_2)} J(x_1, x_2 - 2),$$

and

$$J(x_1 - 1, x_2) = \frac{c_1(x_1 - 1) + c_2 x_2}{q(l_1)} + \frac{l_1 \mu_1}{q(l_1)} J(x_1 - 2, x_2) + \frac{(m - l_1)\mu_2}{q(l_1)} J(x_1 - 1, x_2 - 1).$$

Thus,

$$\begin{aligned}
& f(x_1, x_2) \\
& = \frac{l_1 \mu_1}{q(l_1)} \big(J(x_1 - 2, x_2) - J(x_1 - 1, x_2 - 1) + g(x_1 - 1, x_2) \big) \\
& + \frac{l_1 \mu_1}{q(l_1)} J(x_1 - 1, x_2 - 1) - \frac{l_1 \mu_1}{q(l_1)} g(x_1 - 1, x_2) \\
& + \frac{(m - l_2)\mu_2}{q(l_2)} \big(J(x_1 - 1, x_2 - 1) - J(x_1, x_2 - 2) + g(x_1, x_2 - 1) \big) \\
& - \frac{(m - l_2)\mu_2}{q(l_2)} J(x_1 - 1, x_2 - 1) - \frac{(m - l_2)\mu_2}{q(l_2)} g(x_1, x_2 - 1) \\
& + \frac{c_1(x_1 - 1) + c_2 x_2}{q(l_1)} - \frac{c_1 x_1 + c_2(x_2 - 1)}{q(l_2)} + g(x_1, x_2) \\
& + \frac{(m - l_1)\mu_2}{q(l_1)} J(x_1 - 1, x_2 - 1) - \frac{l_2 \mu_1}{q(l_2)} J(x_1 - 1, x_2 - 1).
\end{aligned}$$

Observe that

$$\frac{c_1(x_1 - 1) + c_2 x_2}{q(l_1)} - \frac{l_1 \mu_1}{q(l_1)} g(x_1 - 1, x_2) = \frac{c_1(x_1 - 1) + c_2 x_2}{m \mu_2}$$

and

$$-\frac{c_1 x_1 + c_2(x_2 - 1)}{q(l_2)} - \frac{(m - l_2)\mu_2}{q(l_2)} g(x_1, x_2 - 1) = -\frac{c_1 x_1 + c_2(x_2 - 1)}{m \mu_1}.$$

Thus,

$$g(x_1, x_2) + \frac{c_1(x_1 - 1) + c_2 x_2}{m\mu_2} - \frac{c_1 x_1 + c_2(x_2 - 1)}{m\mu_1} = B$$

and (7) is derived.

The cases for boundary states can be proved similarly. ■

References

[1] Baras, J.S., Dorsey, A.J. and Makowski, A.M. (1985) "Two Competing Queues with Linear Costs and Geometric Service Requirements: the μc-Rule Is Often Optimal," *Adv. Appl. Prob.* 17, 186-209.

[2] Buyukkoc, C., Varaiya, P. and Walrand, J. (1985) "The $c\mu$ Rule Revisited," *Adv. Appl. Prob.* 17, 237-238.

[3] Chang, C.S., Chao, X.L., Pinedo, M. and Weber, R.R.(1989) "On the Optimality of LEPT and $c\mu$-Rule for Machines in Parallel," to appear in *J. Appl. Prob.*

[4] Frederickson, G.N., Bruno, J. and Downey, P. (1981) "Sequencing Tasks with Exponential Service Times to Minimize the Expected Flow Time or Makespan," *J. Ass. Comp. Mach.* 28, 100-113.

[5] Kampke, T (1987) "On the Optimality of Static Priority Policies in Stochastic Scheduling on Parallel Machines," *J. Appl. Prob.* 24, 430-448.

[6] Kampke, T. (1989) "Optimal Scheduling of Jobs with Exponential Service Times on Identical Parallel Processors," *Oper. Res.* 37, 126-133.

[7] Kawaguchi, T. and Kyan, S. (1986) "Worst Case of an LRF Schedule for the Mean Weighted Flow-Time Problem," *SIAM J. Comput.* 15, 1119-1129.

[8] Pinedo, M. (1983) "Stochastic Scheduling with Release Dates and Due Dates," *Oper. Res.* 31, 559-572.

[9] Ross, S. (1983) *Introduction to Stochastic Dynamic programming*, Academic Press.

[10] Weber, R.R. (1982) "Scheduling Jobs with Stochastic Processing Requirements on Parallel Machines to Minimize Makespan or Flowtime," *J. Appl. Prob.* 19, 167-182.

[11] Weber, R.R. (1983) "Scheduling Stochastic Jobs on Parallel Machines to Minimize Makespan or Flowtime," in Proceeding of the *ORSA/TIMS* Special Interest Meeting: Applied Probability- Computer Science, the Interface, ed. R. Disney, 327-338.

[12] Weiss, G. (1982) "Multiserver Stochastic Scheduling, Deterministic and Stochastic Scheduling," Dempster, Lenstra & Rinnooy Kan [1982], 157-179.

[13] Van Der Heyden, L. (1981) "Scheduling Jobs with Exponential Processing and Arrival Times on Identical Processors so as to Minimize the Expected Makespan," *Math. of Oper. Res.* 6, 305-312.