SIAM J. COMPUT.
Vol. 6, No. 1, March 1977

BOUNDS FOR LPT SCHEDULES ON UNIFORM PROCESSORS*

TEOFILO GONZALEZ†, OSCAR H. IBARRA‡ AND SARTAJ SAHNI‡

Abstract. We study the performance of LPT (largest processing time) schedules with respect to optimal schedules in a nonpreemptive multiprocessor environment. The processors are assumed to have different speeds and the tasks being scheduled are independent.

Key words. LPT schedules, uniform processors, nonpreemptive scheduling, independent tasks

1. Introduction. A uniform processor system [4] is one in which the processors $P_1, \cdots, P_m$ have relative speeds $s_1, \cdots, s_m$ respectively. It is assumed that the speeds have been normalized such that $s_1 = 1$ and $s_i \geqq 1$, $2 \leqq i \leqq m$. The problem of scheduling n independent tasks $(J_1, \cdots, J_n)$ with execution times $(t_1, \cdots, t_n)$ on m uniform processors to obtain a schedule with the optimal (least) finish time is known to be NP-complete [1], [4]. Hence, it appears unlikely that there is any polynomial time bounded algorithm to generate such schedules. For preemptive scheduling, however, optimal finish time algorithms can be obtained in polynomial time [6], [7]. Horowitz and Sahni [4] showed that for any m, polynomial time algorithms exist to obtain schedules with a finish time arbitrarily close to the optimal finish time. The complexity of these algorithms was, however, exponential in m. The purpose of this paper is to study the finish time properties of LPT schedules with respect to the optimal finish time.

DEFINITION. An *LPT (largest processing time) schedule* is a schedule obtained by assigning tasks to processors in order of nonincreasing processing times. When a task is being considered for assignment to a processor, it is assigned to that processor on which its finishing time will be earliest. Ties are broken by assigning the task to the processor with least index.

One may easily verify that for identical processor systems, this definition is equivalent to that of [2, p. 100]. Graham [3] studied LPT schedules for the special case of identical processors, i.e., $s_i = 1$, $1 \leqq i \leqq m$. If $\hat{f}$ is the finish time of the LPT schedule and f^* the optimal finish time, then Graham's result is that $\hat{f}/f^* \leqq \frac{4}{3} - \frac{1}{3m}$ and that this bound is the best possible bound. In § 2 we extend his work to the general case of uniform processors. While the bound we obtain is best possible for $m = 2$, it appears that it is not so for $m > 2$. In view of this, we turn our attention to another special case of uniform processors, i.e., $s_1 = 1$, $1 \leqq i < m$ and $s_m = s \geqq 1$. This case has previously been studied by J. W. S. Liu and C. L. Liu [5]. Using a priority assignment according to lengths of tasks, they show that $f/f^* \leqq 2(m-1+s)/(s+2)$ for $s \leqq 2$ and $f/f^* \leqq (m-1+s)/2$ for $s \geqq 2$, where f is the finish time of the priority schedule.

* Received by the editors January 24, 1975, and in final revised form March 28, 1976. This work was supported in part by the National Science Foundation under Grants DCR72-03728-A01 and DCR74-10081.

† Department of Computer Science, University of Minnesota, Minneapolis, Minnesota. Now at Department of Computer Science, Pennsylvania State University, University Park, Pennsylvania 16802.

‡ Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455.

Similar bounds for list schedules are also obtained by them. We show that for $m \geqq 3$, $\hat{f}/f^* \leqq 3/2 - 1/(2m)$ and that this bound is the best possible for $m = 3$. For $m > 3$ we conjecture that $\hat{f}/f^* \leqq 4/3$.

Before presenting our results we develop the necessary notation and basic results. Throughout the remainder of this paper $\hat{f}$ and f^* will denote the finish times of LPT and optimal schedules respectively. Let S be the set of tasks being scheduled. It will sometimes be necessary to distinguish between finish times of different sets of tasks. To do this, S will appear as a superscript along with $\hat{f}$ or f^* as in $\hat{f}^S$ and f^{*S}. If the number of processors is important, then this number will appear as a subscript as in $\hat{f}_m$, f_m^{*S} etc. We shall refer to the sets of tasks (jobs) by their task execution time. Thus, we speak of a set, S, of tasks $(t_1 \geqq t_2 \geqq \cdots \geqq t_n)$ meaning the execution time of task i is t_i and $t_i \geqq t_{i+1}$, $1 \leqq i < n$. The m processors $P_1, \cdots, P_m$ are assumed ordered such that $s_1 = 1$ and $1 \leqq s_i \leqq s_{i+1}$, $2 \leqq i < m$. The following result from [2, p. 102] is made use of:

LEMMA 1.1. *If for any m, $S = (t_1 \geqq t_2 \geqq \cdots \geqq t_n)$ is the smallest set of tasks for which $\hat{f}/f^* > k$, then t_n determines the finish time $\hat{f}$ (i.e., task n has the latest completion time).*

2. Basic results. In this section, we prove two important lemmas that are used throughout the paper (Lemmas 2.2 and 2.3). We also derive the bound $2m/(m+1)$ for the ratio $\hat{f}/f^*$ for the general m-processor system. Examples are shown for which $\hat{f}/f^*$ approaches $3/2$ as $m \to \infty$.

We begin with the following lemma. Informally, it states that if either the LPT or optimal schedule of an $(m+1)$-processor system has an idle processor, then the ratio $\hat{f}/f^*$ for this schedule is no worse than $\hat{f}/f^*$ for m processors.

LEMMA 2.1. *For $m \geqq 1$, let $g(m, s_2, \cdots, s_m)$ be such that $\hat{f}_m/f_m^* \leqq g(m, s_2, \cdots, s_m)$. Consider any $(m+1)$-processor system with job set $S = (t_1 \geqq t_2 \geqq \cdots \geqq t_n)$ and processor speeds $1 = s_1 \leqq s_2 \leqq \cdots \leqq s_{m+1}$. If a processor is idle in either the LPT or optimal schedule of S, then $\hat{f}_{m+1}^S/f_{m+1}^{*S} \leqq g(m, s_3/s_2, \cdots, s_{m+1}/s_2)$.*

Proof. Suppose in the LPT schedule of S a processor P_i is idle. Then it must be the case that in the optimal schedule, P_i is also idle. Otherwise, $\hat{f}_{m+1}^S \leqq t_n/s_i$, $f_{m+1}^{*S} \geqq t_n/s_i$ and $\hat{f}_{m+1}^S/f_{m+1}^{*S} = 1$. So we need only consider the case when P_i is idle in the optimal schedule. If P_i is idle then clearly P_1 is also idle or can be made idle without increasing f^* by scheduling the jobs from P_1 onto P_i. Consider the m-processor system with job set S and processor speeds $1 = s_2/s_2 \leqq s_3/s_2 \leqq \cdots \leqq s_{m+1}/s_2$. Then by assumption, for this system, $\hat{f}_m^S/f_m^{*S} \leqq g(m, s_3/s_2, \cdots, s_{m+1}/s_2)$. Moreover, $\hat{f}_{m+1}^S \leqq \hat{f}_m^S/s_2$ and $f_{m+1}^{*S} = f_m^{*S}/s_2$. It follows that $\hat{f}_{m+1}^S/\hat{f}_{m+1}^{*S} \leqq g(m, s_3/s_2, \cdots, s_{m+1}/s_2)$. □

The next lemma gives an estimate of $\hat{f}/f^*$ for the case when $\hat{f}$ is determined by the job with the smallest execution time.

LEMMA 2.2. *Consider an m-processor system with job set $S = (t_1 \geqq t_2 \geqq \cdots \geqq t_n)$ and speeds $s_1, \cdots, s_m$. If in the LPT schedule of S, the finish time $\hat{f}$ is determined by t_n, (i.e., if task n has the latest completion time), then $\hat{f}/f^* \leqq 1 + (m-1)t_n/(Qf^*)$, where $Q = \sum s_i$.*

Proof. Let the LPT schedule be as shown in Fig. 2.1, where P_k determines the finish time. Each T_i is the sum (possibly 0) of task times of jobs scheduled on P_i

prior to t_n's assignment, $T_1+\cdots+T_m=t_1+\cdots+t_{n-1}$.

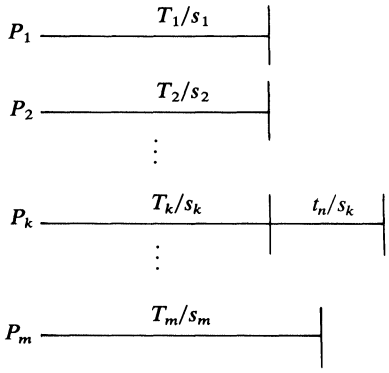

FIG. 2.1

Since task n determines the finish time, $\hat{f}=(T_k+t_n)/s_k$ and $(T_i+t_n)/s_i \geqq \hat{f}$ for $i \neq k$. Hence, $\hat{f}s_i - T_i \leqq t_n$ and so $\hat{f}\sum_{i\neq k} s_i - \sum_{i\neq k} T_i \leqq (m-1)t_n$. This, together with $\hat{f}s_k = T_k + t_n$ yields

$$\begin{aligned}\hat{f}Q &\leqq \sum T_i + mt_n \\ &= \sum t_i + (m-1)t_n.\end{aligned}$$

Since $f^* \geqq \sum t_i/Q$, we get $\hat{f}/f^* \leqq 1+(m-1)t_n/(f^*Q)$. □

Using Lemmas 2.1 and 2.2, we can now derive a bound for the m processor system.

THEOREM 2.1. *For an m-processor system,* $\hat{f}/f^* \leqq 2m/(m+1)$.

Proof. For $m=1$, the theorem obviously holds. Now suppose the theorem holds for $1, 2, \cdots, m-1$ processors but fails for m-processors. Let $S=(t_1 \geqq t_2 \geqq \cdots \geqq t_n)$ be the smallest set of jobs which gives a bound $\hat{f}_m/f_m^* > 2m/(m+1)$. Then by Lemma 1.1, t_n determines the finish time. There are two cases to consider. Both lead to a contradiction.

Case 1. $n \geqq m+1$. Then by Lemma 2.2,

$$\begin{aligned}\hat{f}_m/f_m^* &\leqq 1+\frac{(m-1)t_n}{Qf^*} \\ &\leqq 1+(m-1)t_n/[Q(\textstyle\sum t_i/Q)] \\ &\leqq 1+\frac{(m-1)t_n}{nt_n} = 1+\frac{(m-1)}{n} \leqq 1+\frac{m-1}{m+1} = \frac{2m}{m+1},\end{aligned}$$ a contradiction.

Case 2. $n \leqq m$. Then in the optimal schedule, either each processor has exactly one job or a processor is idle. In the first case, $\hat{f}_m/f_m^*=1$, since no processor can be idle in the LPT schedule (see proof of Lemma 2.1). For the second case $\hat{f}_m^S/f_m^{*S} \leqq \hat{f}_{m-1}^S/f_{m-1}^{*S} \leqq 2(m-1)/m \leqq 2m/(m+1)$ by Lemma 2.1. Either case leads to a contradiction. □

COROLLARY 2.1. *For an m-processor system,* $\hat{f}/f^* < 2$.

The bound of Theorem 2.1 is probably not a tight bound. However, we can show that there are examples approaching the bound 1.5 as $m \to \infty$.

THEOREM 2.2. *For every $m \geqq 2$, there is an example of an m-processor system and a set of jobs S for which $\hat{f}^S/f^{S*} = c$, where c is a positive root of the equation $2s^m - s^{m-1} - \cdots - s - 2 = 0$.*

Proof. The example we shall construct has job set $S = (t_1 \geqq t_2 \geqq \cdots \geqq t_m \geqq t_{m+1})$ (where m is the number of processors) and processor speeds $1 = s_1 \leqq \cdots \leqq s_m$. The t_i's and s_i's will satisfy the following properties (see Fig. 2.2):

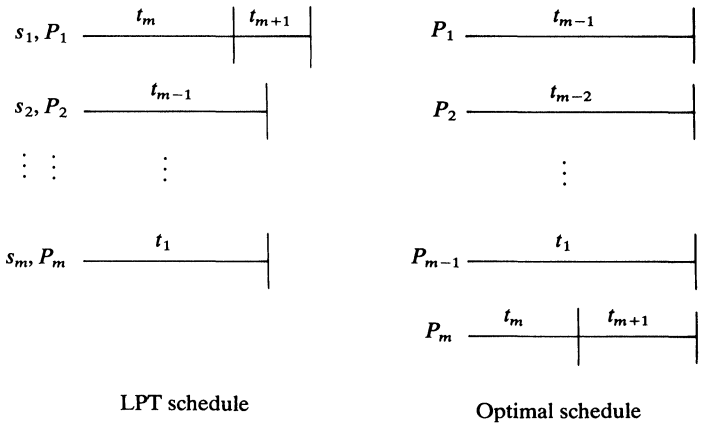

FIG. 2.2

$$\hat{f} = t_m + t_{m+1} \quad \text{and} \quad f^* = \frac{t_m + t_{m+1}}{s_m}, \tag{2.1}$$

$$t_m = t_{m+1} = t, \tag{2.2}$$

$$t_m + t_{m+1} = 2t = \frac{t_i + t}{s_{m-i+1}} \quad \text{for} \quad 1 \leqq i \leqq m-1, \tag{2.3}$$

$$\frac{t_m + t_{m+1}}{s_m} = \frac{2t}{s_m} = \frac{t_i}{s_{m-i}} \quad \text{for} \quad 1 \leqq i \leqq m-1. \tag{2.4}$$

Then $\hat{f}/f^* = 2t/(2t/s_m) = s_m$. From properties (2.1)–(2.4) we can derive the equation for s_m. From (2.3) we get

$$t_i = 2ts_{m-i+1} - t = t(2s_{m-i+1} - 1). \tag{2.5}$$

From (2.4), we have

$$s_m t_i = 2ts_{m-i}. \tag{2.6}$$

Equations (2.5) and (2.6) yield

$$s_{m-i+1} = \frac{2s_{m-i} + s_m}{2s_m} \quad \text{for} \quad 1 \leqq i \leqq m-1. \tag{2.7}$$

Using (2.7) repeatedly for $i = 1, 2, \cdots, m-1$ we get

$$s_m = \frac{2s_{m-1} + s_m}{2s_m} = \frac{2\left(\frac{2s_{m-2} + s_m}{2s_m}\right) + s_m}{2s_m}$$

$$= \frac{2s_{m-2} + s_m + s_m^2}{2s_m^2} = \frac{2\left(\frac{2s_{m-3} + s_m}{2s_m}\right) + s_m + s_m^2}{2s_m^2}$$

$$= \frac{2s_{m-3} + s_m + s_m^2 + s_m^3}{2s_m^3}$$

$$\vdots$$

$$= \frac{2s_1 + s_m + s_m^2 + \cdots + s_m^{m-1}}{2s_m^{m-1}}$$

Hence,

$$s_m = \frac{2 + s_m + s_m^2 + \cdots + s_m^{m-1}}{2s_m^{m-1}} \quad (\text{since } s_1 = 1)$$

or

$$2s_m^m - s_m^{m-1} - s_m^{m-2} - \cdots - s_m - 2 = 0. \tag{2.8}$$

The polynomial on the left-hand side of (2.8) has one sign change and so from Descartes' rule of sign it also has one positive real root. This root must clearly be >1 as otherwise the left-hand side is <0.

Let c be this positive real root of equation (2.8). We can construct an example of an m-processor system with $\hat{f}/f^* = c$ by setting $s_m = c$ and computing $s_2, \cdots, s_{m-1}$ in terms of c using (2.7). (Of course, $s_1 = 1$.) Then by letting $t_m = t_{m+1} = t$, we can determine the values of $t_1, \cdots, t_{m-1}$ in terms of t using (2.4). □

COROLLARY 2.2. *There exist uniform processor systems and job sets S for which $\hat{f}/f^* \approx 1.5$.*

Proof. From Theorem 2.2 we know that there are job sets, S, for which $\hat{f}/f^* = c$ where c is a positive root of (2.8). Let s be a root. Rearranging terms, we get

$$2s^m - 1 = \sum_{0 \leq i < m} s^i$$

$$= \frac{s^m - 1}{s - 1}$$

or $2s^{m+1} - 3s^m - s + 2 = 0$.

Since $s > 1$, for $m \to \infty$ we have $s \to 3/2$ as a root. □

Example. (a) $m=2$: Then we have $2s_2^2-s_2-s=0$, where we find $s_2=(1+\sqrt{17})/4$. Of course, $s_1=1$. Let $t_2=t_3=1$. From (2.4), we find

$$t_1=\frac{2t}{s_2}\cdot s_1=\frac{8}{1+\sqrt{17}}.$$

One easily verifies that $\hat{f}/f^*=(1+\sqrt{17})/4$.

(b) $m=3$: The equation to use is $2s_3^3-s_3^2-s_3-2=0$. $s_3=1.384$ is an approximate root of this equation. Using (2.7), we find $s_2=s_3(2s_3-1)/2=1.223$; $s_1=1$. Let $t_3=t_4=t=1$. Using (2.4), find $t_2=(2t/s_3)\cdot s_2=1.767$ and $t_1=(2t/s_3)\cdot s_1=1.445$. Again we can check that $\hat{f}/f^*$ is approximately 1.384.

(c) Some other roots of (2.8) are 1.493 for $m=10$ and 1.499 for $m=20$.

3. The case $s_i=1$, $1\leqq i\leqq m$, and $s_m\geqq 1$. In this section we study the special case in which all but one of the $m\geqq 1$ processors has a speed of 1. The mth processor P_m has a speed $s\geqq 1$. The main result of this section is stated below as Theorem 3.1.

THEOREM 3.1. *For $m\geqq 2$ the ratio $\hat{f}/f^*$ has the following bounds:*

(i) $\hat{f}/f^*\leqq(1+\sqrt{17})/4$ *for* $m=2$,

(ii) $\hat{f}/f^*\leqq 3/2-1/(2m)$ *for* $m>2$.

Proof. (i) is proved in Lemma 3.2. (ii) follows from Lemmas 3.1–3.6 and the fact that the bound is a monotone increasing function in m.

Before proving the theorem we derive a general bound for $\hat{f}/f^*$ in terms of m and s.

LEMMA 3.1. *For an m-processor system with $s_i=1$ for $1\leqq i<m$ and $s_m=s$, $\hat{f}/f^*\leqq 2(m-1+s)/(m-1+2s)$.*

Proof. If $m=1$, the lemma is obviously true since $\hat{f}/f^*=1$. Now assume that the lemma holds for $1,2,\cdots,m-1$ processors but fails for m ($m\geqq 2$). For this m, let $S=(t_1\geqq t_2\geqq\cdots\geqq t_n)$ be the smallest set of jobs for which $\hat{f}/f^*>2(m-1+s)/(m-1+2s)$. Suppose a processor is idle in either the LPT or optimal schedule of S. Then $\hat{f}_m^S/f_m^{*S}\leqq\hat{f}_{m-1}^S/f_{m-1}^{*S}\leqq 2(m-2+s)/(m-2+2s)\leqq 2(m-1+s)/(m-1+2s)$ by Lemma 2.1.

So we may assume that no processor is idle in either the LPT or optimal schedule of S. We consider two cases, both leading to a contradiction.

Case 1. The LPT schedule is as shown in Fig. 3.1, where each T_i represents the sum of execution times of jobs scheduled on P_i prior to the assignment of t_n, $T_1+\cdots+T_m=t_1+\cdots+t_{n-1}$. By assumption, no processor is idle. Hence $T_i>0$ for $2\leqq i\leqq m$. Since the first $m-1$ processors have speed 1, we may assume that $T_i\geqq T_1$ for $1\leqq i\leqq m-1$. Now if $T_1=0$, then $\hat{f}=t_n$. But $f^*\geqq t_n$ since by assumption no processor is idle in the optimal schedule. Then $\hat{f}/f^*=1$. So we may also assume that $T_1\geqq t_n$.

Thus, $\hat{f}\geqq 2t_n$. From Lemma 2.2 we have $\hat{f}/f^*\leqq 1+(m-1)t_n/(Qf^*)$, where $Q=(m-1)+s$. This implies that $Qf^*\geqq Q\hat{f}-(m-1)t_n\geqq 2Qt_n-(m-1)t_n$. Substituting this inequality back into Lemma 2.2 gives

$$\frac{\hat{f}}{f^*}\leqq 1+\frac{(m-1)t_n}{2Qt_n-(m-1)t_n}=1+\frac{m-1}{2Q-m+1}=\frac{2Q}{2Q-m+1}=\frac{2(m-1+s)}{m-1+2s},$$

a contradiction.

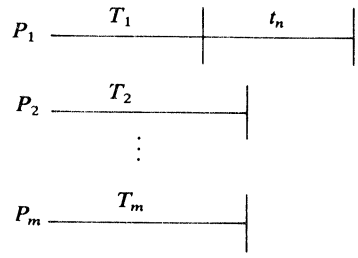

FIG. 3.1

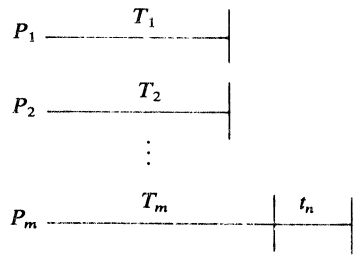

FIG. 3.2

Case 2. Suppose the LPT schedule is as shown in Fig. 3.2, where we again assume that $T_i \geqq T_1 \geqslant t_n$. We may also assume that $T_m > 0$; otherwise $\hat{f}/f^* = 1$ since $\hat{f} = t_n/s$. If $\hat{f} \geqq 2t_n$ then the proof proceeds as in Case 1. Otherwise, $\hat{f} < 2t_n$. Note that $\sum t_i \geqq s\hat{f} + (m-1)t_n$. Therefore,

$$\frac{\hat{f}}{f^*} \leqq \frac{Q\hat{f}}{s\hat{f} + (m-1)t_n} = \frac{Q}{s + (m-1)t_n/\hat{f}}.$$

Since $\hat{f} < 2t_n$ we have

$$\frac{\hat{f}}{f^*} \leqq \frac{Q}{s + (m-1)/2} = \frac{2Q}{m-1+2s} = \frac{2(m-1+s)}{m-1+2s}. \quad \square$$

The bound for $m = 2$ follows from the following lemma.

LEMMA 3.2. *For an m processor system with* $s_i = 1$, $1 \leqq i < m$ *and* $s_m = s$, $\hat{f}/f^* \leqq \frac{1}{4}[(3-m) + \sqrt{(3-m)^2 + 16(m-1)}]$. *Moreover, for* $m = 2$, *the bound is tight.*

Proof. Let $k > 1$ be the desired bound for $\hat{f}/f^*$. Let $Q = \sum s_l = m - 1 + s$. First we show that if $s \leqq 2Q(k-1)/(m-1)$, then $\hat{f}/f^* \leqq k$. Suppose not. Let $S = (t_1 \geqq \cdots \geqq t_n)$ be the smallest set of jobs for which $\hat{f}/f^* > k$. Then t_n determines the finish time and by Lemma 2.2, $\hat{f}/f^* \leqq 1 + (m-1)t_n/(Qf^*)$. Hence $f^* < (m-1)t_n/[Q(k-1)]$. It follows that the number of jobs on each processor in the optimal schedule of S is less than $(m-1)s/[Q(k-1)] \leqq 2$. But then in this case, $\hat{f}/f^* = 1$. This contradicts the assumption that S produces a bound $> k$. Thus if $s \leqq 2Q(k-1)/(m-1)$, then $\hat{f}/f^* \leqq k$. This, in turn, implies that if $Q \leqq (m-1) + 2Q(k-1)/(m-1)$, then $\hat{f}/f^* \leqq k$ or that

$$\text{if} \quad Q \leqq \frac{(m-1)^2}{m-2k+1}, \quad \text{then} \quad \hat{f}/f^* \leqq k. \tag{3.1}$$

Now by Lemma 3.1, we have

$$\frac{\hat{f}}{f^*} \leqq \frac{2(m-1+s)}{m-1+2s} = \frac{2(m-1+s)}{2(m-1+s)-(m-1)} = \frac{2Q}{2Q-(m-1)}.$$

It follows that if $2Q/(2Q-(m-1)) \leqq k$, then $\hat{f}/f^* \leqq k$ or

$$\text{if} \quad Q \geqq \frac{(m-1)k}{2(k-1)} \quad \text{then} \quad \hat{f}/f^* \leqq k. \tag{3.2}$$

To satisfy (3.1) and (3.2) simultaneously, we must have $(m-1)^2/(m-2k+1) \geqq (m-1)k/(2(k-1))$, from which we get $k \geqq \frac{1}{4}[(3-m)+\sqrt{(3-m)^2+16(m-1)}]$. For all such k, $\hat{f}/f^* \leqq k$ for all Q.

For the case $m=2$, we have $k=(1+\sqrt{17})/4$. In § 2 we saw an example with $s_2=(1+\sqrt{17})/4$ for which $\hat{f}/f^*=(1+\sqrt{17})/4$. Hence, this bound is tight for $m=2$. □

In arriving at the proof of the theorem for $m>2$, it is necessary to prove four lemmas. To begin with, we show that if for any set of jobs, S, an optimal schedule has more than one job on any of the processors $P_1, \cdots, P_{m-1}$ then $\hat{f}^S/f^{*S} \leqq 3/2-1/(2m)$.

LEMMA 3.3. *For any set of jobs, S, either* (i) *processors P_1-P_{m-1} have at most one job scheduled on each in every optimal schedule or*

$$\text{(ii)} \qquad \frac{\hat{f}_m^S}{f_m^{*S}} \leqq \frac{3}{2}-\frac{1}{2m}.$$

Proof. Suppose (ii) is not true for some set of jobs. Let $S=(t_1 \geqq t_2 \geqq \cdots \geqq t_n)$ be the smallest set of jobs for which $\hat{f}_m^S/f_m^{*S} > 3/2-1/(2m)$. From Lemma 2.2 we get

$$\frac{\hat{f}_m^S}{f_m^{*S}} \leqq 1+\frac{(m-1)t_n}{(m-1+s)f_m^*} > \frac{3}{2}-\frac{1}{2m}$$

or

$$\frac{(m-1)t_n}{(m-1+s)f_m^*} > \frac{m-1}{2m}$$

or

$$t_n > \frac{m-1+s}{2m} f_m^*$$

$$\geqq \frac{1}{2} f_m^*$$

i.e., $f_m^* < 2t_n$ which, in turn, means that none of the processors P_1-P_{m-1} can have more than one job scheduled on them in an optimal schedule. □

Next, we prove that if $s \geqq m-1$ then $\hat{f}/f^* \leqq 4/3$.

LEMMA 3.4. *If* $s \geqq m-1$ *then* $\hat{f}/f^* \leqq 4/3 \leqq 3/2 - 1/(2m)$ *for* $m > 2$.

Proof. Lemma 3.1 gives

$$\hat{f}/f^* \leqq \frac{2(m-1+s)}{m-1+2s}.$$

The right-hand side of the above inequality is a decreasing function of s. Hence, for $s \geqq m-1$ we obtain

$$\begin{aligned}\frac{\hat{f}_m}{f_m^*} &\leqq \frac{4m-4}{3(m-1)} \\ &= 4/3 \\ &\leqq 3/2 - 1/(2m), \quad m > 2. \quad \square\end{aligned}$$

As a result of Lemmas 3.3 and 3.4 the only counterexamples to Theorem 3.1 are sets of jobs, S, for which the optimal schedules have at most one job on each of $P_1 - P_{m-1}$ and the speed, s, of P_m is $< m-1$. The next two lemmas show that for this kind of an optimal schedule and $s < m-1$ the bound of Theorem 3.1 cannot be violated.

LEMMA 3.5. *Let* $S = (t_1 \geqq t_2 \geqq \cdots \geqq t_n)$ *be the smallest set of jobs for which* $\hat{f}/f^* > 3/2 - 1/(2m)$. *If in the LPT schedule,* t_i *is the only job scheduled on one of the processors,* $P_1, \cdots, P_{m-1}$ *and if in an optimal schedule* t_j *is the only job scheduled on one of the processors,* $P_1, \cdots, P_{m-1}$ *then, either*

(i) $$\hat{f}_m^S / f_m^{*S} \leqq \hat{f}_{m-1}/f_{m-1}^*$$

or

(ii) $$t_i < t_j.$$

Proof. From Lemma 1.1 it follows that t_n determines the finish time $\hat{f}^S$. If any one of the processors $P_1, \cdots, P_m$ is idle in an optimal solution (i.e. no jobs have been scheduled on it), then $f_m^{*S} = f_{m-1}^{*S}$. But, $\hat{f}_m^S \leqq f_{m-1}^S$ and so $\hat{f}_m^S / f_m^{*S} \leqq \hat{f}_{m-1}^S / f_{m-1}^{*S}$. We may therefore assume that no processor is idle in any optimal solution. Hence, $f_m^{*S} \geqq t_n$. If $i = n$, then $\hat{f}_m^S = t_n$ (as t_i is the only job on some processor $P_1, \cdots, P_{m-1}$) and $\hat{f}_m^S / f_m^{*S} \leqq 1$. Therefore $i \neq n$. Now, we have

$$\begin{aligned}f_m^{*S} &= \max\{t_j, f_{m-1}^{*S-\{t_j\}}\} \\ &\geqq f_{m-1}^{*S-\{t_j\}} \\ &\geqq f_{m-1}^{*S-\{t_i\}} \cdots \quad \text{as} \quad t_i \geqq t_j,\end{aligned}$$

but

$$\hat{f}_m^S = \hat{f}_{m-1}^{S-\{t_i\}} \quad \cdots \quad \text{as} \quad i \neq n.$$

Therefore,

$$\frac{\hat{f}_m^S}{f_m^{*S}} \leqq \frac{\hat{f}_{m-1}^{S-\{t_i\}}}{f_{m-1}^{*S-\{t_i\}}} \leqq \frac{\hat{f}_{m-1}}{f_{m-1}^*}.$$

LEMMA 3.6. *When $s < m-1$ and an optimal schedule for any set of jobs S has at most one job on each of processors $P_1 - P_{m-1}$, then $\hat{f}_m/f^*_m \leqq 3/2 - 1/(2m)$.*

Proof. Let $S = (t_1 \geqq t_2 \geqq \cdots \geqq t_n)$ be the smallest set of jobs and m the least $m > 2$ for which the lemma is not true. From Lemma 3.1 we obtain $\hat{f}/f^* \leqq 1 + (m-1)/(m-1+s)(t_n/f^*)$. By assumption $\hat{f}/f^* > 3/2 - 1/(2m)$. Therefore,

$$1 + \frac{(m-1)}{m-1+s}\frac{t_n}{f^*} > \frac{3}{2} - \frac{1}{2m}$$

or

$$f^* < \frac{2m}{m-1+s} t_n. \tag{3.3}$$

If $\#_m$ is the number of jobs on P_m in an optimal schedule then, $f^* \geqq \#_m t_n/s$. Substituting this inequality into (3.3) yields

$$\#_m < \frac{2sm}{m-1+s}. \tag{3.4}$$

The right-hand side of the inequality (3.4) is an increasing function of s. Since $s < m-1$, (3.4) yields the following bound on $\#_m$:

$$\#_m \leqq \frac{2(m-1)m}{2(m-1)} = m.$$

The optimal schedule has at most one job on each of $P_1 - P_{m-1}$. Hence, $n \leqq 2m-2$.

The remainder of the proof shows that if $n \leqq 2m-2$ then Lemma 3.5 can be used to show that $\hat{f}^S_m/f^{*S}_m \leqq \hat{f}^S_{m-1}/f^{*S}_{m-1}$, thus contradicting the assumption that this was the least m for which the lemma was false. (The contradiction comes about as $3/2 - 1/(2m)$ is monotone increasing in m and the fact that when $m = 3$ this bound is $4/3$ which is greater than the known bound for $m = 2$.) Clearly, we may assume that each processor has at least one job scheduled on it in every optimal schedule.

Let k be the smallest index (i.e. largest job) on any of the processors $P_1 - P_{m-1}$ in an optimal schedule. Then, the schedule obtained by assigning job t_{k+i-1} to processor P_i, $1 \leqq i < m$, and the remaining jobs to processor P_m has a finish time no greater than the optimal finish time f^{*S}_m. Such a schedule shall be denoted by OPT_k. Clearly, $1 \leqq k \leqq n-m+2$. Since, $n \leqq 2m-2$ at least one of the processors $P_1 - P_{m-1}$ has exactly one job scheduled on it (every processor must have at least one job on it as otherwise, by the definition of LPT, $\hat{f} \leqq t_n$ but $f^* \geqq t_n$). Let the index of this job be i. Then, t_i must be the largest job amongst jobs scheduled on $P_1 - P_{m-1}$ in the LPT schedule (this again follows from the definition of LPT). But, $s < m-1$ implies $t_i \geqq t_{m-1}$ as LPT cannot schedule all of the first $m-1$ jobs on P_m when $s < m-1$. For all $k \geqq 1$, OPT_k has a job with index $j = k+m-2 \geqq m-1$ on P_{m-1} and this is the only job on P_{m-1}. By the ordering on the jobs, $t_j \leqq t_{m-1}$. So, $t_i \geqq t_j$. Lemma 3.5 now implies that $\hat{f}^S_m/f^{*S}_m \leqq \hat{f}_{m-1}/f^*_{m-1}$; a contradiction. □

Having shown that $\hat{f}/f^*$ is indeed bounded as in Theorem 3.1, the next question is: How good is the bound. From the previous section we know that the

bound for $m = 2$ is tight. Lemma 3.7 shows that the bound is also tight for $m = 3$ and that for all $m > 3$ it is possible to have an $\hat{f}/f^*$ arbitrarily close to $4/3$. Lemma 3.8 shows that for $m = 4$ and 5 there is no set of jobs S for which $\hat{f}/f^* > 4/3$. This shows that the bound of $3/2 - 1/(2m)$ is not a tight bound for all values of m and leads us to conjecture that for $m \geqq 3$ the bound is in fact $4/3$. Note the closeness of this bound of $4/3$ to the bound $4/3 - 1/(3m)$ obtained by Graham [3] for the case of $s = 1$ (i.e., m identical processors).

LEMMA 3.7. *For $m \geqq 3$ and any $\varepsilon > 0$, there is a set of jobs, S, and a speed $s > 1$ for which $\hat{f}/f^* > 4/3 - \varepsilon$.*

Proof. For any $m \geqq 3$ consider the set of jobs $t_1 = 1.5$, $t_2 = 1.5$, $t_j = 1$, $3 \leqq j \leqq m+2$ and $s = 2 + \varepsilon'$ with ε' very close to zero. The LPT schedule has jobs t_1, t_2 and t_{m+2} on P_m with $\hat{f} = 4/(2+\varepsilon')$. One optimal schedule is shown in Fig. 3.3. $f^* = 1.5$. Hence, $\hat{f}/f^* = 8/(6+3\varepsilon') \to 4/3$ as $\varepsilon' \to 0$.

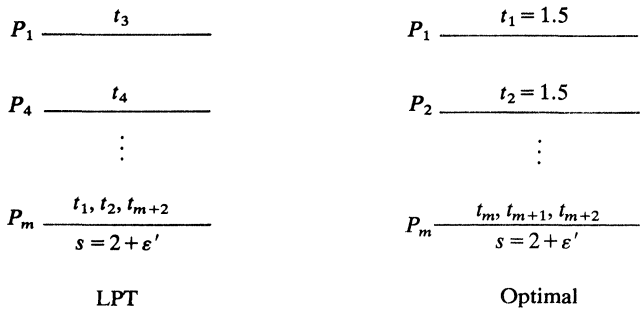

FIG. 3.3. *LPT and optimal schedules for Lemma* 3.7

LEMMA 3.8. *For $m = 4$ and 5, $\hat{f}/f^* \leqq 4/3$.*

Proof. The proof is omitted and may be found in [8].

Conjecture. $\hat{f}/f^* \leqq 4/3$ for $m \geqq 3$ and $s_i = 1$, $1 \leqq i < m$ and $s_m \geqq 1$.

4. Conclusions. We have shown that in the case of uniform processors LPT schedules have a finish time at most twice the optimal finish time. The worst examples we could construct result in LPT schedules with finish times 1.5 times the optimal for $m \to \infty$. For the special case studied in [5] it is shown that $\hat{f}/f^* \leqq 3/2 - 1/(2m)$.

Acknowledgment. We are grateful to the referee for providing a simpler proof of Lemma 3.1.

REFERENCES

[1] J. BRUNO, E. G. COFFMAN, JR. AND R. SETHI, *Scheduling independent tasks to reduce mean finishing-time*, Comm. ACM, 17 (1974), pp. 382–387.

[2] E. G. COFFMAN, JR. AND P. J. DENNING, *Operating Systems Theory*, Prentice-Hall, Englewood Cliffs, N.J., 1973.

[3] R. L. GRAHAM, *Bounds on multiprocessing timing anomalies*, SIAM J. Appl. Math., 17 (1969), pp. 416–429.

[4] E. HOROWITZ AND S. SAHNI, *Exact and approximate algorithms for scheduling non-identical processors*, J. Assoc. Comput. Mach., 23 (1976), pp. 317–327.
[5] J. W. S. LIU AND C. L. LIU, *Bounds on scheduling algorithms for heterogeneous computing systems*, Proc. IFIP, (1974), pp. 349–353.
[6] J. W. S. LIU AND A. YANG, *Optimal scheduling of independent tasks on heterogeneous computing systems*, ACM National Conference, 1974, pp. 38–45.
[7] E. HORVATH AND R. SETHI, *Preemptive schedules for independent tasks*, Computer Science Tech. Rep. 162, Pennsylvania State Univ., College Park, 1975.
[8] T. GONZALEZ, O. H. IBARRA AND S. SAHNI, *Bounds for LPT schedules on uniform processors*, Tech. Rep. 75-1, Univ. of Minnesota, Minneapolis, 1975.