Some New Results on Liu’s Conjecture

Weizhen Mao
Department of Computer Science
The College of William and Mary
Williamsburg, Virginia 23187-8795
wm@cs.wm.edu

Abstract

Consider the problem of scheduling $n$ tasks with precedence constraint $≺$ on $m$ parallel processors so as to minimize the makespan. Let $\omega$ and $\omega'$ be the makespans of the optimal nonpreemptive and optimal preemptive schedules, respectively. Liu’s conjecture states that $\frac{\omega}{\omega'} \leq \frac{2m}{m+1}$. In this paper, we present a simple proof of the conjecture for $≺=\emptyset$, and give an improved bound closer to that in the conjecture than the previously proved bound for arbitrary $≺$.

Keywords: Scheduling, preemption, nonpreemption, analysis of algorithms

1 Introduction

In a multi-processor system, scheduling tasks on the processors can be done with two different strategies. A scheduling strategy is said to be non-preemptive if once the execution of a task begins on a processor, it must continue until its total completion. A scheduling strategy is said to be preemptive if the execution of a task can be interrupted at any time and be resumed later on. Because non-preemption is a special case of preemption, preemption generates schedules at least as good as non-preemption. However, determining when to preempt during the execution of a task may require extra time complexity in the algorithm. We therefore ask the following questions: How good is preemption versus non-preemption? Is it worthwhile to spend a lot of time to have a preemptive schedule?

Let us define the scheduling problem more formally. Given a task system $T$ of $n$ tasks $T_1,T_2,\ldots,T_n$, which are to be scheduled on $m$ parallel and identical processors $P_1,P_2,\ldots,P_m$. The processing time of task $T_j$ is $p_j$ for $j=1,2,\ldots,n$. Moreover, a precedence constraint relation $≺$ exists among tasks in $T$, i.e., by $T_i≺T_j$, we mean that the execution of $T_j$ will not begin until the completion of the execution of $T_i$. The precedence constraint among tasks can be depicted by a directed acyclic graph (DAG), where each node corresponds to a task and there is an arc from $T_i$ to $T_j$ iff $T_i≺T_j$. The goal of the problem is to minimize the makespan, or the maximum completion time $C_{max}$ of the schedule. As pointed out earlier, there are two ways to schedule the tasks: nonpreemption and preemption. We therefore have two variations of the problem. Using the $|\alpha| |\beta| |\gamma|$ notation introduced by Lawler, Lenstra, Rinnooy Kan and Shmoys [?], they can be denoted as $P|\alpha| |\beta| |\gamma |C_{max}$ and $P|\alpha| |\beta| |\gamma ,pmtn|C_{max}$, respectively. FIG. 1 shows the optimal preemptive and optimal nonpreemptive schedules of a task system on two processors.

Given any instance $I$ specified by $m$, $n$, $p_j$’s and $≺$, let $\omega$ denote the makespan of its optimal nonpreemptive schedule, and $\omega'$ denote the makespan of its optimal preemptive schedule. To be more specific, let $NPS(I)$ and $PS(I)$ be the sets of all non-preemptive and preemptive schedules for instance $I$, respectively. Then

$$\omega = \min_{S \in NPS(I)} \{C_{max}(S)\}$$

and

$$\omega' = \min_{S \in PS(I)} \{C_{max}(S)\}.$$  

To compare preemption with non-preemption, we are interested in determining the maximum value of $\frac{\omega}{\omega'}$ over all instances. In the early seventies, Liu [?] studied this problem and made the following conjecture.
Let us first review some results, which were derived prior to Liu’s conjecture. The following lemma is due to Graham [7].

**Lemma 1** Let $\phi$ be the total idle time in the optimal nonpreemptive schedule. Let $C$ be the path, also called the chain, in the precedence constraint graph with the maximum total processing time, and $|C|$ be the total processing time of tasks on the chain. Then we have

$$\phi \leq (m - 1)|C|.$$  

**Proof** We define the list scheduling algorithm as follows. Given a priority list of tasks $L = (T_{r_1}, T_{r_2}, \ldots, T_{r_n})$. Whenever a processor becomes idle, assign the first unexecuted executable task in $L$ to the processor. A task is executable if all its predecessors in $\prec$ have been completed. If there is no executable task in the list, the processor remains idle until there is one. We then call the schedule obtained the list schedule with priority list $L$. Since the optimal nonpreemptive schedule is in fact a list schedule for some priority list, it is also called the optimal list schedule.

Consider the optimal list schedule. Let $\phi_i = (s_i, t_i)$ for $i = 1, 2, \ldots, l$ be the time periods such that there is at least one processor idle at any time in each period and that no processor is idle at any time in periods $(0, s_1), (t_i, s_{i+1})$ for $i = 1, 2, \ldots, l - 1$ and $(t_i, C_{\text{max}})$. Furthermore, to ensure that each $\phi_i$ is the longest possible time period, we assume that $t_i \neq s_{i+1}$ for $i = 1, 2, \ldots, l - 1$.

For $\phi_i = (s_i, t_i)$, there must be a task $T_{i_0}$ with starting time $t_i$. $T_{i_0}$ must have a predecessor $T_i$, finishing at time $t_i$, otherwise $T_{i_0}$ would have been scheduled earlier. For $T_{i_1}$, its starting time in period $\phi_i$, then $T_{i_1}$ must have a predecessor $T_{i_2}$ finishing at the starting time of $T_{i_1}$. This construction continues until we have a subchain $T_{i_1} \prec \cdots \prec T_{i_l}$, such that the sum of their processing times is greater than or equal to the length of $\phi_i$. We also see that the last task in the subchain for $\phi_i$ is a predecessor of the first task in the subchain for $\phi_{i+1}$. Therefore, we have constructed a
chain $\overline{C}$, which is the combination of all $l$ subchains, such that $|\overline{C}| \geq \sum \phi_i$. So, $\phi \leq (m-1) \sum \phi_i \leq (m-1)|\overline{C}| \leq (m-1)|C|$.

Using Lemma 1, Muntz [?] proved the following bound for $\bar{\omega}$.

**Lemma 2**

$$\frac{\omega}{\bar{\omega}} \leq \frac{2m-1}{m}.$$  

**Proof**  It is obvious that $\omega = \frac{1}{m}(\sum p_j + \phi)$. Thus by Lemma 1, we have $\omega = \frac{1}{m}(\sum p_j + \phi) \leq \frac{1}{m} \sum p_j + \frac{m-1}{m}|C|$. On the other hand, the makespan in the optimal preemptive schedule $\omega'$ must be no less than $\frac{1}{m}\sum p_j$ since the best possibility is to let all processors busy all the time. $\omega'$ must also be no less than the length of the chain $|C|$ since all the tasks in the chain have to be executed sequentially. Thus, we have $\omega' \geq \max\{\frac{1}{m} \sum p_j, |C|\}$. So,

$$\frac{\omega}{\omega'} \leq \frac{\frac{1}{m} \sum p_j + \frac{m-1}{m}|C|}{\max\{\frac{1}{m} \sum p_j, |C|\}} = \frac{\frac{1}{m} \sum p_j + \frac{m-1}{m}|C|}{\max\{\frac{1}{m} \sum p_j, |C|\}} \leq 1 + \frac{m-1}{m} \leq \frac{2m-1}{m}.$$  

The bound in Lemma 2 may not be the tightest because no instance that achieves bound $\frac{2m-1}{m}$ has ever been found. On the contrary, the largest achievable bound found so far is $\frac{2m}{m+1}$. So the following question arises: Is there a possibility that the tight bound for the ratio is exactly $\frac{2m}{m+1}$? We will prove in §3 that this is indeed the case when $\omega = \phi = 0$.

### 3 Independent Tasks

**Theorem 1**  When the precedence constraint $\prec$ is empty,

$$\frac{\omega}{\omega'} \leq \frac{2m}{m+1}$$  

and there exists an instance for which “=” holds.

**Proof**  Let us consider the optimal nonpreemptive schedule. Let LPT stand for the longest-processing-time-first list scheduling, which is in fact list scheduling with the tasks in the priority list ordered by decreasing $p_j$’s. Since there is no precedence constraint among the tasks, there is no idle time period during the entire execution of all tasks. See FIG. 2.

Let $T_k$ be the task that finishes last. Let $s_k$ be the starting time of $T_k$. Let $LPT(C_{\text{max}})$ be the makespan of the LPT schedule for the instance. We then have $\omega \leq LPT(C_{\text{max}}) = s_k + p_k \leq \frac{1}{m}(\sum p_j - p_k) + p_k = \frac{1}{m} \sum p_j + \frac{m-1}{m}p_k \leq \omega' + \frac{m-1}{m+1}p_k$. Let us consider the following two cases.

Case 1. $s_k > 0$

Every task $T_j$ starting at time 0 has $p_j \geq p_k$ since it is LPT. And there are at least $m$ such tasks. Let $X$ be the sum of the processing times of these tasks. So,

$$\frac{m-1}{m}p_k = \frac{m-1}{m+1}(p_k + \frac{1}{m}p_k) \leq \frac{m-1}{m+1}(X + \frac{1}{m}p_k) \leq \frac{m-1}{m+1}(\frac{1}{m} \sum p_j - p_k + \frac{1}{m}p_k) = \frac{m-1}{m+1} \sum p_j \leq \frac{m-1}{m+1} \omega'.$$

We then have

$$\frac{\omega}{\omega'} \leq \frac{\omega' + \frac{m-1}{m+1}p_k}{\omega'} \leq \frac{\omega' + \frac{m-1}{m+1} \omega'}{\omega'} = \frac{2m}{m+1}.$$  

Case 2. $s_k = 0$

In this case, LPT is in fact the optimal nonpreemptive schedule. Thus $\omega = p_k$ and $\omega' = p_k$. So,

$$\frac{\omega}{\omega'} = 1 \leq \frac{2m}{m+1}.$$  

The bound $\frac{2m}{m+1}$ is the best possible because it can be achieved by the instance in FIG. 3.
Claim 1 If $\sum p_j \geq \frac{m+1}{2} \omega$, then
\[
\frac{\omega}{\omega'} \leq \frac{2m}{m+1}.
\]

Proof Since the best we can do in any schedule (nonpreemptive as well as preemptive) is never to leave any processor idle, we have
\[
\frac{\omega}{\omega'} \leq \frac{\omega}{\frac{1}{m}(\sum p_j + \phi')}
\leq \frac{1}{m} \sum p_j
\leq \frac{\omega}{\frac{m+1}{2m} \omega}
= \frac{2m}{m+1}.
\]

Claim 2 If $\sum p_j \leq \frac{m^2+1}{m+1} \omega'$, then
\[
\frac{\omega}{\omega'} \leq \frac{2m}{m+1}.
\]

Proof Because $\sum p_j \leq \frac{m^2+1}{m+1} \omega'$, then $\phi' = m\omega' - \sum p_j \geq m\omega' - \frac{m^2+1}{m+1} \omega' = \frac{m-1}{m+1} \omega'$. Lemma 1 tells us that $\phi \leq (m-1)|C|$. So,
\[
\begin{align*}
\omega - m\omega' &= \phi - \phi' \\
&\leq (m-1)|C| - \phi' \\
&\leq (m-1)\omega' - \frac{m-1}{m+1} \omega' \\
&= m(m-1) - \frac{m-1}{m+1} \omega' \\
&= \frac{2m - 1}{m+1} \omega',
\end{align*}
\]
We then have
\[
\frac{\omega}{\omega'} \leq \frac{2m - 1}{m+1}.
\]

Claim 3 If $\frac{m^2+1}{m+1} \omega' < \sum p_j < \frac{m+1}{2} \omega$, then
\[
\frac{\omega}{\omega'} \leq \frac{2m - 1}{m} - \frac{1}{m} \alpha,
\]
where $\alpha = \frac{\phi'}{\omega'}$.

Proof Because $\frac{m^2+1}{m+1} \omega' < \sum p_j < \frac{m+1}{2} \omega$, then $\phi' = m\omega' - \sum p_j < m\omega' - \frac{m^2+1}{m+1} \omega' = \frac{m-1}{m+1} \omega'$. So $0 \leq \alpha < \frac{m-1}{m+1}$. Our approach is simple. We want to increase the idle time of the optimal preemptive schedule by adding a new task so that this case will become the case in Claim 2. We can then use the result in Claim 2.

Suppose that the original task system is $T = \{T_1, T_2, \ldots, T_n\}$. Now let $T_{n+1}$ with processing time $p_{n+1} = \frac{1}{m} \sum p_j - \frac{m^2+1}{m(m-1)} \phi'$ be added to the task system $T$. Assume that for any task $T_j$, $1 \leq j \leq n$, we have $T_j \prec T_{n+1}$. It is easy to see that the new optimal schedule with $T_{n+1}$ added is in fact the old optimal schedule with $T_{n+1}$ appended at the end. See FIG. 4.
Since \( \phi'_{\text{new}} = \phi' + (m-1)p_{n+1} = \phi' + \frac{m-1}{m^2} \sum p_j - \frac{m^2+1}{m^3} \phi' = \frac{m-1}{m^2} \sum p_j - \frac{1}{m^2} \phi' = \frac{m-1}{m^2+1} (\sum p_j + p_{n+1}) \), the new schedule actually falls into the case in Claim 2. Hence we have

\[
\frac{\omega'_{\text{new}}}{\omega'_{\text{new}}} = \frac{\omega + p_{n+1}}{\omega' + p_{n+1}} \leq \frac{2m}{m+1}.
\]

So,

\[
(m+1)\omega \leq 2m\omega' + (m-1)p_{n+1} = 2m\omega' + (m-1)\left(\frac{1}{m^2} \sum p_j - \frac{m^2+1}{m^3} \phi'\right) = 2m\omega' + \frac{m-1}{m^2} (\sum p_j + \phi') - \frac{m+1}{m} \phi' = 2m\omega' + \frac{m-1}{m} \omega' - \frac{m+1}{m} \phi' = \frac{(m+1)(2m-1)}{m} \omega' - \frac{m+1}{m} \phi'.
\]

We then have

\[
\frac{\omega}{\omega'} \leq \frac{2m-1}{m} - \frac{1}{m} \alpha.
\]

In Claim 3, the bound for the ratio \( \omega/\omega' \) depends on the value of \( \alpha \), as can be seen from FIG. 5. The larger \( \alpha \) is, the closer the ratio is to the one in Liu’s conjecture. If \( \alpha = 0 \), the ratio becomes the one in Lemma 2 given by Muntz.

Combining the results in the above three claims, we have the following theorem.

**Theorem 2** For arbitrary \( \prec \) and arbitrary \( m \),

\[
\frac{\omega}{\omega'} \leq \begin{cases} 
\frac{2m}{m+1} & \text{if } \sum p_j \geq \frac{m+1}{2} \omega \\
\frac{m-1}{m \alpha} & \text{else, for } \alpha = \frac{\omega}{\omega'} \in [0, \frac{m-1}{m+1}).
\end{cases}
\]

5 Conclusion

Liu’s conjecture states that the makespan \( \omega \) of the optimal nonpreemptive schedule is always within \( \frac{2m}{m+1} \) times of the makespan \( \omega' \) of the optimal preemptive schedule. In this paper, we have presented two new results on the conjecture. First, we have shown that when there is no precedence constraint among tasks, the conjecture holds. Second, when there exists precedence constraint, the upper bound for \( \omega' \) can be improved. We have considered three cases and proved that for two cases the conjecture is true while in the third case the upper bound is improved by \( \frac{1}{m} \alpha \) for some nonnegative \( \alpha \).

**Acknowledgement**

This research is supported in part by NSF grant CCR-9210372.
References


