Parallel Processing Letters,

A NOTE ON LIMITED PREEMPTION

E. G. COFFMAN, JR. and S. EVEN
Bell Labs, Lucent Technologies
700 Mountain Ave., Murray Hill, NJ 07974
E-mail: {egc, even}@research.bell-labs.com

Received XX 1997
Revised XX 1997
Communicated by

ABSTRACT
The concept of limited preemption is introduced, where a task can be preempted, but can not be moved from one processor to another. For optimal makespan scheduling on two processors, the worst case ratio of the makespan with no preemption to that with limited preemption is shown to be 4/3, while the worst case ratio of the makespan with limited preemption to that with unlimited preemption is 4/3 as well.

Keywords:

1. Introduction

An instance I of the two-processor makespan scheduling problem consists of a set of tasks, each with a given running time, and a *precedence graph* that specifies the partial order in which the tasks must be run, i.e., the graph is directed and acyclic, and if there is an edge from vertex (task) A to vertex B then A must be completed before B can start. At all times in a valid schedule S of I, a processor runs at most one task, no task is run simultaneously on both processors, and the precedence partial order is obeyed. The *makespan* of S is the length of time it takes to complete the running of all tasks; i.e. the earliest time when both processors are idle and every task is finished.

There are two well studied scheduling paradigms [1,3]: in one preemption is allowed at any time, and in the other preemption is forbidden. Allowing preemption means that a running task can be interrupted and resumed later. It is assumed that the total running time of the task is unaffected by the interruption. Denote by ω_P the shortest makespan of all possible schedules, allowing unlimited preemption. Similarly define ω_N, when no preemption is allowed. Coffman and Garey [2] proved that for every instance the following holds:

$$\frac{\omega_N}{\omega_P} \leq \frac{4}{3}\ . \tag{1}$$

They also pointed out the following example:

Example 1

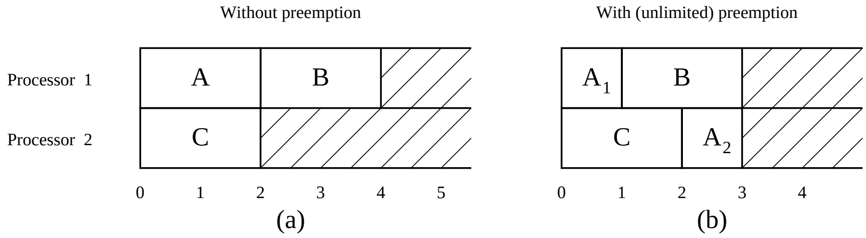

Figure 1:

Assume there are 3 tasks, A, B and C, each taking two units of time to run, and assume the precedence relation is empty, i.e., the tasks are independent and can be run in any order. The nonpreemptive schedule is shown in Figure 1(a), and takes 4 units of time. A preemptive schedule, where task A is preempted after one unit of processing time and is resumed later on the second processor, is shown in Figure 1(b). Clearly, in this case,

$$\frac{\omega_N}{\omega_P} = \frac{4}{3} , \tag{2}$$

proving that the the upper bound on the ratio, as in Equation 1, is tight.

Define *limited preemption* to mean that one is allowed to interrupt the running of a task, but may only resume running it on the *same* processor where it was interrupted. This is a natural assumption if moving the task from one processor to the other is itself a time-consuming task. For a given instance, let us denote by ω_L the shortest makespan possible if limited preemption is allowed. In the next section we show that the least upper bound of each of the ratios ω_N/ω_L and ω_L/ω_P is 4/3.

2. Comparing No Preemption to Limited Preemption

It is easy to see that for every instance the following holds:

$$\frac{\omega_N}{\omega_L} \leq \frac{\omega_N}{\omega_P} \leq \frac{4}{3} , \tag{3}$$

where the first inequality follows from the fact that $\omega_L \geq \omega_P$, and the second follows from Equation 1. However, it is also clear, from Example 1, that no smaller upper bound exists. Note that allowing limited preemption is of no use in this case.

3. Comparing Limited Preemption to (Unlimited) Preemption

In this case the upper bound of 4/3 is just as easy:

$$\frac{\omega_L}{\omega_P} \leq \frac{\omega_N}{\omega_P} \leq \frac{4}{3} , \tag{4}$$

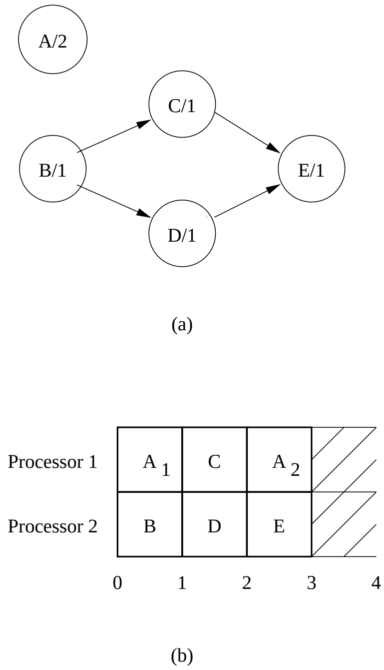

Figure 2:

where the first inequality follows from the fact that $\omega_L \leq \omega_N$, and the second follows again from Equation 1. The following example shows that no smaller upper bound exists.

Example 2

Assume there are 5 tasks, A, B, C, D and E, with running times 2, 1, 1, 1, 1, respectively. The precedence graph is shown in Figure 2(a). It is easy to see that if task A is not preempted then the makespan cannot be less than 4 units of time. However, if task A is broken into two parts, both running on the same processor, then as shown in Figure 2(b), the makespan can be cut to 3 units of time.

[1] Blazewicz, J., Ecker, K. H., Schmidt, G., and Weglarz, J., *Scheduling in Computer and Manufacturing Systems*, Springer-Verlag, New York, 1993, ISBN: 3540559582.

[2] Coffman E.G. Jr. and Garey, M.R., "Proof of the 4/3 Conjecture for Preemptive vs. Nonpreemptive Two-Processor Scheduling", *Journal of the ACM*, Vol. 40, No. 5, Nov. 1993, pp. 991-1018.

[3] Lawler, E. L., Lenstra, J. L., Rinnooy Kan, A. H. G., and Shmoys, D. B., "Sequencing and Scheduling: Algorithms and Complexity," in *Handbooks in Operations*

Research and Management Science, Vol. 4, North Holland, 1993, pp. 445-522.