Scheduling Independent Tasks with Due Times on a Uniform Processor System

SARTAJ SAHNI AND YOOKUN CHO

University of Minnesota, Minneapolis, Minnesota

ABSTRACT. An algorithm to preemptively schedule n tasks on m uniform processors is presented. It is assumed that each task is available at time 0. Associated with each task is a due time by which it is to be completed. The algorithm schedules all tasks to complete by their due times whenever possible. The asymptotic time complexity of the algorithm is $O(n \log n + mn)$. It generates $O(mn)$ preemptions in the worst case. An example of n tasks requiring $O(mn)$ preemptions is also presented. The algorithm can also be used when all tasks have the same due times but different release times.

KEY WORDS AND PHRASES: independent tasks, preemptive schedule, due time, uniform processors, complexity

CR CATEGORIES: 5.25, 5.3, 5.4

1. *Introduction*

Let $P = \{P_1, P_2, \ldots, P_m\}$ be a set of m processors. Let t_i, r_i, and d_i, $1 \leq i \leq n$, be the task times, release times, and due times, respectively, of n independent tasks. Associated with each processor P_i is a speed s_i, $s_i > 0$. Processor P_i has an effective processing capability of s_i units of processing per time unit. Task j can be processed on P_i in t_j/s_i units. The processors are said to be *uniform*, as they operate at a constant speed independent of time. When $s_i = 1$, $1 \leq i \leq m$, the processors are said to be *identical*. In this paper we are concerned only with preemptive schedules. A schedule S for the n tasks is a *DD-schedule* iff the processing of each task commences no earlier than its release time and completes no later than its due time. A DD-schedule will also be referred to as a *feasible* schedule.

For the case of identical processors, Horn [3] presents an $O(n^3)$ algorithm to obtain a DD-schedule for any set of n independent tasks for which such a schedule exists. Sahni [5] presents a fast algorithm that obtains DD-schedules (whenever they exist) when all tasks have either the same release time or the same due time. For the case of two uniform processors, Bruno and Gonzalez [1] present an $O(n^3)$ algorithm that obtains a DD-schedule (whenever one exists). Gonzalez and Sahni [2] have developed an $O(n + m \log m)$ algorithm that can be used when all tasks have the same release time and also the same due time.

In this paper we study the case when all tasks have the same release time. Different tasks may, however, have different due times. Our algorithm to construct a DD-schedule (if one exists) for this case takes $O(mn)$ time in the worst case. DD-schedules containing at most mn preemptions are generated. While the algorithm is discussed in terms of a set of

This research was supported in part by the National Science Foundation under Grants MCS 76-21024 and MCS 78-15455.

Authors' present addresses: S. Sahni, Department of Computer Science, University of Minnesota, 136 Lind Hall, 207 Church St. SE, Minneapolis, MN 55455; Y. Cho, Computer Science Department, Seoul National University, Seoul, Korea.