SIAM J. COMPUT.
Vol. 9, No. 1, February 1980

0097-5397/80/09001-0007 $01.00/0

BOUNDS FOR LIST SCHEDULES ON UNIFORM PROCESSORS*

YOOKUN CHO† AND SARTAJ SAHNI‡

Abstract. Bounds are derived for the worst case performance of list schedules relative to minimum finish time schedules for uniform processor systems. The tasks to be scheduled are assumed to be independent and only nonpreemptive schedules are considered.

Key words. list schedules, nonpreemptive schedules, uniform processors, independent tasks

1. Introduction. A uniform processor system consists of m, $m \geqq 1$, processors $P_1, P_2, \cdots, P_m$. Associated with each processor is a speed s_i, $s_i \geqq 1$. In one unit of time P_i can carry out s_i units of processing. Without loss of generality, we may assume $s_i \leqq s_{i+1}$, $1 \leqq i < m$ and $s_1 = 1$. We are given n independent tasks that are to be processed. Task i requires t_i units of processing (t_i is the *task time* of task i). If task i is assigned to P_j then t_i/s_j time units are needed to finish this task. A *nonpreemptive schedule* is an assignment of tasks to processors such that each task is assigned to exactly one processor. For each processor the order in which tasks are to be processed is also specified. If T_i is the set of tasks assigned to P_i then the *finish time* of P_i is $(\sum_{j \in T_i} t_j)/s_i$. The finish time of the schedule is the time at which all processors $\sum_{j \in T_i} t_j$ have finished processing.

For the case when $s_i = 1$, $1 \leqq i \leqq m$ the processor system defined above is known as a system of *identical processors*. It is well known that finding minimum finish time nonpreemptive schedules for identical processors with $m \geqq 2$ is *NP*-hard (see e.g. [7]). Several heuristics to obtain "near optimal" schedules for identical processors have been studied. Graham [4] has studied the performance of LPT schedules. In an LPT schedule tasks are assigned to processors in nonincreasing order of task times. Whenever a task is to be assigned, it is assigned to that processor on which it will finish earliest. Ties are broken arbitrarily and tasks are processed in the order assigned. Let $\hat{f}$ be the finish time of an LPT schedule for any given task set. Let f^* be the finish time of an optimal schedule. Graham [4] has shown that

$$\hat{f}/f^* \leqq 4/3 - 1/(3m).$$

Another heuristic studied by Graham is the list schedule. This scheduling rule differs from the LPT rule only in the order in which tasks are considered for assignment to processors. A list (or permutation) of the indices $1, 2, \cdots, n$ is provided. Tasks are considered in the order in which they appear on this list. If $\hat{f}$ is the finish time of a list schedule and f^* that of an optimal schedule for any given task set then it is known [3] that:

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

Hence, for identical processor systems LPT schedules are better than arbitrary list schedules by only a constant factor. Note that an LPT schedule is a special case of a list schedule (i.e., the case when the tasks in the list are ordered in nonincreasing order of task times). Other heuristics for identical processor systems have been studied by Coffman, Garey and Johnson [1] and Sahni [9].

---

* Received by the editors July 25, 1978. This research was supported in part by the National Science Foundation under Grant MCS 76-21024.

† Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455. Presently at Seoul National University, Korea.

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