SCHEDULING JOBS ON HETEROGENEOUS PROCESSORS

Rhonda RIGHTER
Department of Decision and Information Sciences, Santa Clara University, Santa Clara, CA 95053, USA

Susan XU
Department of Management Science, College of Business Administration, The Pennsylvania State University, University Park, PA 16802, USA

We consider the problem of scheduling $n$ jobs nonpreemptively on $m$ processors to minimize various weighted cost functions of job completion times. The time it takes processor $j$ to process a job is distributed exponentially with rate parameter $\mu_j$, independent of the other processors. Associated with job $i$ is a weight $w_i$. There are no precedence constraints and any job may be processed on any processor. Assume that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_m$ and $w_1 \geq w_2 \geq \cdots \geq w_n$. Then for certain weighted cost functions, the optimal policy is such that the processors can be partitioned into sets $S_1, \ldots, S_{n+1}$ such that if the fastest available processor is in set $S_t, i = 1, \ldots, n$, then job $i$ should be assigned to it, and if it is $S_{t+1}$, it will never be used. After each assignment the jobs are renumbered (so that job $i+1$ becomes job $i$ if job $i$ is assigned to a processor). The partitioning is independent of the job weights and the states (busy or idle) of the processors. The optimal policy can be determined in at most $\max\{m, n\}$ steps. If all the weights are identical, the optimal policy reduces to a simple threshold rule such that a job should be assigned to the fastest available processor, say $j$, if there are more than $K_j$ jobs waiting. $K_j$ will depend on $\mu_1, \ldots, \mu_j$ but not on $\mu_{j+1}, \ldots, \mu_m$. The optimal policy is also individually optimal in the sense that it minimizes the cost for each job $i$ subject to the constraint that processors will first be offered to the jobs in the order $1, 2, \ldots, n$.

We explicitly characterize the optimal policy for several specific examples of cost functions, such as weighted flow time, weighted discounted flowtime, and weighted number of tardy jobs.

1. Introduction

We have a system with $m$ processors where the processing time of processor $j$ is exponentially distributed with rate $\mu_j$. There are $n$ jobs with weights $w_i, i = 1, \ldots, n$, and we wish to minimize $E(\Sigma_{i=1}^n w_i f(C_i))$ where $C_i$ is the completion time of job $i$ for certain positive increasing cost functions, $f$. We are not allowed to preempt jobs. We show that for these cost functions the optimal policy is to assign job $i$ to the fastest available processor, say processor $j$, if $\nu_{i-1} \leq g(\mu_j) < \nu_i,$
where $0 = \nu_0 \leq \nu_1 \leq \cdots \leq \nu_n$ and where $\nu_i$ depends on $f$ and $\mu_j$, $j = 1, \ldots, n$, but not on the job weights or the total number of jobs, $n$, as long as $i \leq n$. Also, $\nu_i = E(f(C_i))$, the expected cost for job $i$, under the optimal policy if all processors are initially busy, and $g(\mu) = E(f(X))$ where $X$ is exponentially distributed with rate $\mu$. After each assignment the jobs are renumbered (so that job $i + 1$ becomes job $i$ if job $i$ is assigned to a processor). This policy is equivalent to partitioning the processors into sets $S_1, \ldots, S_{n+1}$ such that if the fastest available processor is in set $S_i$, $i = 1, \ldots, n$, then job $i$ should be assigned to it, and if it is in $S_{n+1}$, it will never be used. The partitioning is independent of the job weights and the states (busy or idle) of the processors. The optimal policy can be determined in at most $\max\{m, n\}$ steps. If all the job weights are identical, this policy reduces to a simple threshold rule such that a job should be assigned to the fastest available processor, say $j$, if there are more than $K_j$ jobs waiting. $K_j$ will depend on $\mu_1, \ldots, \mu_j$ but not on $\mu_{j+1}, \ldots, \mu_m$. The optimal policy is also individually optimal in the sense that it minimizes the cost for each job subject to the constraint that processors will first be offered to the jobs in decreasing order of their weights. Also, a processor that is declined by a job will never be used by that job in the future. Thus, by the result of Kumar and Walrand [10] the same policy will still be individually optimal when there are arbitrary arrivals of identical jobs and jobs are offered processors according to their position in queue.

Most work on parallel processor scheduling has assumed the processors are identical and the job processing times depend on the jobs and not the processors. (For example, Glazebrook [6], Pinedo and Weiss [14], Weber [22], Weber, Varaiya and Walrand [21], Weiss [24]). We assume that the processing times depend only on the processors. This may be more appropriate when, for example, we have a packet switching problem in which packets of the same size are to be transmitted over heterogeneous channels.

Kampe [8,9] and Weiss and Pinedo [23] considered preemptive scheduling of jobs on processors where the processing time depends on both the jobs and the processors. Coffman et al. [2] and Xu [25] studied the makespan problem for heterogeneous processors.

Agrawala et al. [1] showed that a threshold policy minimizes the expected flowtime of identical jobs on heterogeneous exponential processors when preemption is not permitted, and Xu [26] showed that the same policy stochastically minimizes flowtime. Righter [16] showed that the same policy minimizes expected weighted flowtime for arbitrary job weights. For related work see Lin and Kumar [11], Walrand [20], Nelson and Towsley [13], Derman, Lieberman and Ross [4], and Courcoubetis and Reiman [3].

This paper generalizes the results of Righter [15,16] and the approach is based on the sequential stochastic assignment problem first considered by Derman, Lieberman and Ross [5]. In the assignment problem there are $n$ activities (jobs) and resources (processors) arrive according to a Poisson process. Activities have