Adv. Appl. Prob. **19**, 749–750 (1987)
Printed in N. Ireland

NECESSARY OPTIMALITY CONDITIONS FOR PRIORITY POLICIES IN STOCHASTIC WEIGHTED FLOWTIME SCHEDULING PROBLEMS

THOMAS KÄMPKE,* *Universität Passau*

Abstract

Conditions implied by the optimality of static priority policies in a special class of stochastic scheduling problems will be derived.

STOCHASTIC SCHEDULING, STATIC PRIORITY POLICY

A recent paper, Kämpke (1987), considered the problem of processing a set of jobs $\{1, \cdots, n\}$ on n parallel processors (or machines) with possibly differing speeds $s_1 \geqq \cdots \geqq s_n \geqq 0$, $s_1 > 0$, a speed of a processor being the rate at which it performs work. It was assumed that the service times of the jobs are independent exponentially distributed with parameters $\lambda_1, \cdots, \lambda_n$; service times refer to machines with speed 1. When a job with parameter λ_i is processed on a machine with speed s_j until its completion, the actual processing time is exponentially distributed with parameter $\lambda_i s_j$. Preemption was permitted at all times. Let t_i denote the time at which job i is completed. The paper showed that the condition

$$\text{(OPT)} \qquad \lambda_1 w_1 \geqq \cdots \geqq \lambda_n w_n \quad \text{and} \quad w_1 \geqq \cdots \geqq w_n$$

is sufficient for the weighted flowtime $\sum_{i=1}^n w_i t_i$ (w_i denoting some positive weight of job i) to be minimized in expectation by the static priority policy which always assigns the uncompleted job of ith least index amongst uncompleted jobs to the ith machine. A priority order on the jobs will be denoted by $1 \subset \cdots \subset n$, giving job i the ith greatest priority. The static priority policy associated with this priority always processes the uncompleted job of ith greatest priority amongst uncompleted jobs on processors i. The given result generalizes certain results of Weiss and Pinedo (1980). More recently, Weber (1986) has established the sufficiency of this condition when jobs have more general service time distributions and all processors have the same speed.

The purpose of this letter is to remark that (OPT) is also a necessary condition if a static priority policy is to be optimal for all constellations of machine speeds $s_1 \geqq \cdots \geqq s_n \geqq 0$, $s_1 > 0$. To see this, first consider the case $s_1 = 1$, $s_2 = \cdots = s_n = 0$. The static priority $1 \subset \cdots \subset n$ is optimal iff $w_1/\mathrm{EX}_1 \geqq \cdots \geqq w_n/\mathrm{EX}_n$, (for example, see the proof of Smith's rule in Conway et al. (1967)), where EX_i is the mean service time of job i. Because $\mathrm{EX}_i = 1/\lambda_i$ it remains to show $w_1 \geqq \cdots \geqq w_n$. In the situation $s_1 > \cdots > s_n > 0$ with positive probability any two jobs u and v with $u < v$ may be the last to be completed. Let G_{uv} (respectively G_{vu}) denote the expectation of the sum of the weighted flowtimes of these two jobs when u (respectively v) is assigned to the fastest processor and v (respectively u) to the next fastest processor. The optimality of $1 \subset \cdots \subset n$, the memoryless property of the exponential distribution and the allowed

Received 9 December 1986; revision received 7 April 1987.

* Postal address: Lehrstuhl für Informatik und OR, Universität Passau, Innstrasse 27, 8390 Passau, West Germany.

preemptions imply $G_{uv} - G_{vu} \leqq 0$. Thus simple calculations lead to

$$w_v(s_1 - s_2)/(\lambda_u s_1 + \lambda_v s_2) - w_u(s_1 - s_2)/(\lambda_u s_2 + \lambda_v s_1) \leqq 0.$$

Now, $s_2 \to s_1$ results in $w_u \geqq w_v$, completing the argument.

Allowing processors of different speeds is essential. Consider processing of four jobs on 1, 2 and 3 machines of identical speeds. Suppose the jobs have weights $w = (w_1, \cdots, w_4)$ and parameters $\lambda = (\lambda_1, \cdots, \lambda_4)$. There are examples which show that if (OPT) does not hold: (a) there may be a single static priority policy which is optimal for any number of processors $\leqq 3$, $w = (10^6, 10^6, 10, 10 + 10^{-6})$, $\lambda = (10^6, 10^6, 10, 1)$, (b) there may be optimal static priority policies, but they may depend on the number of processors, $w = (10^7, 10^6, 10, 15)$, $\lambda = (10^6, 10^6, 20, 10)$, and (c) there may be no static priority policy which is optimal, $w = (10^6, 10^6, 2, 10)$, $\lambda = (10^6, 1, 10, 1)$. In case (a) $1 \subset 2 \subset 3 \subset 4$ is optimal and in case (b) $1 \subset 2 \subset 3 \subset 4$ is optimal for 1 processor and $1 \subset 2 \subset 4 \subset 3$ is optimal for 2 and 3 processors. In case (c) $1 \subset 2 \subset 3 \subset 4$ is optimal for 1 processor, $1 \subset 2 \subset 4 \subset 3$ is optimal for 3 processors and for 2 processors it is optimal to start with jobs 1 and 2; if job 1 finishes before job 2, continue with $2 \subset 3 \subset 4$, otherwise with $1 \subset 4 \subset 3$.

References

CONWAY, R. W., MAXWELL, W. L. AND MILLER, L. W. (1967) *Theory of Scheduling.* Addison-Wesley, Reading.

KÄMPKE, T. (1987) On the optimality of static priority policies in stochastic scheduling on parallel machines. *J. Appl. Prob.* **24,** 430–448.

WEBER, R. R. (1986) Stochastic scheduling on parallel processors and minimization of concave functions of completion times. Preprint.

WEISS, G. AND PINEDO, M. (1980) Scheduling tasks with exponential service times on non-identical processors to minimize various cost functions. *J. Appl. Prob.* **17,** 187–202.