Replication and sequencing of unreliable jobs on parallel machines

Alessandro Agnetis, Mario Benini, Paolo Detti, Marco Pranzo
Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche,
Università di Siena, via Roma 56, Siena
{alessandro.agnetis, paolo.detti, marco.pranzo}@unisi.it, mbenini@diism.unisi.it
Ben Hermans
Research Center for Operations Research and Statistics,
Katholieke Universitat Leuven, Belgium
ben.hermans@kuleuven.be

Abstract. In *unreliable job scheduling problems*, each job j has a certain probability π_j of being successfully carried out, in which case it brings a revenue R_j. If a machine fails during the execution of a job, the current and all subsequently scheduled jobs on that machine are lost (and produce no revenue). The problem is how to sequence the jobs on each machine to maximize the expected revenue. The problem is easily solvable for $m = 1$ while it is strongly NP-hard for $m \geq 2$. For general m, list scheduling provides a 0.8531-approximate solution. In order to increase the chances of successful completion of a job, a strategy consists of *replicating* the jobs on various machines. This is a strategy used e.g. in some computer systems to execute programs on different machines, in order to hedge against unrecoverable interruptions. However, it must be taken into account that even if two or more copies of the same job are successfully completed on different machines, the revenue is attained only once. In this context, we consider two distinct problems. Given n unreliable jobs, $m \geq 2$ machines, and m copies of each job (one per machine), the first problem consists of finding the job sequence on each machine that maximizes the expected revenue. We show that for $m = 2$ this problem is NP-hard. We propose a mathematical programming formulation and assess the behavior of some heuristic algorithms. The second problem consists in finding the sequence on each machine that maximizes the probability that *at least one copy* of each job is successfully carried out. We show that this problem can be easily solved when $m = 2$, as well as for m machines and two jobs.

Keywords: Unreliable jobs, complexity, job replication.