Schedule on parallel machines in the case of individual machine-set

Zsuzsanna Vaik

There is a less researched area of the parallel machines scheduling, when there is an M_j machine-set for each job j, where it can be scheduled. We would like to minimize the latest job's finishing time, that is the makespan, C_{max} so, that each job can be processed only by one machine at a time and one machine can process at most one job at a time. This problem is $\mathcal{NP}$-complete, since its special case, the $P\|C_{max}$ problem, (when each job can be processed by each machine), is known to be $\mathcal{NP}$-complete [1]. If we have restrictions for the M_j machine-sets, the well-known list-schedule gives a nearly optimal solution. We give a better approximation algorithm for the optional problem.

M. Pinedo [2] has studied that special case of the problem, when the job's processing times are 1, and he has showed that,if the M_j sets are laminals, then an easy list-schedule gives an optimal solution. We show for the general case, when the M_j sets are optional ($P|M_j, p_j = 1|C_{max}$), that it is a network-flow-problem. Moreover, from this, we have an algorythm for that special case, when we allow the preemtion for the jobs ($P|M_j, pmtn|C_{max}$). From this algorithm we have received a minimax formula for the optimal makespan too.

We study the problem when we allow a special preemption, called $pmtn^*$, when a job can be split, but it should be processed immediately by another machine. This special preemption gives a better solution, that we show in an example, that the optimal value of the problem $P|M_j, pmtn^*|C_{max}$ is less than the equivalent $P|M_j|C_{max}$ problems optimal value. But we can observe, that if $M_j = M$ for each job, than the special preemption gives no better optimal value, so in this case the problem is equivalent with the $P\|C_{max}$ problem, which is $\mathcal{NP}$-complete. We have a 2-approximate algorythm for this problem too.

References

[1] M.R.Garey, D.S. Johnson [1978]: Strong NP- completeness results: motivaton, examples and implications, Journal of the Association for Computing Machinery 25, 499-508

[2] M.Pinedo [2002]: Scheduling Theory, Algorithms, and Systems, Second Edition, Prentice Hall