On Flowshop and Jobshop Schedules

Extended Abstract

Teofilo Gonzalez & Sartaj Sahni

1. Introduction

A shop consists of $m \geq 1$ processors (or machines). Each of these processors performs a different task. There are $n \geq 1$ jobs. Each job i has m tasks. The processing time for task j of job i is $t_{j,i}$. Task j of job i is to be processed on processor j, $1 \leq j \leq m$. A schedule for processor j is a sequence of tuples $(\ell_i, s_{\ell_i}, f_{\ell_i})$, $1 \leq i \leq r$. The ℓ_i are job indexes, s_{ℓ_i} is the start time of job ℓ_i and f_{ℓ_i} is its finish time. Job ℓ_i is processed continuously on processor j from s_{ℓ_i} to f_{ℓ_i}. The tuples in the schedule are ordered such that $s_{\ell_i} < f_{\ell_i} \leq s_{\ell_{i+1}}$, $1 \leq i < r$. There may be more than one tuple per job and it is assumed that $\ell_i \neq \ell_{i+1}$, $1 \leq i < r$. It is also required that each job i spends exactly $t_{j,i}$ total time on processor j. A schedule for a m-shop is a set of m processor schedules. One for each processor in the shop. In addition these m processor schedules must be such that no job is to be processes simultaneously on 2 or more processors. A shop schedule will be abbreviated to schedule in future references. The finish time of a schedule is the latest completion time of the individual processor schedules and represents the time at which all tasks have been completed. An optimal finish time (OFT) schedule is one which has the least finish time amongst all schedules. A non-preemptive schedule is one in which individual processor schedules have at most one tuple (i, s_i, f_i) for each job i to be scheduled. For any processor j, this allows for $t_{j,i} = 0$ and also requires that $f_i - s_i = t_{j,i}$. A schedule in which no restriction is placed on the number of tuples per job per processor is preemptive. Note that all non-preemptive schedules are also preemptive while the reverse is not true. For any schedule, S, we define the finish time, $f_i(S)$ of job i to be the earliest time at which all tasks of job i have been completed. The mean flow time, mft(S), is defined to be the quantity $\Sigma f_i(S)$. (Note that mft(S) is not actually the average finish time, given by mft(S)/n. However minimizing mft(S) is equivalent to minimizing the average finish time.) An optimal mean flow time (OMFT) schedule is a schedule which has the least mft amongst all possible schedules for that job set.

In this paper we shall investigate optimal schedules for the following shop models:

I) Flow Shop

There are $n \geq 1$ jobs to be scheduled on $m \geq 1$ processors with the restriction that for every job i, the processing of task $j + 1$ cannot begin until the processing of task j is complete, $1 \leq j < m$.

II) Job Shop

There are $n \geq 1$ jobs to be scheduled on $m \geq 1$ processors with the restriction that the tasks for each job are ordered. The processing of a task cannot begin until the processing of all tasks preceding it have been completed. Several tasks may be specified for an individual processor. The notation $t_{j,i,k}$ will be used to indicate the kth task of job i. This task is to be processed on processor j.

Several strategies for obtaining OFT and OMFT schedules for flow shops and job shops have been advanced (see for example [2] and [3]). Branch and bound strategies for these problems are investigated in [7] and [9]. Despite all the research effort devoted to these problems there are no known efficient algorithms. In [1] and [4] it is shown that these problems are NP-Complete when one is restricted to nonpreemptive schedules. For details regarding our notation of NP-Complete problems, the reader is referred to [8] and [10]. In section 2 we extend the NP-Completeness results of [1] and [4] for OFT nonpreemptive schedules to the case of preemptive schedules. A more restricted version of the OFT nonpreemptive flow shop problem is also shown to be NP-Complete. In section 3 we obtain bounds comparing optimal and arbitrary active schedules for flow shops and job shops. In this section we also present heuristics that result in schedules with a mft and finish time better than the worst active schedules. Finally, a comparison is made between the finish times of preemptive and nonpreemptive schedules.

2. Complexity of Preemptive and Nonpreemptive Scheduling

2.1 Flow Shop

OFT nonpreemptive schedules for the two processor (m=2) flow shop can be obtained in O(n logn) time using Johnson's algorithm [3, p. 83]. For the case $m=2$ one can easily show that an OFT preemptive schedule has the same finish time as an OFT nonpreemptive schedule. Hence, Johnson's algorithm also gives an OFT preemptive schedule. A linear time algorithm is presented in [11], which guarantees OFT preemptive and nonpreemptive schedules for the two processor open shop. It is interesting to note that by eliminating the task ordering, a more efficient algorithm is obtained.

In this section we show that when $m > 2$ finding OFT preemptive and nonpreemptive flow shop schedules is NP-Complete. This is true even when the jobs are restricted to have at most two non-zero tasks each. This then gives us the simplest case of the flow shop problem that is both NP-Complete and for which no polynomial algorithm is known (note that when jobs have only 1 task per job, OFT schedules may be trivially obtained). When the complexity of the algorithm is measured in terms of the sum of the lengths of the tasks, the flow shop problem remains NP-Complete as shown in [4]. This result can also be extended for preemptive schedules. However the job with the maximum number of tasks is increased to 3. In [11] it is shown that for $m > 2$, the OFT preemptive schedules for the open shop can be obtained in polynomial time. For nonpreemptive schedules, the problem remains NP-Complete.

2.2 Job Shop

The preceding results trivially imply that a severely restricted form of the job shop problem for $m > 2$ is NP-Complete. For the job shop with $m=2$, however, no polynomial time algorithm is known. In [3, p. 105] an $O(n \log n)$ algorithm to obtain OFT nonpreemptive schedules when $m = 2$ and the jobs are restricted to have at most two nonzero tasks is presented. For this case, OFT preemptive schedules may be similarly obtained. For the nonpreemptive case, it is known [1,2] that when $m = 2$ and the job mix consists of $n-1$ jobs with one nonzero task each and an additional job with three nonzero tasks then the problem is NP-Complete. We extend this result to the case of preemptive schedules. We show that finding OFT preemptive schedules when $m=2$ is NP-Complete even when the job mix contains only two jobs with three nonzero tasks. In [4] it is shown that, if the complexity is measured in terms of the sum of the length of the tasks, the two processor is also NP-Complete. This result can also be extended to include preemptive schedules, however there is a job with N/k tasks, for some constant k.

3. Approximate Solutions

Since the problems of finding OFT and OMFT schedules for flow shops and job shops is NP-Complete (see [4] for NP-Completeness of OMFT) we turn our attention to obtaining schedules whose performance approximates that of optimal schedules. To begin with, we derive a bound for the ratio of worst and best schedules for the two performance measures being considered. We then present approximation algorithms that generate schedules with a worst case bound smaller than this. In examining "worst" schedules, we restrict ourselves to active schedules. An active schedule is a schedule in which at all times from start to finish some processor is busy (i.e. it is processing a task). For a given set of jobs and a schedule S we denote by f(S) the finish time of S and by mft(S) the mean flow time of S. We show that for any active schedule $mft(S)/mft(S^*) \le n$.

Since rather crude approximations are used to obtain this bound, it is surprising that Johnson's OFT algorithm on 2 processors achieves this bound. I.e. there are OFT schedules S such that $mft(S)/mft(S^*) = n$. Note that the bound of n holds for job shop schedules.

A simple heuristic that results in schedules with a mft which in the worst case is closer to the optimal is obtained by processing jobs in order of nonincreasing L_i (L_i = sum of task times for job i). This heuristic will be referred to as SPT (see [3], p. 76). It is shown that this heuristic produces schedules with $mft(S)/mft(S^*) = m$.

Let us now turn our attention to the finish time properties of active schedules. For this case we show that any active schedule has $f(S)/f(S^*) \le m$.

Once again, as in the case of mft, the proof technique would seem to indicate that any "reasonable" heuristic would result in schedules with a worst case bound less than m. This unfortunately is not the case. We define by LPT the heuristic: schedule jobs in order of nondecreasing L_i. Note that this heuristic is similar to the one used by Graham [6] to schedule identical processors and by Gonzalez, Ibarra and Sahni [5] to schedule uniform processors. In both these earlier applications of the heuristic, LPT schedules had a worst case finish time at most a "small" constant times the optimal finish time. This is no longer the case for flow shop and job shop schedules. For this LPT heuristic, we show that the bound of m is tight.

The worst case bound of m for active schedules in a flow shop can be reduced to $\lceil m/2 \rceil$ by using the following heuristic H: Divide the m processors into $\lceil m/2 \rceil$ groups each group containing at most two processors. The processors in group i are the 2i-1st and 2ith ones. Johnson's O(n logn) algorithm is used to obtain optimal finish time schedules for each of these $\lceil m/2 \rceil$ groups of machines. These $\lceil m/2 \rceil$ optimal schedules are then concatenated to obtain a schedule for the original m processor flow shop.

Since an optimal two processor flow shop schedule can be obtained in time O(nlogn), the total time needed by algorithm H is O(m nlogn).

We close this section with a comparison of OFT preemptive and nonpreemptive schedules. If S_p^* and S_n^* are OFT preemptive and nonpreemptive schedules respectively, then from algorithm H it follows that $f(S_n^*)/f(S_p^*) \leq \lceil m/2 \rceil$. We then show that this bound is tight for m=3 and m=4.

For the case of a job shop, we conclude from the property of active schedules that $f(S_n^*)/f(S_p^*) \leq m$. We show that this is a tight bound for m=2.

References

1. Brucker, P., Lenstra, J.K. and Rinooy Kan, A.H.G., "Complexity of Machine Scheduling Problems", Mathematisch Centrum, Amsterdam, Technical Report BW 43/75, April 1975.

2. Coffman, E.G. Jr., "Computer and Job Shop Scheduling Theory", John Wiley and Sons, to appear.

3. Conway, R.W., Maxwell, W.L.M. and Miller, L.W., "Theory of Scheduling", Addison-Wesley, 1967.

4. Garey, M.R., Johnson, D.S. and Sethi, R., "Complexity of Flow Shop and Job Shop Scheduling", University of Pennsylvania, Technical Report #168, June 1975.

5. Gonzalez, T., Ibarra, O.H. and Sahni, S., "Bounds for LPT Schedules on Uniform Processors", University of Minnesota, Technical Report #75-1, Jan. 1975.

6. Graham, R.L., "Bounds on Multiprocessing Timing Anomalies", SIAM Journal on Applied Mathematics, Vol. 17, No. 2, March 1969.

7. Ignall, E. and Schrage, L., "Application of the Branch and Bound Technique to Some Flow Shop Scheduling Problems", Operation Research 13, No. 3, May 1965.

8. Karp, R.M., "Reducibility Among Combinatorial Problems", Complexity of Computer Computation, R.E. Miller and J.W. Thatcher (Eds.), Plenum Press, New York, 1972, pp. 85-104.

9. Kohler, W.H. and Steiglitz, K., "Exact, Approximate and Guaranteed Accuracy Algorithms for the Flow Shop Problem n/2/F/F", JACM, Vol. 22, #1, Jan. 1975, pp. 106-114.

10. Sahni, S., "Computationally Related Problems", SIAM Journal on Computing, Vol. 3, No. 4, Dec. 1974, pp. 262-279.

11. Gonzalez, T. and Sahni, S., "Open Shop Scheduling to Minimize Finish Time", to appear in JACM.