International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.1, Issue No.6, pg : 269-271 01 Dec. 2012

Solving Flow-shop Sequencing Problem using Scheduling Algorithm based on Search and Prune Technique

Smita Verma, Megha Jain and Ranjana Singh
Department of Applied Mathematics & Computational Science,
S.G.S. Institute of Technology & Science, 23-Park Road, Indore-452003 (M.P.)
Email: sverma@sgsits.ac.in

Abstract
In sequencing problems, there are one or more jobs to be done and one or more machines are available for this purpose. If at least one machine center includes more than one machine, the scheduling problem becomes a flexible flow-shop problem. Flexible flow shops are thus generalization of simple flow shops. In this paper, we are solving flexible flow shop sequencing problem of two machine centers, using an algorithm. The proposed algorithm will be suitable for a medium-sized number of jobs; it is an optimal algorithm, entirely using the search-and-prune technique

I: Introduction:
A heuristic algorithm for solving flexible flow –shop problems of two machine centers is proposed by Sriskandarajah and Sethi in 1989. In this paper we use a scheduling algorithm based on the Search-and-prune technique to get the optimal solutions, which can also be used to measure the performance of the algorithm by some other algorithms. The proposed optimal algorithm is stated below in section 2.
The search-and-prune procedure proposed in this paper is used to schedule jobs for a flow shop with more than two machines. An upper bound is used to increase the performance of the procedure.
Given a set of *n* flow-shop jobs, each having *m (m>2)* tasks ($T_{11}, T_{21}, \ldots, T_{m1}, T_{12}, T_{22}, \ldots, T_{(m-1)n}, T_{mn}$) that must be executed in the same sequence on m machines ($P_1, P_2, \ldots, P_m$), scheduling the minimum completion time of the last job.

II. Optimal flexible flow-shop algorithm with an illustration:
Here input is a set of *n* jobs, each having *m (m>2)* tasks, to be executed respectively on each of *m* machine centers with *p* parallel machine and output is a schedule with a suboptimal completion time.
Considering an example parallel to the algorithm, five jobs, J_1 to J_5, each having three tasks (t_{1j}, t_{2j}, t_{3j}) that are to be scheduled via three operations. Each operation is executed by a machine center includes two parallel machines. Assume the execution times of these jobs are listed in Table 1.

Table 1: Processing times for the five jobs.

Execution Time	Jobs	J_1	J_2	J_3	J_4	J_5
t_{1i}		3	2	5	2	4
t_{2i}		7	3	5	5	6
t_{3i}		5	3	1	3	6

Input: A Set of n jobs, each having m (m >2) tasks, to be executed respectively on each of m machine centers with p parallel machines.
Output: A Schedule with on optimal completion time.

Step 1: Set the initial upper bound $v_{\max}$ of the final completion time as ∞.
Here, assuming the Upper bound $v_{\max}$ to be 28.

Step 2: For each possible combination of task allocation and permutation of task sequence, do the following steps.

Step 3: In each machine center, set the initial completion time of each machine to zero.
Initialize $d_1 = d_2 = 0$, where d_i is the initial completion time of D_{ji}.

Step 4: Set the variable g to one, where g represents the number of the current machine center to be processed.

Step 5: Schedule the first tasks of all jobs in the machines of the first machine center. That is, for each task T_{1i} of the i-th job allocated to the j-th machine D_{j1} in the first machine center, do the following sub steps according to the scheduling order in the permutation and combination generated.
(a) Add the processing time t_{1i} to the completion time d_{j1} of the machine D_{j1}. That is:

$d_{j1} = d_{j1} + t_{1i}$ and $c_{1i} = d_{j1}$.

$d_{11} = 0 + 4 = 4$ And $c_{1i} = d_{11} = 4$

$d_{21} = 4 + 1 = 5$ And $c_{1i} = d_{21} = 5$

$d_{31} = 5 + 5 = 10$ And $c_{1i} = d_{31} = 10$

$d_{41} = 10 + 2 = 12$ And $c_{1i} = d_{41} = 12$

$d_{51} = 12 + 5 = 17$ And $c_{1i} = d_{51} = 17$

(a) If 17 is not larger than 28($v_{\max}$) then

(b) If d_{j1} is larger than $v_{\max}$, neglect all the permutations and combinations with this sequence in the first machine center and go to Step 2 for trying Another permutation and combination. But here, this step is not applicable.

Step 6: Set g = g + 1, i.e., Set g = 1+1 = 2.

Step 7: Schedule the g-th tasks of all jobs in the machines of the g-th machine centers according to the permutation and combination generated. For each task T_{gi} of the i-th job allocated to the j-th machine D_{jg} in the g-th machine center, do the following sub steps in the scheduled order:

(a) Find the completion time d_{jg} of the machine D_{jg} as:

d_{jg} = max ($d_{jg}, c_{(g-1)i}$) + t_{gi}, and $c_{gi} = d_{jg}$.

d_{12} = max (0, 4) + 8 = 12 and $c_{2i} = d_{12} = 12$

d_{22} = max (0, 5) + 5 = 10 and $c_{2i} = d_{22}$ = 10

d_{32} = max (0, 10) + 2 =12 and $c_{2i} = d_{32}$ = 12

d_{42} = max (0, 12) + 5 = 17 and $c_{2i} = d_{42}$ = 17

d_{52} = Max (0, 17) = 5 = 22 and $c_{2i} = d_{52}$ = 22

(b) If d_{jg} is larger than $v_{\max}$, neglect all the permutations and combinations With this sequence in the first g machine centers and go to step 2 for Trying another permutation and combination. But as 22 is not greater than 28, this step is not applicable.

Step 8: Repeat Steps 6 and 7 until $g > m$.

Set g = 2 + 1= 3

(a) d_{13} = max (0,12) + 3 =15 and $c_{3i} = d_{13}$ = 15

d_{23} = max (0, 10) + 2 = 12 and $c_{3i} = d_{23}$ = 12

d_{33} = max (0.12) + 4 = 16 and $c_{3i} = d_{33}$ = 16

d_{43} = max (0, 17) + 3 = 20 and $c_{3i} = d_{43}$ = 20

d_{53} = max (0, 22) + 6 =28 and $c_{3i} = d_{53}$ = 28

(b) If 28 is equal to 28, then

d_{13} = 0 + 3 = 3 and $c_{3i} = d_{13}$ = 3

d_{23} = 3 + 2 = 5 and $c_{3i} = d_{23}$ = 5

d_{33} = 5 + 4 = 9 and $c_{3i} = d_{33}$ = 9

d_{43} = 9 + 3 = 12 and $c_{3i} = d_{43}$ = 12

d_{53} = 12 + 6 = 18 and $c_{3i} = d_{53}$ = 18

Step 9: Set the completion time d_m of the current schedule = $\max_{j=1}^{p}(d_{jm})$ among the p machines in the m-th machine center.

$$= \max_{j=1}^{p} (17,22,18) = 22.$$

Among the p machines than in the m-th machine center.

Step 10: If d_m is smaller than $v_{\max}$, then set $v_{\max} = d_m$.

If 22 is smaller than 28($v_{\max}$), then Set $v_{\max}$ = 22

Step 11: Repeat Steps 2 to 10 until all the possible permutation and combinations have been tested.

Step 12: Set the optimal final completion time of the job scheduling $ff = v_{\max}$.

Step 11: Set the optimal final completion time of the job scheduling

$$ff = 22$$

After this step, a globally optimal completion time ff has been found.

III. Conclusion:

In this paper, we have proposed an algorithm to solve flexible flow-shop problems of more than two machine centers. It is an extension of Sriskandarajah and Sethi's method. This algorithm is suitable for a medium-sized number of jobs; it is an optimal algorithm, entirely using the search-and-prune technique. It can work only when the job number is small.

References

1. Campbell, H. G., R. A. Dudek and M. L. Smith, 1970. A heuristic algorithm for the *n* job, *m* machine sequencing problem. Management Science. 16: B630-b637.

2. Chitale A.K., A Text book of Operations Research, Jain Bros. , , New Delhi .

3. Hardgrave, W. W., and Nemhauser, G, L., "A Geometric Model and a Graphical Algorithm for a Sequencing Problem," Operations Research, 11, 889-900 (1963).

4. Hong T. P., P. Y. Huang, G. Horng and C. L. Wang, Three Algorithms for Flexible Flow-shop Scheduling, American Journal of Applied Sciences 4 (11): 889-896, 2007.

5. Johnson, S. M., 1954. Optimal two- and three-stage production schedules with set-up times included. Naval Research Logistics Quarterly, 1: 61-68.

6. Logendran, R. and N. Nudtasomboon, 1991. Minimizing the make span of a group scheduling problem: a new heuristic. International Journal of Production Economics, 22: 217-230.

7. Sriskandarajah, C. and S. P. Sethi, 1989. Scheduling algorithms for flexible flow shops: worst and average case performance. European Jounal of Operational Research, 43: 143-160.