AN ALGORITHM FOR SOLVING N-JOB, M-MACHINE FLOW SHOP SCHEDULING PROBLEM WITH OBJECTIVE OF MINIMIZING MAKESPAN

G. Vasudevan
Research Scholar PG & Research Department of Mathematics, Arignar Anna Government Arts College, Cheyyar, Tamil Nadu, India.

Dr. S. Jayakumar
Assistant Professor & Head, PG & Research Department of Mathematics, Arignar Anna Government Arts College, Cheyyar, Tamil Nadu, India.

ABSTRACT

In this paper the authors investigate the problem of minimizing the Makespan using Heuristic approach. The author compared the result with the existing algorithms namely Palmer’s Heuristic (a slope order index), CDS Heuristic, NEH Heuristic algorithm, Gupta heuristic, RA (rapid access) Heuristic found in the literature and it was found that our algorithm perform superior than the other algorithms found in the literature.

Keywords: Heuristic, Makespan, Flow Shop Scheduling problem, Jayavasudev.

http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=11&IType=11

1. INTRODUCTION

Sequence problem has its origin in operator theory, manufacturing industries, queuing discipline, time table algorithm, etc., Suppose there are ‘n’ jobs to perform on ‘m’ different machines the effectiveness is measured by any given sequence of jobs at each machine and the most suitable sequence is selected. Therefore we get n! Theoretically possible sequence. It is always possible to select the best sequence but it is practically impossible because of large number of computations in particular \( n = 8 \) the total number of possible sequence will be \( 8! = 40320 \).
An Algorithm for Solving n-Job, m-Machine Flow Shop Scheduling problem with Objective of Minimizing Makespan

Hence effectiveness can be computed before selecting most suitable one but this approach is practically impossible to adopt. Hence we look into Heuristic to approach to solve this type of problems.

Flow shop scheduling is a decision making procedure that is used on a regular basis in many manufacturing and service industries. The purpose is to optimize the objectives with the allocations of resources over a period of time. Resources may be in a work shop machines, crews at a construction site and runway of the airport. It plays an important role in most manufacturing and service system as well as information processing environment some of the objectives for this type of problem is total job completion time, total flow time, Makespan, tardiness, lateness, due date, etc.

2. LITERATURE REVIEW
One of the prior Heuristic Johnson’s algorithm (1954) consider for the two machine flow shop problem with objective of minimizing Makespan after that in 1965 Palmer proposed a Heuristic algorithm which is a slope order index to sequence of jobs on the machines based on the processing time and known as the Palmer’s Heuristic for the flow shop scheduling problem. It was to give priority job so that jobs with processing times that tends to increase from machine to machines will receive higher priority for flow shop scheduling problem.

Campbell, Dudek and Smith (CDS) (1970) proposed a Heuristic that was extension of Johnson’s Algorithm for a FSSP with Makespan minimization.

Gupta [1971] suggested another heuristic which was similar to Palmer’s heuristic. He defined the slope index in a different manner by taking into account some attractive facts about optimality of Johnson’s rule for the three machine problems.

Dannenbring [1977] developed a Heuristic algorithm called RA (rapid access) which combines the merits of Palmers Heuristic (slope index) and the CDS algorithms. Its result is to give a best solution as fast and simply as feasible. As an alternative of solving m-1 artificial two machine problems, it solves only one artificial problem using Johnson’s algorithm(1954) in which the processing times are decide from a stand by scheme.

The Nawaz, Enscore, and Ham (NEH) [1983] heuristic algorithm is based on the assumption that a job with high total processing time on all the machines should be given higher priority than job with low total processing time. The NEH algorithm does not transform that original m-machine problem in to one artificial two-machine problem. It builds the final sequence in a constructive way, adding a new job at each step and finding the best partial solution.

3. TERMINOLOGY AND NOTATION
1. Number of machines: for example DD obtained from the teller of the bank and it has been proceed through filling the application, submitted to the teller, the teller in turn handed over to the manager and finally we collect the DD.
   It is a single job problem with four machines (works or services).
2. Processing order it refers to that order in which various machines are require to finish the job.
3. Processing time: the notation $P_{ij}$ will denote the processing time required for the $i^{th}$ job on the $j^{th}$ machine ($i = 1,2,3, ... n, j = 1,2,3, ... m$).
4. Minimizing the Makespan: this is the time of completion of last job on last machine.
5. No passing rule that is operation once started must be complete before going into the next machine.
4. ASSUMPTION IN A FSSP
1. A machine can process only one operation at a time.
2. Each operation once started must be performed till its completion.
3. There are only one type of each machine.
4. The time interval for processing are independent of the other in which operations are performed.
5. Each operation must be computed before any other operation which it must be proceed, can begin.
6. A job is processed based on ordering requirements.
7. All jobs must be started and ready at time $t = 0$.
8. The time require to transfer a job between machines is negligible.

5. JAYAVASUDEV ALGORITHM
Step 1: For the given problem choose two Machines which have largest processing time in all the processing times and chosen it for allegation.
Step 2: The given problem must reduced to 2-Machine n-Job problem.

Table 1 Numerical problem

<table>
<thead>
<tr>
<th>Machines</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>J2</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>J3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Solution by Javasudev Algorithm
We find two machines which has largest processing times.
The largest processing time is 7 in M6 and M7
Therefore we get M6, M7 for further processing
Now 7-machine 3-job problem is reduced to 2-machine 3-job problem

Table 2

<table>
<thead>
<tr>
<th></th>
<th>M6</th>
<th>M7</th>
</tr>
</thead>
<tbody>
<tr>
<td>J1</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>J2</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>J3</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Using Johnson’s algorithm we get the sequence is J2, J3, J1
Completion time of jobs on machines for the Sequence J2, J3, J1 is given in the following table
An Algorithm for Solving n-Job, m-Machine Flow Shop Scheduling problem with Objective of Minimizing Makespan

Table 3

<table>
<thead>
<tr>
<th>Machines</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
</tr>
</thead>
<tbody>
<tr>
<td>J2</td>
<td>0-1</td>
<td>1-5</td>
<td>5-10</td>
<td>10-16</td>
<td>16-19</td>
<td>19-21</td>
<td>21-26</td>
</tr>
<tr>
<td>J3</td>
<td>1-5</td>
<td>5-10</td>
<td>10-16</td>
<td>16-20</td>
<td>20-25</td>
<td>25-31</td>
<td>31-38</td>
</tr>
<tr>
<td>J1</td>
<td>5-7</td>
<td>10-13</td>
<td>16-21</td>
<td>21-26</td>
<td>26-32</td>
<td>32-39</td>
<td>39-42</td>
</tr>
</tbody>
</table>

The Makespan of Jayavasudev Algorithm is 42 time units
The best Sequence of Jayavasudev Algorithm based on Makespan is J2, J3, J1.

4. RESULT ANALYSIS
Comparison of result of above problem stated in 5.1 among Jayavasudev, Palmer, CDS, NEH, RA and Gupta

Table 4

<table>
<thead>
<tr>
<th>No. Of observations</th>
<th>Technique</th>
<th>Optimal sequence</th>
<th>Makespan</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Palmer</td>
<td>J3, J1, J2 and J1,J3,J2</td>
<td>46</td>
</tr>
<tr>
<td>02</td>
<td>CDS</td>
<td>J1, J2, J3</td>
<td>43</td>
</tr>
<tr>
<td>03</td>
<td>Gupta</td>
<td>J1, J2, J3</td>
<td>43</td>
</tr>
<tr>
<td>04</td>
<td>RA</td>
<td>J3, J1, J2</td>
<td>46</td>
</tr>
<tr>
<td>05</td>
<td>NEH</td>
<td>J2, J3, J1</td>
<td>42</td>
</tr>
<tr>
<td>06</td>
<td>Jayavasudev</td>
<td>J2, J3, J1</td>
<td>42</td>
</tr>
</tbody>
</table>

7. CONCLUSION
Using different algorithms available in the literature namely Johnson, Palmer’s, CDS, Gupta, RA and NEH Heuristic. Jayavasudev Algorithm performs better than the algorithms available in the literature and this is the best suitable algorithm for solving FSSP with the objective of minimizing the Makespan.

REFERENCES


Sahu A K. (2009). Efficient heuristics for scheduling tasks on a flow shop environment to optimize make span National Institute of Technology, Rourkela INDIA.