A DIFFERENT TECHNIQUE FOR SOLVING SEQUENCING PROBLEM

M.Geetha
Assistant Professor, Department of Mathematics,
Saradha Gangadharan College, Puducherry- 605 004, E-mail: m.geetha@yahoo.co.in

Abstract- In this paper, a new method is employed to solve the sequencing problems namely, maximin-minimax method and compared the results with the existing Johnson’s method. Also, the procedure adopted for solving the sequencing problems is easiest and involves the minimum numbers of iterations to obtain the sequence of jobs. In addition, this method is verified by means of the numerical example and shown that the solutions obtained are consistent and efficient.

Key Words- Sequencing Problem, Johnson’s Method

1.INTRODUCTION

Sequencing problem is considered to be one of the classic and important applications of operations research. The main role of the classical sequencing problem is to find the optimal sequence of the jobs on machines so as to minimize the total amount of time required to complete the process of all the jobs. The effectiveness of the sequencing problem can be measured in terms of minimized costs, maximized profits, minimized elapsed time and meeting due dates etc. In the past, because of its practical and significant use in production field many researchers have shown their interest in sequencing problems. One of the renowned work in the field of sequencing considered till date is by Johnson’s, who gave the algorithm in 1954 for production scheduling in which he had minimized the total idle time of machines and the total production times of the jobs. Later in 1967 Smith and Dudek developed a general algorithm for the solution of the n- job on m- machine sequencing problem of the flow shop when no passing is allowed. A heuristic algorithm for solving general sequencing or flow shop scheduling problem was given by Johnny and Chang (1991) for minimizing elapsed time in no-wait flow-shop scheduling. Maggu (2002) gave the technique to minimize the total idle time of machines or the total production time of the jobs on the two machines production scheduling problems. Iyer .P.Sankara (2008), in Operations Research gave the techniques to solve job sequencing problem. Also, Srikant Gupta et al.(2016) gave an algorithm for solving Job shop sequencing problem and compared the results with Johnson’s method.Anita (2017) gave a deviation technique to solve the job sequencing problem.In this paper, we proposed a procedure for solving the job scheduling problems named as maximin-minimax Method. This method is used to frame a sequence of jobs for processing the n jobs on m machines in such a way that the total elapsed time is minimized .Finally, by means of the numerical example this is verified and the results are compared with the Johnson’s method.

1. BASIC TERMS USED IN SEQUENCING :

   i. Job: The jobs or items or customers or orders are the primary stimulus for sequencing. There should be a certain number of jobs say ‘n’ to be processed or sequenced.
   
   ii. Number of Machines: A machine is characterized by a certain processing capability or facilities through which a job must pass before it is completed in the shop. It may not be necessarily a mechanical device. Even human being assigned jobs may be taken as machines. There must be certain number of machines say ‘k’ to be used for processing the jobs.
   
   iii. Processing Time: Every operation requires certain time at each of machine. If the time is certain then the determination of schedule is easy. When the processing times are uncertain then the schedule is complex.
   
   iv. Total Elapsed Time: It is the time between starting the first job and completing the last one.
   
   v. Idle time: It is the time the machine remains idle during the total elapsed time.
   
   vi. Technological order: Different jobs may have different technological order. It refers to the order in which various machines are required for completing the jobs.
   
   vii. No passing rule: It implies that passing is not allowed i.e. the same order of jobs is maintained over each machine. If each of the ‘n’ jobs is to be processed...
through ‘m’ machines in order of M₁, M₂, M₃, M₄, then this rule will mean that each job will go to machine M₁ first then to M₂ and lastly to M₄ after M₃.

1.1 Types of sequencing problems:
There can be many types of sequencing problems which are as follows:
i. Problem with ‘n’ jobs through one machine.
ii. Problem with ‘n’ jobs through two machines.
iii. Problem with ‘n’ jobs through three machines.
Here the objective is to find out the optimum sequence of the jobs to be processed and starting and finishing time of various jobs through all the machines.

1.2 Basic assumptions under sequencing problems:
Following are the basic assumptions underlying a sequencing problem:
i. No machine can process more than one job at a time.
ii. The processing times on different machines are independent of the order in which they are processed.
iii. The time involved in moving a job from one machine to another is negligibly small.
iv. Each job once started on a machine is to be performed up to completion on that machine.
v. All machines are of different types.
vi. All jobs are completely known and are ready for processing.
vii. A job is processed as soon as possible but only in the order specified.

These assumptions are considered to make the sequencing problem a simple one otherwise complicacy may arise.

2. PROCESSING OF N JOBS ON 2 MACHINES
This section presents a new method to solve the sequencing problem which is different from the preceding methods. We call it as ‘Maximin-Minimax Method’. The various steps of the procedure are as follows:

Step 1: List the jobs along with their processing times in a table as shown below:

<table>
<thead>
<tr>
<th>Jobs</th>
<th>J₁</th>
<th>J₂</th>
<th>J₃</th>
<th>...</th>
<th>Jₖ</th>
<th>...</th>
<th>Jₙ</th>
</tr>
</thead>
<tbody>
<tr>
<td>M₁</td>
<td>t₁₁</td>
<td>t₁₂</td>
<td>t₁₃</td>
<td></td>
<td>t₁ₖ</td>
<td></td>
<td>t₁ₙ</td>
</tr>
<tr>
<td>M₂</td>
<td>t₂₁</td>
<td>t₂₂</td>
<td>t₂₃</td>
<td></td>
<td>t₂ₖ</td>
<td></td>
<td>t₂ₙ</td>
</tr>
</tbody>
</table>

Step 2: Examine the rows for processing times on machines M₁ and M₂ and find the minimum processing time in each row. Similarly, select the maximum processing time for each column. Go to step 3.

Step 3: Select the maximum of all the minimum processing time of n jobs. Let the minimum of maximum processing time be occurred at kth job on ith machine. Mathematically, we can say

\[ \text{Max} \{ \text{Min} \{(t_{11}, t_{12}, ..., t_{1n}) \} \} = t_{2k} \]

Proceed to next step.

Step 4: Select the minimum of all the maximum processing time of 2 jobs. Let the minimum of maximum processing time occurred at ith machine for the kth job. Mathematically, we can say

\[ \text{Min} \{ \text{Max} \{(t_{11}, t_{12}), (t_{21}, t_{22}), ..., (t_{1n}, t_{2n}) \} \} = t_{2k} \]

Move on to next step.

Step 5: The following cases arise:
i. If maximin value equals the minimax value i.e.,

\[ \text{Max} \{ \text{Min} \{(t_{11}, t_{12}, ..., t_{1n}) \} \} = t_{2k} = \text{Min} \{ \text{Max} \{(t_{11}, t_{12}), (t_{21}, t_{22}), ..., (t_{1n}, t_{2n}) \} \} \]

Then, find the Minimum of Maximin and Minimax, in that column choose minimum and process first or last accordingly. If it corresponds to kth job, then process last. Proceed to next step.

ii. If maximin value not equals the minimax value i.e.,

\[ \text{Max} \{ \text{Min} \{(t_{11}, t_{12}, ..., t_{1n}) \} \} = t_{1k} = \text{Min} \{ \text{Max} \{(t_{11}, t_{12}), (t_{21}, t_{22}), ..., (t_{1n}, t_{2n}) \} \} \]

Then, find the Minimum of Maximin and Minimax, in that column choose minimum and process first or last accordingly. Let it be t₁k, then process kth job first. If the minimum occurs in t₁n, then process the nth job last. Go to next step.

Step 6: If more than one place has Maximin or Minimax value, then choose the Next minimum of Maximin and Minimax. Then proceed the steps from 4.

Step 7: Cross off the jobs already assigned and repeat steps 2 to 4, placing the remaining jobs first or next to last, until all the jobs have been assigned.

Step 8: Calculate idle time for machines M₁ and M₂:
i. Idle time for M₁ = Total elapsed time – (time when the last job in a sequence finishes on M₁)

\[ \text{Idle time for } M₁ = \text{Time when job in a sequence finishes on } M₁ + \sum_{j=2}^{n} \{( \text{time when the } j^{th} \text{ job in a sequence starts on } M₂) - (\text{time when the } (j-1)^{th} \text{ job in a sequence finishes on } M₂) \} \]

ii. Idle time for M₂ = Time at which first job in a sequence finishes on M₁ + \sum_{j=2}^{n} \{( \text{time when the } j^{th} \text{ job in a sequence starts on } M₂) - (\text{time when the } (j-1)^{th} \text{ job in a sequence finishes on } M₂) \}
Step 9: The total elapsed time to process all jobs through two machines is as under:
Total elapsed time = \( \text{Time when the } n^{th} \text{ job in a sequence finishes on machine } M_2 \leq \sum_{j=1}^{n} t_{2j} + \sum_{j=1}^{n} I_{2j} \) Where \( t_{2j} \) = time required for processing \( j^{th} \) job on machine \( M_2 \), \( I_{2j} \) = time for which machine \( M_2 \) remains idle after processing \( (j-1)^{th} \) job and before starting work on \( j^{th} \) job.

2.1 NUMERICAL EXAMPLE

There are 5 jobs each of which must go through the two machines A and B in the order AB. Processing time are given below in table 2.

### Table 1: processing of 5 jobs in 2 machines

<table>
<thead>
<tr>
<th>Jobs</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

Using steps from 1 to 5, leads to an optimal sequence as 1 2 6 3 5 4.

The Minimum total elapsed time is calculated for the obtained sequence in table 3 as follows:

### Table 2: Comparison of Total elapsed Time for the job sequence

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Machine A</th>
<th>Machine B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time In</td>
<td>Time Out</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>m</td>
<td>t_{ni}</td>
<td>t_{ni}</td>
</tr>
</tbody>
</table>

The Comparison between methods is given in table below

### Table 3: Comparison of Total elapsed Time and Idle time for preceding and our method

<table>
<thead>
<tr>
<th>Methods</th>
<th>Optimal Sequence</th>
<th>Idle Time for Machine A</th>
<th>Idle Time for Machine B</th>
<th>Total Elapsed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson’s Method</td>
<td>1-6-2-3-4-5</td>
<td>29 hours</td>
<td>01 hour</td>
<td>03 hours</td>
</tr>
<tr>
<td>Maximin-Minimax Method</td>
<td>1-2-6-3-5-4</td>
<td>29 hours</td>
<td>01 hour</td>
<td>03 hours</td>
</tr>
</tbody>
</table>

3. PROCESSING OF n JOBS through k MACHINES

Step 1: The processing time of n jobs (1,2,...,n) on m machines (1,2,...,m) is given table 4:

### Table 4: Processing of n jobs in m machines

<table>
<thead>
<tr>
<th>Jobs</th>
<th>Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>t_{i1}</td>
</tr>
<tr>
<td>2</td>
<td>t_{i1}</td>
</tr>
<tr>
<td>i</td>
<td>t_{i1}</td>
</tr>
<tr>
<td>m</td>
<td>t_{ni}</td>
</tr>
</tbody>
</table>

Step 2: Find min \( t_{ij} \), \( \text{min } t_{ij} \) and maximum of each of \( t_{i1},t_{i2},...,t_{ik},t_{ij} \) for all \( j = 1,2,...,n \).

Step 3: Check the following:
(i) Min. \( t_{ij} \geq \text{Max. } t_{ij} \) for \( i = 2,3,...,k-1 \) (or)
(ii) Min. \( t_{ij} \geq \text{Max. } t_{ij} \) for \( i = 2,3,...,k-1 \)

Step 4: If the inequalities of Step 3 are not satisfied, method fails. Otherwise go to next step.

Step 5: Convert the k machine problem into two machine problem by introducing two fictitious machines G and H such that \( t_{ij} = t_{ij} + t_{ij} + ... + t_{i-1,j} \) and \( t_{ij} = t_{ij} + t_{ij} + ... + t_{ij} \)

Step 6: Repeat all the Steps 3 to Step 8 of n jobs on 2 machines procedure detailed above.
3.1 NUMERICAL EXAMPLE

There are 4 jobs, each of which must go through the five machines A, B, C, D and E in the order ABCDE. Processing time is given below.

Table 5: Processing of 4 jobs in 5 machines

<table>
<thead>
<tr>
<th>Machines</th>
<th>Jobs</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

By applying Steps leads to an optimal sequence as

1 3 2 4

The Comparison between methods is given in table below

Table 6: Comparison of Total elapsed time and Idle time for preceding and our method

<table>
<thead>
<tr>
<th>Methods</th>
<th>Optimal Sequence</th>
<th>Total Elapsed time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Johnson’s Method</td>
<td>1-4-3-2</td>
<td>43 hours</td>
</tr>
<tr>
<td>Maximin-Minimax</td>
<td>1-3-2-4</td>
<td>43 hours (same Idle time for machines also)</td>
</tr>
</tbody>
</table>

4. CONCLUSION

In this Paper, a procedure is proposed for solving the job scheduling problems named as maximin-minimax Method for processing the n jobs on m machines in such a way that the total elapsed time is minimized. Finally, by means of the numerical example this is verified and the results are compared with the Johnson’s method.

REFERENCES