CHAPTER – 4
FLOWSHOP SCHEDULING WITH NO-IDLE CONSTRAINT

In flow shop sequencing problems, the target is to achieve an optimal schedule for jobs which when which carried out on the machines will optimise some of characterized criteria like total elapsed time or rental cost etc. Every job will carried out on these machines in a predetermined manner. The exploration into flow shop problems has drawn an extraordinary consideration in the most recent decades with the intend to expand the viability of modern creation. Johnson, S.M. [45] gave technique for decision the ideal arrangement for n machines flow shop problem with the core objective of reduction of the make span. Scharge, L. and Ignal, E. [43] implemented Branch and Bound strategy for acquiring a sequence for the minimization of flow time. Chandrasekharan, R. [18] has specified cases to acquire a schedule which reduce all elapsed time subject to least makespan. Anup [4], Singh, T.P. [80], [81], Bagga [6], Das and Maggu [60], Yoshida and Hitomi [92], Szwarch [86], and so forth inferred the best method for two or multistage flow shop problems considering the different restraints and criterion. Singh, T.P. and Gupta, D. [80], [81] associated probabilities with processing time and set up time in their studies. Later, Singh, T.P., Gupta, D. [39], [40], [41] concentrated on two stage general flow shop problem for minimization of rental cost under a pre-defined rental approach in which the probabilities have been associated with processing time on every machine. Narain [66], [67] concentrated on the flow shop problem with the goal being absolute rental cost. The aggregate rental cost is minimized when in-active time on every machine is zero.

Under the no-idle situation, machines work continuously without any break; that is, machines should not remain idle once they start processing the first job. The no-idle situation arises in real life world, when machines have to be hired to complete an assignment. Minimization of the total expected hiring cost of the machines would be the objective in these type of situations. The total expected hiring cost of the machines will be at a minimum when idle times on the machines are
minimum. Hence, the total expected hiring cost of the machines will be minimum when the idle times of all the machines are zero and under no-idle situation each machine is to be hired for time that is equal to the sum of processing times of all the jobs on it. We are extending the study done by Gupta, D. [41] including the concept of no-idle scheduling by associating probabilities to the processing time of the jobs. The present paper is an attempt to study the n by m general flow shop scheduling with an objective to develop a heuristic algorithm such that no machine remains idle.


4.1 PRACTICAL SITUATION

Numerous circumstances exist in our everyday schedule in plants and mechanical creation concerns and so on in which distinctive jobs are carried out on different machines in a certain manner. For example, in a foundry work shop the drawing of iron round, cutting, manufacturing, machining, warming, completing and pressing of completed articles have a settled in a certain order of processing which cannot be changed. Different functional circumstances likewise happen when individual has the jobs yet does not have its own particular machine or does not have sufficient money or does not have financial ability to purchase machine. In such cases, the machine has to be taken on lease or hired for the completion of jobs as a goal. Hiring of machines is a reasonably priced and rapid arrangement in various manufacturing which are instantly compelled by the accessibility of restricted assets because of the late worldwide financial subsidence. Hiring makes capable sparing working capital and allows up degree to new development.

4.2 NOTATIONS & DEFINITIONS

The various notations used throughout the paper are as follows:

- \( S \): Given fixed sequence of jobs
- \( M_j \): Machine \( j, j = 1, 2, 3 \ldots m \)
- \( a_{i,j} \): Processing time of \( i^{th} \) job on machine \( M_j \)
\( p_{i,j} \): Probability associated to the processing time \( a_{i,j} \)
\( A_{i,j} \): Expected processing time of \( i^{th} \) job on machine \( M_j \)
\( t_{i,j}(S) \): Completion time of \( i^{th} \) job of sequence \( S \) on machine \( M_j \)
\( I_{i,j}(S) \): Idle time of machine \( M_j \) for \( i^{th} \) job in the sequence \( S \)
\( I'_{i,j}(S) \): Idle time of machine \( M_j \) for \( i^{th} \) job in the sequence \( S \) when machine \( M_j \) starts at latest time \( L_j \).

### 4.3 DEFINITION

Completion time of \( i^{th} \) job on machine \( M_j \) is denoted by \( t_{i,j} \) and is defined as:

\[
t_{i,j} = \max (t_{i-1,j}, t_{i,j-1}) + A_{i,j} \quad \text{for} \quad j \geq 2.
\]

\[
t_{i,j} = \max (t_{i-1,j}, t_{i,j-1}) + A_{i,j}.
\]

Where \( A_{i,j} = \) expected processing time of \( i^{th} \) job on machine \( j \).

### 4.4 THEOREM

The time at which machine \( M_r \) should be taken on rent (or starts processing jobs) to have zero idle time on \( M_r \) is

\[
H_r = \max_{1 \leq k \leq n} \{ Y_k \} , \ r=2,3,-,-,-m.
\]

where

\[
Y_k = t^{(k,r-1)} - \sum_{i=r}^{k-1} p_{i,r} x a_{ir} \quad \text{for} \quad k > 1
\]

\[
Y_k = t^{(k,r-1)} - \sum_{i=1}^{k-1} A_{ir}
\]

\[
Y_1 = t^{(1,r-1)}.
\]

**Proof:** Proof is based on mathematical induction. It will be shown that if machine \( M_r \) starts processing jobs at time \( H_r \), then the idle time of \( M_r \) is zero.

For \( r=2 \),

\[
H_2 = \max_{1 \leq k \leq n} \{ Y_k \}
\]

Let \( Y_q = \max_{1 \leq k \leq n} \{ Y_k \} \)

Therefore, \( Y_q \geq Y_k \quad \text{for} \quad k=1,2,3,-,-,-n.\)

For \( k=1 \), \( Y_q \geq Y_1 \)

Which implies \( H_2 \geq t^{(1,1)} \)

Or \( t^{(1,1)} \leq H_2 \)  \quad (1)
From equation (1), if machine $M_2$ is taken on rent at time $H_2$, then it will start processing the first job without waiting. Therefore, idle time of machine $M_2$ for first job is zero when it starts processing jobs at time $H_2$.

$Y_q \geq Y_k$ for $k=2,3-\ldots,n$.

$Y_q + \sum_{i=1}^{k-1} A_{i,2} \geq Y_k + \sum_{i=1}^{k-1} A_{i,2}$

i.e $H_2 + \sum_{i=1}^{k-1} A_{i,2} \geq t_{(k,1)}' - \sum_{i=1}^{k-1} A_{i,2} + \sum_{i=1}^{k-1} A_{i,2}$

i.e $t_{(k-1,2)}' \geq t_{(k,1)}'$

Or $t_{(k,1)}' \leq t_{(k-1,2)}'$ for $k=2,3-\ldots,n$.  

-------------------------

Therefore, $I_{(k,2)}' = \max [t_{(k-1,2)}' - t_{(k-l,2)}', 0]$  

From equation (2)

$I_{(k,2)}' = 0$ for $k=2,3-\ldots,n$

Therefore, the result holds for $r=2$.

Let the result holds for $r=5$.

Now we shall also show that result is also true for $r=s+1$.

But

$H_{s+1} = \max_{1 \leq k \leq n} \{ Y_k \}$

Let $Y_t = \max_{1 \leq k \leq n} \{ Y_k \}$

Therefore, $Y_t \geq Y_1$ i.e $H_{s+1} \geq t_{(s,1)}'$

$t_{(s,1)}' \leq H_{s+1}$

-------------------------

From equation (3), if machine $M_{s+1}$ is taken on rent at time $H_{s+1}$, then it will start processing the first job without waiting. Therefore, idle time of machine $M_{s+1}$ for first job is zero when it starts processing jobs at time $H_{s+1}$.

For $k=2,3-\ldots,n$

$Y_t \geq Y_k$

Which implies $Y_t + \sum_{i=1}^{k-1} A_{i,s+1} \geq Y_k + \sum_{i=1}^{k-1} A_{i,s+1}$

$H_{s+1} + \sum_{i=1}^{k-1} A_{(i,s+1)} \geq t_{(k,s)}' - \sum_{i=1}^{k-1} A_{i,s+1} + \sum_{i=1}^{k-1} A_{i,s+1}$

which implies

$t_{(k-1,s+1)}' \geq t_{(k,s)}'$

$t_{(k,s)}' \leq t_{(k-1,s+1)}'$ for $k=2,3-\ldots,n$  

-------------------------

Therefore, $I_{(k,s+1)}' = \max [t_{(k,s)}' - t_{(k-1,s+1)}, 0]$  

From equation (4),

$I_{(k,s+1)}' = 0$ for $k=2,3-\ldots,n$
Therefore, the result is true for $r = s+1$ also.

### 4.5 ASSUMPTIONS

1. No machine processes more than one job at a time.
2. Pre-emption of jobs is not allowed.
3. Machines never breakdown during the scheduling process.
4. Each job is processed through each of the machines once and only once.
5. All the jobs and the machines are available at the beginning of the processing.
6. Jobs are independent of each other.

### 4.6 ALGORITHM

The algorithm given in this chapter gives the technique to find the times at which machines are hired so that then-active time gets to be zero, minimizing total expected rental cost under the given policy.

**Step 1:** Calculate the expected processing time $A_{ij} = a_{ij} \times p_{ij}, \forall i, j = 1, 2, 3, \ldots, m$

**Step 2:** For the fixed given sequence $S$, prepare the In-Out table for the machine pair $(M_j, M_{j+1})$, $j = 1, 2, 3, \ldots, m-1$ as two machine flowshop sequence problem.

**Step 3:** Compute $K_{j+1}$ for the machine pair $(M_j, M_{j+1})$ by the formula

$$K_{j+1} = \sum_{i=1}^{n} A_{ij} + 1, 2, 3, \ldots, m-1$$

**Step 4:** Calculate latest time $L_j$ for the machine pair $(M_j, M_{j+1})$ by the formula

$$H_j = H_{j-1} + K_j \quad j = 3, 4, 5, \ldots, m-1$$

$$H_2 = K_2 \text{ and } H_1 = 0$$

**Step 5:** Prepare the In-Out table for the machines with latest times $L_j$, the idle time is zero for all machines.

### 4.7 see adendix

### 4.8 FUZZY LOGIC ENGINE

The calculations has been done using c++ program by using different values of processing times and probabilities. Based on the results, fuzzy logic engine has
been formed in matlab fuzzy logic toolbox. The fuzzy logic rule base is based on the results generated from the c++ program. The membership function values and fuzzy rules are formed on the base of data generated from c++ program. The figure 4.8.1 below shows the fuzzy logic system which consist of two inputs: processing time and probability and three output variables namely H2, H3 and H4.

Figure 4.8.1

The Figure 4.8.2 shown below illustrates the membership functions for input processing time.
Figure 4.8.2

The Figure 4.8.3 shown below illustrates the membership functions for input Probability.

Figure 4.8.3

The Figure 4.8.4 shown below illustrates the membership functions for output H3.
The following Figure 4.8.5 shown below illustrates firing of the rule base:

**Surfaces:**

The control surface for output H2 is shown in Figure 4.8.6 below:
Figure 4.8.6

The control surface for output H3 is shown in Figure 4.8.7 below:

Figure 4.8.7

The control surface for output H4 is shown in Figure 4.8.8 below:
4.9 NUMERICAL ILLUSTRATION

Consider a 5-job, 4-machine sequencing problem whose processing times with their corresponding probabilities are given in the tableau 4.9.1.

<table>
<thead>
<tr>
<th>Jobs</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>$M_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>$a_{i,1}$</td>
<td>$p_{i,1}$</td>
<td>$a_{i,2}$</td>
<td>$p_{i,2}$</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>0.2</td>
<td>50</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>0.2</td>
<td>40</td>
<td>0.1</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>0.1</td>
<td>30</td>
<td>0.3</td>
</tr>
<tr>
<td>4</td>
<td>30</td>
<td>0.3</td>
<td>20</td>
<td>0.4</td>
</tr>
<tr>
<td>5</td>
<td>30</td>
<td>0.2</td>
<td>80</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Tableau 4.9.1

Here our objective is to obtain the latest time of the machines so that no machine remains idle.

Solution:
As per step 1 find expected processing times are given in table tableau 4.9.2.

<table>
<thead>
<tr>
<th>Jobs</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>$M_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{i1}$</td>
<td>$A_{i2}$</td>
<td>$A_{i3}$</td>
<td>$A_{i4}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>9</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>8</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Tableau 4.9.2

As per step 2 In-Out table for pair $(M_1, M_2)$ as shown in figure 4.9.3

<table>
<thead>
<tr>
<th>Jobs</th>
<th>$M_1$</th>
<th>$M_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{i1}$</td>
<td>$A_{i2}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0-4</td>
<td>4-9</td>
</tr>
<tr>
<td>2</td>
<td>4-6</td>
<td>9-13</td>
</tr>
<tr>
<td>3</td>
<td>6-9</td>
<td>13-22</td>
</tr>
<tr>
<td>4</td>
<td>9-18</td>
<td>33-41</td>
</tr>
<tr>
<td>5</td>
<td>18-24</td>
<td>41-49</td>
</tr>
</tbody>
</table>

Tableau 4.9.3

As per step 3 and step 4

$K_2 = t_{n,2} - \sum_{i=1}^{s} A_{i,2} = 49-34=15$

$H_2 = K_2 = 15$ units also we have $H_1=0$

As per step 2 In-Out table for pair $(M_2, M_3)$ as shown in tableau 4.9.4

<table>
<thead>
<tr>
<th>Jobs</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{i2}$</td>
<td>$A_{i3}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0-5</td>
<td>5-7</td>
</tr>
<tr>
<td>2</td>
<td>5-9</td>
<td>9-13</td>
</tr>
<tr>
<td>3</td>
<td>9-18</td>
<td>18-24</td>
</tr>
<tr>
<td>4</td>
<td>18-26</td>
<td>26-34</td>
</tr>
<tr>
<td>5</td>
<td>26-34</td>
<td>34-35</td>
</tr>
</tbody>
</table>

Tableau 4.9.4

As per step 3 and 4
\[ K_3 = t_{5,3} - \sum_{i=1}^{5} A_{i,3} = 35 - 21 = 14 \]

\[ H_3 = H_2 + K_3 = 15 + 14 = 29 \]

As per step 2 In-Out table for pair \((M_3, M_4)\) as shown in tableau 4.9.5

<table>
<thead>
<tr>
<th>Jobs</th>
<th>(M_3)</th>
<th>(M_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(A_{i,3})</td>
<td>(A_{i,4})</td>
</tr>
<tr>
<td>1</td>
<td>0-2</td>
<td>2-14</td>
</tr>
<tr>
<td>2</td>
<td>2-6</td>
<td>14-22</td>
</tr>
<tr>
<td>3</td>
<td>6-12</td>
<td>22-25</td>
</tr>
<tr>
<td>4</td>
<td>12-20</td>
<td>25-27</td>
</tr>
<tr>
<td>5</td>
<td>20-21</td>
<td>27-29</td>
</tr>
</tbody>
</table>

**Tableau 4.9.5**

As per step 3 and 4

\[ K_4 = t_{5,4} - \sum_{i=1}^{5} A_{i,4} = 2 \]

\[ H_4 = H_3 + K_4 = 29 + 2 = 31 \]

As per step 5

The In-out table for the machines with idle time zero is as shown in tableau 4.9.6

<table>
<thead>
<tr>
<th>Jobs</th>
<th>(M_1)</th>
<th>(M_2)</th>
<th>(M_3)</th>
<th>(M_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(A_{i,1})</td>
<td>(A_{i,2})</td>
<td>(A_{i,3})</td>
<td>(A_{i,4})</td>
</tr>
<tr>
<td>1</td>
<td>0-4</td>
<td>15-20</td>
<td>29-31</td>
<td>31-43</td>
</tr>
<tr>
<td>2</td>
<td>4-6</td>
<td>20-24</td>
<td>31-35</td>
<td>43-51</td>
</tr>
<tr>
<td>3</td>
<td>6-9</td>
<td>24-33</td>
<td>35-41</td>
<td>51-54</td>
</tr>
<tr>
<td>4</td>
<td>9-18</td>
<td>33-41</td>
<td>41-49</td>
<td>54-56</td>
</tr>
<tr>
<td>5</td>
<td>18-24</td>
<td>41-49</td>
<td>49-50</td>
<td>56-58</td>
</tr>
</tbody>
</table>

**Tableau 4.9.6**

Hence we conclude that no machine remains idle.
Conclusion

The proposed algorithm provides the latest time at which processing of jobs on 2\textsuperscript{nd}, 3\textsuperscript{rd}, and so on m\textsuperscript{th} machine must be started such that these machines work consistently with no break until the keep going Jobs are not finished on them. The first machine has no idle time and hence, works continuously, i.e. the proposed algorithm helps the decision makers in determining the best latest time at which machines should be hired for a given set of jobs so as to minimize the total expected rental cost with no-idle constraint. The can be extended by introducing concepts of independent setup time, transportation time, job block criteria, non-availability constraints of machines for a certain interval of time.