HEURISTIC APPROACH FOR BICRITERIA IN CONSTRAINED N X 2 FLOW SHOP SCHEDULING PROBLEM

Deepak Gupta *

Shefali Aggarwal **

Sameer Sharma ***

Seema ***

ABSTRACT

A bicriteria in n- jobs, 2- machines flow shop scheduling to minimize the rental cost of the machines under a specific rental policy in which the processing time and independent setup time each associated with probabilities including break down interval and job block criteria is considered. Further the jobs are attached with weights to indicate their relative importance. In this paper, a new heuristic algorithm has been developed for minimizing the makespan as well as the rental cost of machines which is simple and straight forward. A computer programme followed by a numerical illustration is given to substantiate the algorithm.

Keywords: *Flowshop Scheduling, Heuristic, Processing Time, Setup Time, Job Block, Breakdown Interval, Weights of jobs.*

* Professor, Department of Mathematics, Maharishi Markandeshwar University, Mullana, Haryana, India.

** Research Scholar, Department of Mathematics, Maharishi Markandeshwar University, Mullana, Haryana.

*** Assistant Professor, Department of Mathematics, D.A.V. College, Jalandhar, Punjab, India.

1. INTRODUCTION

Scheduling problems have been extensively investigated by many researchers. Its aim is to determine the sequence for processing jobs on a given set of machines. Its importance and relevance to industry has promoted researchers to study it from different perspectives. Flowshop is the classical and most studied manufacturing environment in scheduling literature. In a general flowshop the jobs (operations) are not pre - emissionable, the jobs are available for processing at time zero and set up times are sequence independent. Makespan and total flow time are commonly used performance measure in flowshop scheduling literature. The development of various heuristic methods that consider a single measure of performance, viz, makespan has been observed. However, the desirability of a schedule being evaluated by more than one performance measure is often cited in the literature. Apart from makespan objective, other significant objectives in flowshop scheduling problem is minimisation of total flow time of all jobs and hence the concept of bicriteria become significant. Further, the classical scheduling literature commonly assumes that the machines are never unavailable during the process. This assumption might be justified in some cases but it does not apply if certain maintenance requirements, break– downs or other constraints that causes machine not to be available for processing have to be considered. The temporal lack of machine availability is known as 'break – down'. The scheduling problem practically depends upon the important factors namely, Job block criteria which is due to priority of one job over the another, Weightage of job which is due to relative importance of a job over another job(s) and Setup times which include obtaining tools, positioning word – in – process material, return tooling, cleaning up, setting the required jigs and fixtures, adjusting tools and inspecting material. These concepts are separately studied by various researchers such as Johnson (1954), Belman (1956), Smith (1956), Baker (1974), Maggu & Das (1977), Singh T.P. (1985), Adiri (1989), Akturk and Gorgulu (1999), Chandramouli (2005), Belwal and Mittal (2008), Pandian & Rajendran (2010), Khodadadi (2011), Gupta & Sharma (2011).

Gupta & Sharma (2011) studied bicriteria in n × 2 flow shop scheduling under specified rental policy, processing time and setup time associated with probabilities including job block. This paper is an attempt to extend the study made by Gupta & Sharma (2011) by introducing the concept of break – down interval and weightage in jobs. Thus the problem discussed here is wider and practically more applicable and will have significant results in process industry. We have obtained an algorithm which gives minimum possible rental cost while minimising total elapsed time simultaneously.

2. PRACTICAL SITUATION

Many applied and experimental situations exist in our day-to-day working in factories and industrial production concerns etc. The practical situation may be taken in a paper mill, sugar factory and oil refinery etc. where various qualities of paper, sugar and oil are produced with relative importance i.e. weight in jobs, hence weightage of jobs is significant. Further many practical situations occur in real life when one has got the assignments but does not have one's own machine or does not have enough money or does not want to take risk of investing huge amount of money to purchase machine. Under such circumstances, the machine has to be taken on rent in order to complete the assignments. In his starting career, we find a medical practitioner does not buy expensive machines say X-ray machine, the Ultra Sound Machine, Rotating Triple Head Single Positron Emission Computed Tomography Scanner, Patient Monitoring Equipment, and Laboratory Equipment etc., but instead takes on rent. Rental of medical equipment is an affordable and quick solution for hospitals, nursing homes, physicians, which are presently constrained by the availability of limited funds due to the recent global economic recession. Renting enables saving working capital, gives option for having the equipment, and allows upgradation to new technology. The idea of job block has practical significance to create a balance between a cost of providing priority in service to the customer and cost of giving service with non priority. Another event which is mostly considered in the models is the break-down of machines. There may also be delays due to material, changes in release and tail dates, tools unavailability, failure of electric current, the shift pattern of the facility and fluctuations in processing times. Hence the criterion of break-down interval becomes significant.

3. NOTATIONS

S : Sequence of jobs 1, 2, 3,…., n

S_k : Sequence obtained by applying Johnson's procedure, k = 1, 2 , 3, -----

M_j : Machine j, j= 1,2

M : Minimum makespan

a_{ij} : Processing time of i^{th} job on machine M_j

p_{ij} : Probability associated to the processing time a_{ij}

s_{ij} : Set up time of i^{th} job on machine M_j

q_{ij} : Probability associated to the set up time s_{ij}

A_{ij} : Expected processing time of i^{th} job on machine M_j

S_{ij} : Expected set up time of i^{th} job on machine M_j

A'_{ij} : Expected flow time of i^{th} job on machine M_j

w_i : Weight of i^{th} job

A''_{ij} : Weighted flow time of i^{th} job on machine M_j

β : Equivalent job for job – block

$A^1_{\alpha i}$: Expected processing time of i^{th} job after break-down effect on machine M_1

$A^2_{\alpha i}$: Expected processing time of i^{th} job after break-down effect on machine M_2

L : Length of the break-down interval

$L_j(S_k)$: The latest time when machine M_j is taken on rent for sequence S_k

$t_{ij}(S_k)$: Completion time of i^{th} job of sequence S_k on machine M_j

$t'_{ij}(S_k)$: Completion time of i^{th} job of sequence S_k on machine M_j when machine M_j start processing jobs at time $L_j(S_k)$

$I_{ij}(S_k)$: Idle time of machine M_j for job i in the sequence S_k

$U_j(S_k)$: Utilization time for which machine M_j is required, when M_j starts processing jobs at time $L_j(S_k)$

$R(S_k)$: Total rental cost for the sequence S_k of all machine

C_i : Rental cost of i^{th} machine

3.1 Definition

Completion time of i^{th} job on machine M_j is denoted by t_{ij} and is defined as :

$$t_{ij} = \max(t_{i-1,j} + s_{(i-1)j} \times q_{(i-1)j}\ ,\ t_{i,j-1}) + a_{ij} \times p_{ij} \quad \text{for } j \geq 2.$$

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

where $A_{i,,j}$= Expected processing time of i^{th} job on j^{th} machine

$S_{i,j}$= Expected setup time of i^{th} job on j^{th} machine.

3.2 Definition

Completion time of i^{th} job on machine M_j starts processing jobs at time L_j is denoted by t'_{ij} and is defined as

$$t'_{i,j} = L_j + \sum_{k=1}^{i} A_{k,j} + \sum_{k=1}^{i-1} S_{k,j} = \sum_{k=1}^{i} I_{k,j} + \sum_{k=1}^{i} A_{k,j} + \sum_{k=1}^{i-1} S_{k,j}$$

Also $t'_{i,j} = \max(t_{i,j-1}, t'_{i-1,j} + S_{i-1,j}) + A_{i,j}$.

4. RENTAL POLICY

The machines will be taken on rent as and when they are required and are returned as and when they are no longer required. .i.e. the first machine will be taken on rent in the starting of the processing the jobs, 2nd machine will be taken on rent at time when 1st job is completed on 1st machine.

5. PROBLEM FORMULATION

Let some job i (i = 1,2,……..,n) are to be processed on two machines M_j (j = 1,2) under the specified rental policy P. Let a_{ij} be the processing time of i^{th} job on j^{th} machine with probabilities p_{ij} and s_{ij} be the setup time of i^{th} job on j^{th} machine with probabilities q_{ij}. Let w_i be the weight of i^{th} job. Our aim is to find the sequence $\{S_k\}$ of the jobs which minimize the rental cost of the machines while minimizing total elapsed time.

The mathematical model of the problem in matrix form can be stated as:

Jobs	Machine M_1				Machine M_2				Weight of job
i	a_{i1}	p_{i1}	s_{i1}	q_{i1}	a_{i2}	p_{i2}	s_{i2}	q_{i2}	w_i
1	a_{11}	p_{11}	s_{11}	q_{11}	a_{12}	p_{12}	s_{12}	q_{12}	w_1
2	a_{21}	p_{21}	s_{21}	q_{21}	a_{22}	p_{22}	s_{22}	q_{22}	w_2
3	a_{31}	p_{31}	s_{31}	q_{31}	a_{32}	p_{32}	s_{32}	q_{32}	w_3
4	a_{41}	p_{41}	s_{41}	q_{41}	a_{42}	p_{42}	s_{42}	q_{42}	w_4
5	a_{51}	p_{51}	s_{51}	q_{51}	a_{52}	p_{52}	s_{52}	q_{52}	w_5

Table 1

Mathematically, the problem is stated as

Minimize $U_j(S_k)$ and

Minimize $R(S_k) = t_{n1}(S_k)\times C_1 + U_j(S_k)\times C_2$

Subject to constraint: Rental Policy (P)

Our objective is to minimize rental cost of machines while minimizing total elapsed time.

6. THEOREM

The processing of jobs on M_2 at time $L_2 = \sum_{i=1}^{n} I_{i,2}$ keeps $t_{n,2}$ unaltered:

Proof. Let $t'_{i,2}$ be the completion time of i^{th} job on machine M_2 when M_2 starts processing of jobs at L_2. We shall prove the theorem with the help of mathematical induction.

Let P(n) : $t'_{n,2} = t_{n,2}$

Basic step: For n = 1, j =2;

$$t'_{1,2} = L_2 + \sum_{k=1}^{1} A_{k,2} + \sum_{k=1}^{1-1} S_{k,2} = \sum_{k=1}^{1} I_{k,2} + \sum_{k=1}^{1} A_{k,2} + \sum_{k=1}^{1-1} S_{k,2}$$

$$= \sum_{k=1}^{1} I_{k,2} + A_{1,2} = I_{1,2} + A_{1,2} = A_{1,1} + A_{1,2} = t_{1,2} ,$$

$\therefore$ P(1) is true.

Induction Step: Let P(m) be true, i.e., $t'_{m,2} =$ $t_{m,2}$

Now we shall show that P(m+1) is also true, i.e., $t'_{m+1,2} = t_{m+1,2}$

Since $t'_{m+1,2} = \max(t_{m+1,1}, t'_{m,2} + S_{m,2}) + A_{m+1,2}$

$$= \max\left(t_{m+1,1}, t_{m,2} + S_{m,2}\right) + A_{m+1,2} \qquad \text{(By Assumption)}$$

$$= t_{m+1,2}$$

Therefore, *P(m+1)* is true whenever *P(m)* is true.

Hence by Principle of Mathematical Induction P(n) is true for all n i.e. $t'_{n,2} = t_{n,2}$ for all n.

Remark: If M_2 starts processing the job at $L_2 = t_{n,2} - \sum_{i=1}^{n} A_{i,2} - \sum_{i=1}^{n-1} S_{i,2}$, then total time elapsed $t_{n,2}$ is not altered and M_2 is engaged for minimum time. If M_2 starts processing the jobs at time L_2 then it can be easily shown that. $t_{n,2} = L_2 + \sum_{i=1}^{n} A_{i,2} + \sum_{i=1}^{n-1} S_{i,2}$.

7. ALGORITHM

Step 1: Calculate the expected processing times and expected set up times as follows

$A_{ij} = a_{ij} \times p_{ij}$ and $S_{ij} = s_{ij} \times q_{ij} \quad \forall i, j$

Step 2: Calculate the expected flow time for the two machines M_1 and M_2 as follows

$A'_{i1} = A_{i1} - S_{i2}$ and $A'_{i2} = A_{i2} - S_{i1}\ \forall i$

Step 3: If min (A'_{i1}, A'_{i2})= A'_{i1} , then $G_i = A'_{i1} + w_i$, $H_i = A'_{i2}$ and

If min (A'_{i1}, A'_{i2})= A'_{i2} , then $H_i = A'_{i2} + w_i$, $G_i = A'_{i2}$.

Step 4: Find the weighted flow time for two machine M_1 and M_2 as follows

$A''_{i1} = G_i / w_i$ and $A''_{i2} = H_i / w_i \ \forall i$

Step 5: Take equivalent job $\beta(k,m)$ and calculate the processing time $A''_{\beta 1}$ and $A''_{\beta 2}$ on the guide lines of Maggu and Das [1977] as follows

$$A''_{\beta 1} = A''_{k1} + A''_{m1} - \min(A''_{m1}, A''_{k2})$$

$$A''_{\beta 2} = A''_{k2} + A''_{m2} - \min(A''_{m1}, A''_{k2})$$

Step 6: Define a new reduced problem with the processing times A''_{i1} and A''_{i2} as defined in step 4 and jobs (k,m) are replaced by single equivalent job β with processing time $A''_{\beta 1}$ and $A''_{\beta 2}$ as defined in step 5.

Step 7: Using Johnson's technique [1] obtain all the sequences S_k having minimum elapsed time. Let these be S_1, S_2, ----------.

Step 8: Prepare a flow time tables for the sequences obtained in step 7 and read the effect of break-down interval (a ,b) on different jobs on the lines of *Singh T.P.* [1985].

Step 9: Form a reduced problem with processing times $A^1_{\alpha i}$ & $A^2_{\alpha i}$

If the break-down interval (a, b) has effect on job *i* then

$A^1_{\alpha i} = A_{i1} + L$ and

$A^2_{\alpha i} = A_{i2} + L$; Where L = b – a, the length of break-down interval

If the break-down interval (a, b) has no effect on ith job then

$A^1_{\alpha i} = A_{i1}$

$A^2_{\alpha i} = A_{i2}$

Step 10: Now repeat the procedure to get the optimal sequence S'_k

Step 11 : Compute total elapsed time $t_{n,2}(S'_k)$ *k = 1,2,3,*----, by preparing in-out tables for S'_k.

Step 12 : Compute $L_2(S'_k)$ for each sequence S'_k as follows

$$L_2(S'_k) = t_{n,2}(S'_k) - \sum_{i=1}^{n} A_{i,2}(S'_k) - \sum_{i=1}^{n-1} S_{i,2}(S'_k)$$

Step 13 : Find utilization time of 2nd machine for each sequence S'_k as

$$U_2(S'_k) = t_{n2}(S'_k) - L_2(S'_k).$$

Step 14 : Find minimum of $\{(U_2(S'_k)\}$; $k = 1,2,3,\ldots$

Let it for sequence S'_p. Then S'_p is the optimal sequence and minimum rental cost for the sequence S'_p is

$$R(S'_p) = t_{n,1}(S'_p) \times C_1 + U_2(S'_p) \times C_2.$$

8. PROGRAMME

```
#include<iostream.h>
#include<stdio.h>
```

```
#include<conio.h>
#include<process.h>

int n,j;
float
a1[16],b1[16],a11[16],b11[16],g[16],h[16],g1[16],h1[16],g12[16],h12[16],sa1[16],sb1[16];
float macha[16],machb[16],macha1[16],machb1[16],cost_a,cost_b,cost;
int f=1;int group[16];//variables to store two job blocks
float minval,minv,maxv;
int bd1,bd2;// Breakdown interval
float gbeta=0.0,hbeta=0.0;float gbeta1=0.0,hbeta1=0.0;

void main()
{
clrscr();
int a[16],b[16],sa[16],sb[16],j[16],w[16];float p[16],q[16],u[16],v[16];float maxv;
cout<<"How many Jobs (<=15) : ";cin>>n;

if(n<1 || n>15)
        {
cout<<endl<<"Wrong input, No. of jobs should be less than 15..\n Exitting";
        getch();exit(0);
        }
for(int i=1;i<=n;i++)
        {
        j[i]=i;
cout<<"\nEnter the processing time and its probability, Setup time and its probability of
"<<i<<" job for machine A : ";cin>>a[i]>>p[i]>>sa[i]>>u[i];
cout<<"\nEnter the processing time and its probability, Setup time and its probability of
"<<i<<" job for machine B : ";cin>>b[i]>>q[i]>>sb[i]>>v[i];
cout<<"\nEnter the weightage of "<<i<<"job:";cin>>w[i];
//Calculate the expected processing times of the jobs for the machines:
        a1[i] = a[i]*p[i];b1[i] = b[i]*q[i];
//Calculate the expected setup times of the jobs for the machines:
```

```
        sa1[i] = sa[i]*u[i];sb1[i] = sb[i]*v[i];
        }
cout<<"\nEnter the two breakdown interval:";cin>>bd1>>bd2;
cout<<"\nEnter the rental cost of Machine A:";cin>>cost_a;
cout<<"\nEnter the rental cost of Machine B:";cin>>cost_b;
cout<<endl<<"Expected processing time of machine A and B: \n";
for(i=1;i<=n;i++)
        {
cout<<j[i]<<"\t"<<a1[i]<<"\t"<<b1[i]<<"\t";cout<<sa1[i]<<"\t"<<sb1[i]<<"\t"<<w[i];
        cout<<endl;
        }
//Calculate the final expected processing time for machines
cout<<endl<<"Final expected processing time of machin A and B:\n";
for(i=1;i<=n;i++)
        {
                g1[i]=a1[i]-sb1[i];h1[i]=b1[i]-sa1[i];
        }
for(i=1;i<=n;i++)
        {
        if(g1[i]<h1[i])
        {
        g12[i]=g1[i]+w[i];h12[i]=h1[i];
        }
else
        {
        h12[i]=h1[i]+w[i];g12[i]=g1[i];
        }}
for(i=1;i<=n;i++)
        {
                g[i]=g12[i]/w[i];h[i]=h12[i]/w[i];
        }
for(i=1;i<=n;i++)
        {
        cout<<"\n\n"<<j[i]<<"\t"<<g[i]<<"\t"<<h[i]<<"\t"<<w[i];cout<<endl;
```

```
        }
        cout<<"\nEnter the two job blocks(two numbers from 1 to "<<n<<"):";
        cin>>group[0]>>group[1];
  //calculate G_Beta and H_Beta
if(g[group[1]]<h[group[0]])
        {
        minv=g[group[1]];
        }
else
        {
        minv=h[group[0]];
        }
gbeta=g[group[0]]+g[group[1]]-minv;hbeta=h[group[0]]+h[group[1]]-minv;
cout<<endl<<endl<<"G_Beta="<<gbeta;cout<<endl<<"H_Beta="<<hbeta;
int j1[16];float g11[16],h11[16];
for(i=1;i<=n;i++)
        {
        if(j[i]==group[0]||j[i]==group[1])
        {f--;}
else
        {j1[f]=j[i];}
        f++;
        }
        j1[n-1]=17;
for(i=1;i<=n-2;i++)
        {
        g11[i]=g[j1[i]];h11[i]=h[j1[i]];
        }
        g11[n-1]=gbeta;h11[n-1]=hbeta;
        cout<<endl<<endl<<"displaying original scheduling table"<<endl;
for(i=1;i<=n-1;i++)
        {
        cout<<j1[i]<<"\t"<<g11[i]<<"\t"<<h11[i]<<endl;
        }
```

```
        float mingh[16];char ch[16];
for(i=1;i<=n-1;i++)
        {
        if(g11[i]<h11[i])
        {mingh[i]=g11[i];ch[i]='g';}
else
        {mingh[i]=h11[i];ch[i]='h';}}
        for(i=1;i<=n-1;i++)
        {
        for(int j=1;j<=n-1;j++)
        if(mingh[i]<mingh[j])
                {
float temp=mingh[i]; int temp1=j1[i]; char d=ch[i];mingh[i]=mingh[j]; j1[i]=j1[j];
ch[i]=ch[j];mingh[j]=temp; j1[j]=temp1; ch[j]=d;
                } }
// calculate beta scheduling
float sbeta[16];int t=1,s=0;
for(i=1;i<=n-1;i++)
        {
        if(ch[i]=='h')
        { sbeta[(n-s-1)]=j1[i];s++;}
else if(ch[i]=='g')
        {sbeta[t]=j1[i];t++;}}
int arr1[16], m=1;cout<<endl<<endl<<"Job Scheduling:"<<"\t";
for(i=1;i<=n-1;i++)
        {if(sbeta[i]==17)
        {arr1[m]=group[0];arr1[m+1]=group[1];
        cout<<group[0]<<" " <<group[1]<<" ";m=m+2;continue;}
else
        {cout<<sbeta[i]<<" ";arr1[m]=sbeta[i];
        m++;}}
//calculating total computation sequence
  float time=0.0,macha11[16],maxv1[16];
        macha[1]=time+a1[arr1[1]];
```

```
        for(i=2;i<=n;i++)
        {macha11[i]=macha[i-1]+sa1[arr1[i-1]];macha[i]=macha11[i]+a1[arr1[i]];}
        machb[1]=macha[1]+b1[arr1[1]];
for(i=2;i<=n;i++)
        {if((machb[i-1]+sb1[arr1[i-1]])>(macha[i]))
        maxv1[i]=machb[i-1]+sb1[arr1[i-1]];
else
        maxv1[i]=macha[i];machb[i]=maxv1[i]+b1[arr1[i]];}
cout<<endl<<endl<<"In-Out Table is:"<<endl<<endl;
cout<<"Jobs"<<"\t"<<"Machine M1"<<"\t"<<"\t"<<"Machine M2"<<endl;
cout<<arr1[1]<<"\t"<<time<<"--"<<macha[1]<<" \t"<<"\t"<<macha[1]<<"--
"<<machb[1]<<endl;
if((time<=bd1 && macha[1]<=bd1)||(time>=bd2 && macha[1]>=bd2))
        {a1[arr1[1]]=a1[arr1[1]];}
else
        {a1[arr1[1]]+=(bd2-bd1);}
if((macha[1])<=bd1 && machb[1]<=bd1||(macha[1])>=bd2 && machb[1]>=bd2)
        {b1[arr1[1]]=b1[arr1[1]];}
else
        {b1[arr1[1]]+=(bd2-bd1);}
for(i=2;i<=n;i++)
        {
cout<<arr1[i]<<"\t"<<macha11[i]<<"--"<<macha[i]<<" "<<"\t"<<maxv1[i]<<"--
"<<machb[i]<<endl;
if(macha11[i]<=bd1 && macha[i]<=bd1 || macha11[i]>=bd2 && macha[i]>=bd2)
        {a1[arr1[i]]=a1[arr1[i]];}
else
        {a1[arr1[i]]+=(bd2-bd1);}
if(maxv1[i]<=bd1 && machb[i]<=bd1 || maxv1[i]>=bd2 && machb[i]>=bd2)
        {b1[arr1[i]]=b1[arr1[i]];}
else
        {b1[arr1[i]]+=(bd2-bd1);}}
int j11[16];
```

```
for(i=1;i<=n;i++)
{       j11[i]=i;
        a11[arr1[i]]=a1[arr1[i]];
        b11[arr1[i]]=b1[arr1[i]];
}
cout<<endl<<"Modified Processing time after breakdown for the machines is:\n";
cout<<"Jobs"<<"\t"<<"Machine M1"<<"\t"<<"\t"<<"Machine M2"
<<"\t"<<"\t"<<"Weightage"<<endl;
for(i=1;i<=n;i++)
{cout<<endl;cout<<j11[i]<<"\t"<<a11[i]<<"\t"<<b11[i]<<"\t"<<w[i];cout<<endl;}
float maxa12,minb12,minb22,maxc12;float g12[16],h12[16];
//Function for two ficticious machine G and H
for(i=1;i<=n;i++)
        { g12[i]=a11[i]-sb1[i]; h12[i]=b11[i]-sa1[i];}
cout<<endl<<"Expected processing time for two fictious machines G and H: \n";
  for(i=1;i<=n;i++)
{cout<<endl;  cout<<j11[i]<<"\t"<<g12[i]<<"\t"<<h12[i]<<"\t"<<w[i];  cout<<endl;}
//To find minimum of G & H
float minv1;
for (i=1;i<=n;i++)
        if(g12[i]<=h12[i])
        {g11[i]=g12[i]+w[i];h11[i]=h12[i];}
else
        {g11[i]=g12[i];h11[i]=h12[i]+w[i];}
float g21[16],h21[16];
for(i=1;i<=n;i++)
        {g21[i]=g11[i]/w[i];h21[i]=h11[i]/w[i];}
cout<<endl<<endl<<"displaying original scheduling table"<<endl;
for(i=1;i<=n;i++)
{cout<<j11[i]<<"\t"<<g21[i]<<"\t"<<h21[i]<<endl;}
//calculate G_Beta and H_Beta
if(g21[group[1]]<h21[group[0]])
        {minv1=g21[group[1]];}
else
```

```
          {minv1=h21[group[0]];}
gbeta1=g21[group[0]]+g21[group[1]]-minv1;
hbeta1=h21[group[0]]+h21[group[1]]-minv1;
cout<<endl<<endl<<"G_Beta1="<<gbeta1;cout<<endl<<"H_Beta1="<<hbeta1;
int j2[16];float g14[16],h14[16];int f1=1;
for(i=1;i<=n;i++)
          {if(j11[i]==group[0]||j11[i]==group[1])
          {f1--;}
else
          {j2[f1]=j11[i];} f1++;}
          j2[n-1]=17;
for(i=1;i<=n-2;i++)
          {g14[i]=g21[j2[i]];h14[i]=h21[j2[i]];}
g14[n-1]=gbeta1;h14[n-1]=hbeta1;
cout<<endl<<endl<<"displaying original scheduling table"<<endl;
for(i=1;i<=n-1;i++)
{cout<<j2[i]<<"\t"<<g14[i]<<"\t"<<h14[i]<<endl;}
float mingh1[16];
          char ch1[16];
 for(i=1;i<=n-1;i++)
   {if(g14[i]<h14[i])
                    {mingh1[i]=g14[i];ch1[i]='g'; }
          else
                    {mingh1[i]=h14[i];ch1[i]='h';}
          }
                    for(i=1;i<=n-1;i++)
          {for(int j=1;j<=n-1;j++)
                    if(mingh1[i]<mingh1[j])
                    {
float temp=mingh1[i]; int temp1=j2[i]; char d=ch1[i];mingh1[i]=mingh1[j]; j2[i]=j2[j];
ch1[i]=ch1[j];mingh1[j]=temp; j2[j]=temp1; ch1[j]=d;
                    }
             }
// calculate beta scheduling
```

```
float sbeta1[16];int t2=1,s21=0;
for(i=1;i<=n-1;i++)
        {if(ch1[i]=='h')
        { sbeta1[(n-s21-1)]=j2[i]; s21++;}
else    if(ch1[i]=='g')
        {sbeta1[t2]=j2[i];t2++;}}
int arr2[16], m1=1;
cout<<endl<<endl<<"Job Scheduling:"<<"\t";
for(i=1;i<=n-1;i++)
        {if(sbeta1[i]==17)
        { arr2[m1]=group[0];arr2[m1+1]=group[1];
        cout<<group[0]<<" "<<group[1]<<" ";m1=m1+2;continue;}
else    {cout<<sbeta1[i]<<" ";arr2[m1]=sbeta1[i];m1++;}}
//calculating total computation sequence;
float time1=0.0,macha12[16] ;
float maxv11[16],maxv21[16];
        macha1[1]=time1+a11[arr2[1]];
for(i=2;i<=n;i++)
{macha12[i]=macha1[i-1]+sa1[arr2[i-1]];macha1[i]=macha12[i]+a11[arr2[i]];}
machb1[1]=macha1[1]+b11[arr2[1]];
for(i=2;i<=n;i++)
        {
        if((machb1[i-1]+sb1[arr2[i-1]])>(macha1[i]))
     {maxv11[i]=machb1[i-1]+sb1[arr2[i-1]];}
  else
     {maxv11[i]=macha1[i];}
        machb1[i]=maxv11[i]+b1[arr2[i]];}
//displaying solution
cout<<"\n\n\n\n\n\t\t\t    #####THE SOLUTION##### ";
cout<<"\n\n\t***************************************************************";
cout<<"\n\n\n\t    Optimal Sequence is : ";
for(i=1;i<=n;i++)
        {cout<<" "<<arr2[i];}
cout<<endl<<endl<<"In-Out Table is:"<<endl<<endl;
```

```
cout<<"Jobs"<<"\t"<<"Machine M1"<<"\t"<<"\t"<<"Machine M2" <<endl;
cout<<arr2[1]<<"\t"<<time1<<"--"<<macha1[1]<<" \t"<<"\t"<<macha1[1]<<"--
"<<machb1[1]<<endl;
for(i=2;i<=n;i++)
        {cout<<arr2[i]<<"\t"<<macha12[i]<<"--"<<macha1[i]<<" "<<"\t"<<maxv11[i]<<"--
"<<machb1[i]<<endl;}
cout<<"\n\n\nTotal Elapsed Time (T) = "<<machb1[n];
float L2,L_2,min,u2;L2=machb1[n];float sum_2,sum_3;arr2[0]=0,sb1[0]=0;
for(i=1;i<=n;i++)
        {sum_2=0.0,sum_3=0.0;
for(int j=1;j<=i;j++)
        {sum_3=sum_3+sb1[arr2[j-1]];}
for(int k=1;k<=i;k++)
        {sum_2=sum_2+b1[arr2[k]];}}
L_2=L2-sum_2-sum_3;
cout<<"\nLatest time for which B is taken on Rent="<<"\t"<<L_2;
u2=machb1[n]-L_2;
cout<<"\n\nUtilization Time of Machine M2="<<u2;
cost=(macha1[n]*cost_a)+(u2*cost_b);
cout<<"\n\nThe Minimum Possible Rental Cost is="<<cost;
cout<<"\n\n\t*****************************************************************";
getch();
}
```

9. NUMERICAL ILLUSTRATION

Consider 5 jobs, 2 machine flow shop problem with weights of jobs, processing time and setup time associated with their respective probabilities as given in the following table and jobs 2, 4 are to be processed as a group job (2,4) with breakdown interval as (6,10). The rental cost per unit time for machines M_1 and M_2 are 10 units and 11 units respectively. Our objective is to obtain optimal schedule to minimize the total production time / total elapsed time subject to minimization of the total rental cost of the machines, under the rental policy P.

Job	Machine M_1				Machine M_2				Weight of job
i	a_{i1}	p_{i1}	s_{i1}	q_{i1}	a_{i2}	p_{i2}	s_{i2}	q_{i2}	w_i
1	18	0.1	6	0.1	13	0.1	3	0.1	2
2	12	0.3	5	0.2	8	0.3	4	0.3	3
3	14	0.3	4	0.3	16	0.1	6	0.4	5
4	13	0.2	7	0.3	14	0.2	5	0.1	6
5	15	0.1	4	0.1	6	0.3	4	0.1	4

Table 2

Solution: As per step 1: Expected processing and setup times for machines M_1 and M_2 are as shown in table 3.

Job	Machine M_1		Machine M_2		Weight of job
i	A_{i1}	S_{i1}	A_{i2}	S_{i2}	w_i
1	1.8	0.6	1.3	0.3	2
2	3.6	1.0	2.4	1.2	3
3	4.2	1.2	1.6	2.4	5
4	2.6	2.1	2.8	0.5	6
5	1.5	0.4	1.8	0.4	4

Table 3

As per step 2, 3 & 4: The weighted flow time for two machines M_1 and M_2 are as shown in table 4.

Job	Machine M_1	Machine M_2
i	A''_{i1}	A''_{i2}
1	0.75	1.35
2	0.8	1.4666
3	0.36	1.08
4	0.35	1.1166
5	1.275	0.35

Table 4

As per step 5: Here $\beta = (2,4)$

$A''_{\beta 1} = 0.8 + 0.35 - 0.35 = 0.8$, $A''_{\beta 2} = 1.4666 + 1.1166 - 0.35 = 2.2332$

As per step 7 : Using Johnson's method optimal sequence is S = 3 – 1 – 2 – 4 – 5

As per step 8: The In-Out table for the sequence S is as shown in table 5.

Jobs	Machine M_1	Machine M_2
i	In - Out	In - Out
3	0 – 4.2	4.2 – 5.8
1	5.4 – 7.2	8.2 – 9.5
2	7.8 – 11.4	11.4 – 13.8
4	12.4 – 15.0	15.0 – 17.8
5	17.1 – 18.6	18.6 – 20.4

Table 5

As per step 9&10: The optimal sequence after breakdown effect is S' = 3 – 2 – 4 – 1 – 5

As per step 11: The In-Out table for the sequence S' is as shown in table 6.

Jobs	Machine M_1	Machine M_2
i	In - Out	In - Out
3	0 – 4.2	4.2 – 5.8
2	5.4 – 13.0	13.0 – 15.4
4	14.0 – 16.6	16.6 – 19.4
1	18.7 – 24.5	24.5 – 29.8
5	25.1 – 26.6	30.1 – 31.9

Table 6

Total elapsed time $t_{n,2}(S') = 31.9$ units

As per Step 12: The latest time at which Machine M_2 is taken on rent

$$L_2(S') = t_{n,2}(S') - \sum_{i=1}^{n} A_{i,2}(S') - \sum_{i=1}^{n-1} S_{i,2}(S') = 31.9 - 13.9 - 4.4 = 13.6 \text{ units}$$

As per step 13: The utilization time of Machine M_2 is

$$U_2(S') = t_{n2}(S') - L_2(S') = 31.9 - 13.6 = 18.3 \text{ units}$$

The Biobjective In – Out table is as shown in table 7.

Jobs	Machine M_1	Machine M_2
i	In - Out	In - Out
3	0 – 4.2	13.6 – 15.2
2	5.4 – 13.0	17.6 – 20.0
4	14.0 – 16.6	21.2 – 24.0
1	18.7 – 24.5	24.5 – 29.8
5	25.1 – 26.6	30.1 – 31.9

Table 7

Total Minimum Rental Cost = $R(S') = t_{n,1}(S') \times C_1 + U_2(S') \times C_2$ = 467.3 units

CONCLUSION

If the machine M_2 is taken on rent when it is required and is returned as soon as it completes the last job, the starting of processing of jobs at time $L_2(S) = t_{n,2}(S) - \sum_{i=1}^{n} A_{i,2}(S) - \sum_{i=1}^{n-1} S_{i,2}(S)$ on M_2 will, reduce the idle time of all jobs on it. Also rental cost of M_1 will always be minimum as idle time of M_1 is always zero. The study may further be extending by introducing the concept of transportation time, jobs in a string of disjoint job-block etc.

REFERENCES

1. Johnson, S.M.(1954), "Optimal two and three stage production schedule with set up times included", *Naval Research Logistics Quart.* 1(1), pp.61-68.
2. Bellman,R.(1956) "Mathematical aspects of scheduling theory", *J. Soc. Indust. Appl. Math.*, 4(1), pp.168-205.
3. Smith, W.E.(1956), "Various optimizers for single stage production", *Naval Research Logistics* 3 , pp 59-66.
4. Maggu P.L. & Das G. (1977), "Equivalent jobs for job block in job scheduling", *Opsearch*, 14(4), pp.277-281.
5. Singh , T.P. (1985), "On n x 2 flow shop problem solving job block, Transportation times, Arbitrary time and Break-down machine times", *PAMS*, XXI, No. 1-2.
6. Adiri; I., Bruno, J., Frostig, E. and Kan, R.A.H.G(1989), "Single machine flow time scheduling with a single break-down", *Acta Information*, 26(7), pp.679-696.
7. Chandramouli, A.B. (2005),"Heuristic Approach for n-job,3-machine flow shop scheduling problem involving transportation time, breakdown interval and weights of jobs", M*athematical and Computational Applications*, 10(2), pp.301-305.

8. Belwal, O.K. and Mittal Amit(2008), "n jobs machine flow shop scheduling problem with break-down of machines, transportation time and equivalent job block", *Bulletin of Pure & Applied Sciences-Mathematics*, Jan-June,2008 Source Volume 27(1).
9. P.Pandian and P.Rajendran(2010), "Solving constrained flow-shop scheduling problems with three machines", I*nt. J. Contemp. Math. Sciences,* **5**(19), pp.921-929.
10. Khodadadi, A., (2008), "Development of a new heuristic for three machines flow-shop scheduling problem with transportation time of jobs", *World Applied Sciences Journal,* 5(5), pp.598-601
11. Gupta Deepak & Sharma Sameer (2011), "Minimizing rental cost under specified rental policy in two stage flow shop, the processing time associated with probabilities including breakdown interval and Job-block criteria", *European Journal of Business and Management*, 3(2), pp.85-103.
12. Gupta, D., Sharma, S., Seema and Shefali (2011), "Bicriteria in $n \times 2$ flow shop scheduling under specified rental policy, processing time and setup time each associated with probabilities including job-block", *Industrial Engineering Letters,* 1(1), pp.1 -12.