Bicriteria in n x 3 Flow Shop Scheduling Under Specified Rental Policy, Processing Time Associated with Probabilities including Transportation Time and Job Block Criteria

Sameer Sharma (Corresponding Author)
Research Scholar, Department of Mathematics,
Maharishi Markandeshwar University, Mullana, Haryana, India
samsharma31@yahoo.com

Deepak Gupta
Prof. & Head, Department of Mathematics,
Maharishi Markandeshwar University, Mullana, Haryana, India
guptadeepak2003@yahoo.co.in

Seema Sharma
Assistant Prof, Department of Mathematics,
D.A.V.College, Jalandhar, Punjab, India
seemasharma7788@yahoo.com

Abstract

This paper deals with bicriteria in n-jobs, 3-machines flowshop scheduling problem in which the processing times are associated with probabilities including transportation time and job block criteria. The objective of the study is to obtain an optimal solution for minimizing the bicriteria taken as minimizing the total rental cost of the machines subject to obtains the minimum makespan. A heuristic approach method to find optimal or near optimal sequence has been discussed. A computer programme followed by a numerical illustration is give to clarify the algorithm.

Keywords: Flowshop Scheduling, Heuristic, Processing Time, Transportation Time, Rental Cost, Idle Time, Job block, Makespan

1. Introduction

A flow shop scheduling problem deals with the processing of *i* jobs on *j* machines and determining the sequence and timing of each job on each machine in a fixed order of the machines such that some performance criterion is maximized or minimized. Classical flow shop scheduling problems are mainly concerned with completion time related objectives. However, in modern manufacturing and operations management, the minimization of mean flow time/rental cost of the machines and makespan are the significant factors as for the reason of upward stress of competition on the markets. Recently scheduling, so as to approximate more than one criterion received considerable attention. The bicriteria scheduling problems are motivated by the fact that they are more meaningful from practical point of view. In most manufacturing systems, finished and semi-finished jobs are transferred from one machine to another for further processing. In most of the published literature explicitly or implicitly assumes that either there is an infinite number of jobs are transported instantaneously from one machine to another without transportation time involved. However, there are many situations where the transportation times are quite significant and can not be simply neglected. For example, when the machines on which jobs are to be processed are planted at different places and these jobs require forms of loading-time of jobs, moving time and then unloading-time of jobs. One of the earliest results in flowshop scheduling theory is an algorithm given by Johnson (1954) for scheduling jobs in a two machine flowshop to minimize the time at which all jobs are completed. Smith (1967) considered minimization of mean flow time and maximum tardiness. Van Wassenhove and Gelders (1980) studied minimization of maximum tardiness and mean flow time explicitly as objective. Maggu & Das (1980) consider a two machine flow shop problem with transportation times of jobs in which there is a sufficient number of transporters so that whenever a job is completed at the first machine it can be transported to the second machine immediately, with a job dependent transportation time. Some of the noteworthy heuristic approaches are due to Sen and Gupta (1983), Dileepan et al. (1988), Panwalker (1991), Chandersekharan (1992), Bagga and Bhambani (1997), Narain and Bagga (1998), Chakarvrthy (1999), Chen and Lee. (2001), Narain (2006) and Gupta & Sharma (2011).The basic concept of equivalent job for a job – block has been investigated by Maggu & Das (1977) and established an equivalent job-block theorem. Maggu et al.(1981) studied n jobs two machine sequencing problem with transportation time including equivalent job-for-job block. 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.

Gupta Deepak et al. (2007) studied bicriteria in n jobs two machines flow shop scheduling under predefined rental policy which gives minimum possible rental cost while minimizing total elapsed time. The present paper is an attempt to extend the study made by Gupta Deepak et al. by introducing a bicriteria in n jobs three machines flow shop under specified rental policy. This paper differs with Gupta Deepak et al. (2007) first in the sense that we have proposed heuristic algorithm for three machines based on Johnson's technique, secondly the job block criteria given by Maggu and Das (1977) has been included in the problem and third, the times required by jobs for their transportation from one machine to the other machines is considered. We have obtained an algorithm which gives minimum possible rental cost of machines while minimizing total elapsed time simultaneously.

2. Practical Situation

Various 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. For example, 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.

Many applied and experimental situations exist in our day-to-day working in factories and industrial production concerns etc. In many manufacturing companies different jobs are processed on various machines. These jobs are required to process in a machine shop A, B, C, ---- in a specified order. When the machines on which jobs are to be processed are planted at different places the transportation time (which include loading time, moving time and unloading time etc.) has a significant role in production concern. Further the priority of one job over the other may be significant due to the relative importance of the jobs. It may be because of urgency or demand of that particular job. Hence, the job block criteria become important.

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, 3

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}

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

$L_j(S_k)$: 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 $E_j(S_k)$

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

$T_{i,j\to k}$: Transportation time of i^{th} job from j^{th} machine to k^{th} machine

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

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

β : Equivalent job for job – block.

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

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

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

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

3.2 Definition: Completion time of i^{th} job on machine M_j when M_j starts processing jobs at time L_j is denoted by $t'_{i,j}$ and is defined as

$$t'_{i,j} = L_j + \sum_{k=1}^{i} A_{k,j} = \sum_{k=1}^{i} I_{k,j} + \sum_{k=1}^{i} A_{k,j}\,.\ \text{Also } t'_{i,j} = \max(t_{i,j-1}, t'_{i-1,j}) + A_{i,j}\,.$$

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 and transported to 2nd machine, 3rd machine will be taken on rent at time when 1st job is completed on the 2nd machine and transported.

5. Problem Formulation

Let some job i (i = 1,2,……..,n) is to be processed on three machines M_j (j = 1,2,3) under the specified rental policy P. let a_{ij} be the processing time of i^{th} job on j^{th} machine and let p_{ij} be the probabilities associated with a_{ij}. Let A_{ij} be the expected processing time of i^{th} job on j^{th} machine and $T_{i,j\to k}$ be the transportation time of i^{th} job from j^{th} machine to k^{th} machine. Our aim is to find the sequence $\{S_k\}$ of the jobs which minimize the rental cost of all the three machines while minimizing total elapsed time. The mathematical model of the problem in matrix form can be stated as:

Jobs	Machine A		$T_{i,1\to 2}$	Machine B		$T_{i,2\to 3}$	Machine C	
i	a_{i1}	p_{i1}		a_{i2}	p_{i2}		a_{i3}	p_{i3}
1	a_{11}	p_{11}	$T_{1,1\to 2}$	a_{12}	p_{12}	$T_{1,2\to 3}$	a_{13}	p_{13}
2	a_{21}	p_{21}	$T_{2,1\to 2}$	a_{22}	p_{22}	$T_{2,2\to 3}$	a_{23}	p_{23}
3	a_{31}	p_{31}	$T_{3,1\to 2}$	a_{32}	p_{32}	$T_{3,2\to 3}$	a_{33}	p_{33}
4	a_{41}	p_{41}	$T_{4,1\to 2}$	a_{42}	p_{42}	$T_{4,2\to 3}$	a_{43}	p_{43}
-	-	-	-	-	-	-	-	-
n	a_{n1}	p_{n1}	$T_{n,1\to 2}$	a_{n2}	p_{n2}	$T_{n,2\to 3}$	a_{n3}	p_{n3}

(Table 1)

Mathematically, the problem is stated as:

Minimize $U_j(S_k)$ and Minimize $R(S_k) = \sum_{i=1}^{n} A_{i1} \times C_1 + U_2(S_k) \times C_2 + \sum_{i=1}^{n} A_{i3} \times C_3$

Subject to constraint: Rental Policy (P)

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

6. Theorems

6.1 Theorem: The processing of jobs on M_3 at time $L_3 = \sum_{i=1}^{n} I_{i,3}$ keeps $t_{n,3}$ unaltered.

Proof: Let $t'_{i,3}$ be the competition time of i^{th} job on machine M_3 when M_3 starts processing of jobs at time L_3. We shall prove the theorem with the help of Mathematical Induction.

Let $P(n): t'_{n,3} = t_{n,3}$

Basic Step: For n = 1

$$t'_{1,3} = L_3 + A_{1,3} = I_{1,3} + A_{1,3} = (A_{1,1} + (T_{1,1\to 2} + A_{1,2}) + T_{1,2\to 3}) + A_{1,3} = t_{1,3}.$$

Therefore P(1) is true.

Induction Step: Let P (k) be true. i.e. $t'_{k,3} = t_{k,3}$.

Now, we shall show that P(k+1) is also true.

i.e. $t'_{k+1,3} = t_{k+1,3}$

But $t'_{k+1,3} = \max(t_{k+1,2}, t'_{k,3}) + T_{k+1,2\to 3} + A_{k+1,3}$ (As per Definition 2)

$$\therefore t'_{k+1,3} = \max(t_{k+1,2}, L_3 + \sum_{i=1}^{k} A_{i,3}) + T_{k+1,2\to 3} + A_{k+1,3} =$$

$$\max(t_{k+1,2}, \sum_{i=1}^{k+1} I_{1,3} + \sum_{i=1}^{k} A_{i,3}) + T_{k+1,2\to 3} + A_{k+1,3}$$

$$= \max(t_{k+1,2}, \sum_{i=1}^{k} I_{1,3} + \sum_{i=1}^{k} A_{i,3} + I_{k+1,3}) + T_{k+1,2\to 3} + A_{k+1,3}$$

$$= \max(t_{k+1,2}, t_{k,3} + I_{k+1,3}) + T_{k+1,2\to 3} + A_{k+1,3}$$

$$= \max(t_{k+1,2}, t'_{k,3} + I_{k+1,3}) + T_{k+1,2\to 3} + A_{k+1,3} \quad \text{(by assumption)}$$

$$= \max\left(t_{k+1,2}, t'_{k,3} + \max\left(\left(t_{k+1,2} - t_{k,3}\right), 0\right)\right) + T_{k+1,2\to 3} + A_{k+1,3}$$

$$= \max\left(t_{k+1,2}, t_{k,3}\right) + T_{k+1,2\to 3} + A_{k+1,3} = t_{k+1,3}$$

Hence by principle of mathematical induction P(n) is true for all n, .i.e. $t'_{n,3} = t_{n,3}$.

Lemma 6.1 If M_3 starts processing jobs at $L_3 = \sum_{i=1}^{n} I_{i,3}$ then

(i). $L_3 > t_{1,2}$

(ii). $t'_{k+1,3} \geq t_{k,2}$, $k > 1$.

6.2 Theorem: The processing of jobs on M_2 at time $L_2 = \min_{i \leq k \leq n} \{Y_k\}$ keeps total elapsed time unaltered where $Y_1 = L_3 - A_{1,2} - T_{1,2\to 3}$ and $Y_k = t'_{k-1,3} - \sum_{i=1}^{k} A_{i,2} - \sum_{i=1}^{k} T_{i,2\to 3}; k > 1$.

Proof. We have $L_2 = \min_{i \leq k \leq n} \{Y_k\} = Y_r$ (say)

In particular for k =1

$$Y_r \leq Y_1 \Rightarrow Y_r + A_{1,2} + T_{1,2\to 3} \leq Y_1 + A_{1,2} + T_{1,2\to 3}$$

$$\Rightarrow Y_r + A_{1,2} + T_{1,2\to 3} \leq L_3 \quad \text{----- (1)} \quad \left(\because Y_1 = L_3 - A_{1,2} - T_{1,2\to 3}\right)$$

By Lemma 1; we have

$$t_{1,2} \leq L_3 \quad \text{---- (2)}$$

Also, $t'_{1,2} = \max\left(Y_r + A_{1,2} + T_{1,2\to3}, t_{1,2}\right)$

On combining, we get

$$t'_{1,2} \le L_3$$

For k >1, As $Y_r = \min_{i\le k\le n}\{Y_k\}$

$$\Rightarrow Y_r \le Y_k; \qquad k = 2,3\ldots\ldots\ldots,n$$

$$\Rightarrow Y_r + \sum_{i=1}^{k} A_{i,2} + \sum_{i=1}^{k} T_{i,2\to3} \le Y_k + \sum_{i=1}^{k} A_{i,2} + \sum_{i=1}^{k} T_{i,2\to3}$$

$$\Rightarrow Y_r + \sum_{i=1}^{k} A_{i,2} + \sum_{i=1}^{k} T_{i,2\to3} \le t'_{k-1,3} \quad \text{---- (3)}$$

By Lemma 1; we have

$$t_{k,2} \le t'_{k-1,3} \quad \text{---- (4)}$$

Also, $t'_{k,2} = \max\left(Y_r + \sum_{i=1}^{k} A_{i,2} + \sum_{i=1}^{k} T_{i,2\to3}, t_{k,2}\right)$

Using (3) and (4) , we get

$$t'_{k,2} \le t'_{k-1,3}$$

Taking $k = n$, we have

$$t'_{n,2} \le t'_{n-1,3} \quad \text{---- (5)}$$

Total time elapsed = $t_{n,3}$

$$= \max\left(t'_{n,2}, t'_{n-1,3}\right) + A_{n,3} + T_{n,2\to3} = t'_{n-1,3} + A_{n,3} + T_{n,2\to3} = t'_{n,3}. \text{ (using 5)}$$

Hence, the total time elapsed remains unaltered if M_2 starts processing jobs at time $L_2 = \min_{i\le k\le n}\{Y_k\}$.

6.3 Theorem: The processing time of jobs on M_2 at time $L_2 > \min_{i\le k\le n}\{Y_k\}$ increase the total time elapsed, where $Y_1 = L_3 - A_{1,2} - T_{1,2\to3}$ and $Y_k = t'_{k-1,3} - \sum_{i=1}^{k} A_{i,2} - \sum_{i=1}^{k} T_{i,2\to3}; k > 1$.

The proof of the theorem can be obtained on the same lines as of the previous Theorem 6.2.

By Theorem 1, if M_3 starts processing jobs at time $L_3 = t_{n,3} - \sum_{i=1}^{n} A_{i,3}$ then the total elapsed time $t_{n,3}$ is not altered and M_3 is engaged for minimum time equal to sum of processing times of all the jobs on M_3, .i.e. reducing the idle time of M_3 to zero. Moreover total elapsed time/rental cost of M_1 is always least as idle time of M_1 is always zero. Therefore the objective remains to minimize the elapsed time and hence the rental cost of M_2.The following algorithm provides the procedure to determine the times at which machines should be taken on rent to minimize the total rental cost without altering the total elapsed time in three machine flow shop problem under rental policy (P).

7. Algorithm

Step 1: Calculate expected processing time $A_{ij} = a_{ij} \times p_{ij}; \forall i, j = 1,2,3$.

Step 2: Check the condition

either $\text{Min}\left\{A_{i1} + T_{i,1\to2}\right\} \ge \text{Max}\left\{A_{i2} + T_{i,2\to3}\right\}$

or $\text{Min}\left\{T_{i,2\to3} + A_{i3}\right\} \ge \text{Max}\left\{A_{i2} + T_{i,2\to3}\right\}$ or Both for all i.

If the conditions are satisfied then go to Step 3, else the data is not in the standard form.

Step 3: Introduce the two fictitious machines G and H with processing times G_i and H_i as

$G_i = A_{i1} + T_{i,1\to2} + A_{i2} + T_{i,2\to3}$, $H_i = T_{i,1\to2} + A_{i2} + T_{i,2\to3} + A_{i3}$ for all i.

Step 4: Find the expected processing time of job block $\beta = (k, m)$ on fictitious machines G & H using equivalent job block criterion given by Maggu & Das (1977). Find G_β and H_β using

$$G_\beta = G_k + G_m - \min(G_m, H_k),\ H_\beta = H_k + H_m - \min(G_m, H_k).$$

Step 5: Define a new reduced problem with processing time G_i and H_i as defined in step 3 and replace job block *(k, m)* by a single equivalent job β with processing times G_β and H_β as defined in step 4.

Step 6: using Johnson's procedure, obtain all the sequences S_k having minimum elapsed time. Let these be $S_1, S_2, ..., S_r$.

Step 7: Prepare In-Out tables for S_k and compute total elapsed time $t_{n3}(S_k)$.

Step 8: Compute latest time L_3 of machine M_3 for sequence S_k as $L_3(S_k) = t_{n3}(S_k) - \sum_{i=1}^{n} A_{i3}$

Step 9: For the sequence S_k (k = 1,2,…………...,r), compute

I. $t_{n2}(S_k)$

II. $Y_1(S_k) = L_3(S_1) - A_{1,2}(S_k) - T_{1,1\to 2}$

III. $Y_q(S_k) = L_3(S_1) - \sum_{i=1}^{q} A_{12}(S_k) - \sum_{i=1}^{q} T_{i,2\to 3} + \sum_{i=1}^{q-1} A_{i,3} + \sum_{i=1}^{q-1} T_{i,1\to 2}; q = 2,3,......, n$

IV. $L_2(S_k) = \min_{1\le q\le n} \{Y_q(S_k)\}$

V. $U_2(S_k) = t_{n2}(S_k) - L_2(S_k)$.

Step 10: Find min $\{U_2(S_k)\}; k = 1, 2, ..., r$

Let it be for the sequence S_p, and then sequence S_p will be the optimal sequence.

Step 11: Compute total rental cost of all the three machines for sequence S_p as:

$$R(S_p) = \sum_{i=1}^{n} A_{i1} \times C_1 + U_2(S_p) \times C_2 + \sum_{i=1}^{n} A_{i3} \times C_{3.}$$

8. Programme

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<process.h>
int n,j;float a1[16],b1[16],c1[16],g[16],h[16],T12[16],T23[16], macha[16],machb[16],machc[16];
float cost_a,cost_b,cost_c,cost;
int f=1;int group[2];//variables to store two job blocks
float minval,minv,maxv1[16],maxv2[16], gbeta=0.0,hbeta=0.0;
void main()
{       clrscr();
        int a[16],b[16],c[16],j[16];float p[16],q[16],r[16];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 of "<<i<<" job for machine A and Transportation time from Machine A to B : ";cin>>a[i]>>p[i]>>T12[i]; cout<<"\nEnter the processing time and its probability of "<<i<<" job for machine B and Transportation time from Machine B to C : ";cin>>b[i]>>q[i]>>T23[i];
cout<<"\nEnter the processing time and its probability of "<<i<<"job for machine C: ";
```

```
cin>>c[i]>>r[i];
        //Calculate the expected processing times of the jobs for the machines:
        a1[i] = a[i]*p[i];b1[i] = b[i]*q[i];c1[i] = c[i]*r[i];}
        cout<<"\nEnter the rental cost of Machine M1:";cin>>cost_a;
        cout<<"\nEnter the rental cost of Machine M2:";cin>>cost_b;
        cout<<"\nEnter the rental cost of Machine M3:";cin>>cost_c;
        cout<<endl<<"Expected processing time of machine A, B and C: \n";
for(i=1;i<=n;i++)
{cout<<j[i]<<"\t"<<a1[i]<<"\t"<<b1[i]<<"\t"<<c1[i]<<"\t";cout<<endl;}
        //Finding smallest in a1
float mina1;mina1=a1[1]+T12[1];
for(i=2;i<n;i++){if(a1[i]+T12[i]<mina1)   mina1=a1[i]+T12[i];}
//For finding largest in b1
float maxb1;maxb1=b1[1]+T23[1];
for(i=2;i<n;i++){if(b1[i]+T23[i]>maxb1)maxb1=b1[i]+T23[i];}
//Finding smallest in c1
float       minc1;minc1=c1[1]+T23[1];for(i=2;i<n;i++){if(c1[i]+T23[i]<minc1)
minc1=c1[i]+T23[i];}
if(mina1>=maxb1||minc1>=maxb1)
{for(i=1;i<=n;i++)
{g[i]=a1[i]+T12[i]+b1[i]+T23[i];h[i]=T12[i]+b1[i]+T23[i]+c1[i];}}
else    {cout<<"\n data is not in Standard Form...\nExitting";getch();exit(0);}
cout<<endl<<"Expected processing time for two fictious machines G and H: \n";
 for(i=1;i<=n;i++)
 {cout<<endl;cout<<j[i]<<"\t"<<g[i]<<"\t"<<h[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 g1[16],h1[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++){g1[i]=g[j1[i]];h1[i]=h[j1[i]];}
g1[n-1]=gbeta;h1[n-1]=hbeta;cout<<endl<<endl<<"displaying original scheduling table"<<endl;
for(i=1;i<=n-1;i++){cout<<j1[i]<<"\t"<<g1[i]<<"\t"<<h1[i]<<endl;}float mingh[16];char ch[16];
 for(i=1;i<=n-1;i++)
  {if(g1[i]<h1[i])
        {mingh[i]=g1[i];ch[i]='g';}
else    {mingh[i]=h1[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;macha[1]=time+a1[arr1[1]];
for(i=2;i<=n;i++)
        {macha[i]=macha[i-1]+a1[arr1[i]];}machb[1]=macha[1]+b1[arr1[1]]+T12[arr1[1]];
for(i=2;i<=n;i++)
        {if((machb[i-1])>(macha[i]+T12[arr1[i]]))maxv1[i]=machb[i-1];
Else    maxv1[i]=macha[i]+T12[arr1[i]];machb[i]=maxv1[i]+b1[arr1[i]];}
        machc[1]=machb[1]+c1[arr1[1]]+T23[arr1[1]];
for(i=2;i<=n;i++)
        {if((machc[i-1])>(machb[i]+T23[arr1[i]]))maxv2[i]=machc[i-1];
else    maxv2[i]=machb[i]+T23[arr1[i]];machc[i]=maxv2[i]+c1[arr1[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<<" "<<arr1[i];} cout<<endl<<endl<<"In-Out Table is:"<<endl<<endl;
cout<<"Jobs"<<"\t"<<"Machine    M1"<<"\t"<<"\t"<<"Machine    M2"    <<"\t"<<"\t"<<"Machine
M3"<<endl;
cout<<arr1[1]<<"\t"<<time<<"--"<<macha[1]<<"\t"<<"\t"<<macha[1]+T12[arr1[1]]<<"--
"<<machb[1]<<" \t"<<"\t"<<machb[1]+T23[arr1[1]]<<"--"<<machc[1]<<endl;
for(i=2;i<=n;i++)
{cout<<arr1[i]<<"\t"<<macha[i-1]<<"--"<<macha[i]<<" "<<"\t"<<maxv1[i]<<"--"<<machb[i]<<"
"<<"\t"<<maxv2[i]<<"--"<<machc[i]<<endl;}
cout<<"\n\n\nTotal Elapsed Time (T) = "<<machc[n];
float L3,Y[16],min,u2;float sum1=0.0,sum2=0.0,sum3=0.0;
for(i=1;i<=n;i++)
{sum1=sum1+a1[i];sum2=sum2+b1[i];sum3=sum3+c1[i];}L3=machc[n]-sum3;
cout<<"\n\nLatest Time When Machine M3 is Taken on Rent:"<<L3;
```

```
cout<<"\n\nTotal Completion Time of Jobs on Machine M2:"<<machb[n];
Y[1]=L3-b1[arr1[1]]-T23[arr1[1]];cout<<"\n\n\tY[1]\t="<<Y[1];float sum_2,sum_3;
for(i=2;i<=n;i++)
{sum_2=0.0,sum_3=0.0;for(int j=1;j<=i-1;j++){sum_3=sum_3+c1[arr1[j]]+T12[arr1[j]];}
for(int k=1;k<=i;k++)
{sum_2=sum_2+b1[arr1[k]]+T23[arr1[k]];}Y[i]=L3+sum_3-
sum_2;cout<<"\n\n\tY["<<i<<"]\t="<<Y[i];}
min=Y[1];
for(i=2;i<n;i++){if(Y[i]<min)min=Y[i];}
cout<<"\n\nMinimum of Y[i]="<<min;u2=machb[n]-min;
cout<<"\n\nUtilization Time of Machine M2="<<u2;cost=(sum1*cost_a)+(u2*cost_b)+(sum3*cost_c);
cout<<"\n\nThe Minimum Possible Rental Cost is="<<cost;
cout<<"\n\n\t*************************************************************";
getch();
}
```

9. Numerical Illustration

Consider 5 jobs, 3 machine flow shop problem with processing time associated with their respective probabilities and transportation time as given in table and jobs 2 and 4 are processed as a group job (2, 4). The rental cost per unit time for machines M_1, M_2 and M_3 are 6 units, 11 units and 7 units respectively, under the rental policy P.

Jobs	Machine M_1		$T_{i,1\to 2}$	Machine M_2		$T_{i,2\to 3}$	Machine M_3	
i	a_{i1}	p_{i1}		a_{i2}	p_{i2}		a_{i3}	p_{i3}
1	18	0.1	2	4	0.2	2	13	0.1
2	12	0.3	1	6	0.2	1	8	0.3
3	14	0.3	3	5	0.2	2	16	0.1
4	13	0.2	2	4	0.2	2	4	0.2
5	15	0.1	4	6	0.2	1	6	0.3

(Table 2)

Our objective is to obtain an optimal schedule for above said problem to minimize the total production time / total elapsed time subject to minimization of the total rental cost of the machines.

Solution: As per Step 1;The expected processing times for machines M_1, M_2 and M_3 are as in **table 3.**

As per Step 2: Here, Min $\{A_{i1} + T_{i,1\to 2}\} \geq$ Max $\{A_{i2} + T_{i,2\to 3}\}$

As per Step 3,4,5 & 6: The optimal sequence is S = 5 – 3 –1 - β , .i.e. S= 5 – 3 – 1 – 2 – 4

As per Step 7: The In – Out table for the optimal sequence S is as in **table 4.**

$Y_1 = 11.4 - 1.2 - 1.0 = 9.2$

As per Step 8: $L_3(S) = t_{n3}(S) - \sum_{i=1}^{n} A_{i,3}(S) = 19.3 - 7.9 = 11.4$

$Y_2 = 11.4 - 5.2 + 5.8 = 12.0$

$Y_3 = 11.4 - 8 + 4.1 = 13.8$

As per Step 9: For sequence S, we have $t_{n2}(S) = 20.4$ and $Y_4 = 11.4 - 10.2 + 13.7 = 14.9$

The new reduced Bi-objective In – Out table is as shown in **table 5.**

$Y_5 = 11.4 - 13 + 17.1 = 15.5$

The latest possible time at which machine M_2 should be taken on rent $L_2(S) = Min\{Y_k\} = \mathbf{9.2}$ units.

Also, utilization time of machine M_2 = $U_2(S)$ = 7.3 units. $U_2(S) = t_{n2}(S) - L_2(S) = 16.5 - 9.2 = 7.3$

Total Minimum rental cost = $R(S_p) = \sum_{i=1}^{n} A_{i1} \times C_1 + U_2(S_p) \times C_2 + \sum_{i=1}^{n} A_{i3} \times C_{3.}$

$$= 13.7 \times 6 + 7.3 \times 11 + 7.9 \times 7 = 217.8 \text{ Units}$$

Hence 5 – 3 – 1 – 2 – 4 is the optimal sequence with total rental cost of machines as 217.8 units when M_1 starts processing job (.i.e. taken on rent) at time 0 units, M_2 at 9.2 units and M_3 at time 11.4 units.

10 Conclusion

If M_3 starts processing jobs at time $L_3 = t_{n,3} - \sum_{i=1}^{n} A_{i,3}$ then the total elapsed time $t_{n,3}$ is not altered and M_3 is engaged for minimum time equal to sum of processing times of all the jobs on M_3, i.e. reducing the idle time of M_3 to zero. 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 $L_2(S_k) = \min_{1 \le q \le n} \{Y_q(S_k)\}$, $Y_1(S_k) = L_3(S_1) - A_{1,2}(S_k) - T_{1,1\to 2}$, $Y_q(S_k) = L_3(S_1) - \sum_{i=1}^{q} A_{i2}(S_k) - \sum_{i=1}^{q} T_{i,2\to 3} + \sum_{i=1}^{q-1} A_{i,3} + \sum_{i=1}^{q-1} T_{i,1\to 2}; q = 2,3,......,n$ on M_2 will, reduce the idle time of all jobs on it. Therefore total rental cost of M_2 will be minimum. Also rental cost of M_1 and M_3 will always be minimum since idle times of M_1 and M_3 is always zero.

References

Bagga, P.C. & Ambika Bhambani (1997),"Bicriteria in flow shop scheduling problem", *Journal of Combinatorics, Information and System Sciences*, 22(1), 63-83.

Chandrasekharan, R. (1992), "Two Stage flow shop scheduling problem with bicriteria", *Operational Res. Soc*, 43(9), 871-884.

Chakarvarthy, K. & Rajendrah, C. (1999), "A heuristic for scheduling in a flow shop with bicriteria of makespan and maximum tardiness minimization", *Production Planning & Control*, 10(7), 707-714.

Chen, ZL. & Lee, CY. (2001), "Machine scheduling with transportation considerations" *USA; Journal of scheduling*, 4, 3-24.

Chikhi, N. (2008), **"**Two machine flow-shop with transportation time**"**, *Thesis of Magister*, Faculty of Mathematics, USTHB University, Algiers.

Dileepan, P. & Sen, T. (1988)**,** "Bicriteria state scheduling research for a single machine", *Omega*, 16, 53-59.

Gupta, D., Singh, T.P. & Kumar, R. (2007), "Bicriteria in scheduling under specified rental policy, processing time associated with probabilities including job block concept", *Proceedings of VIII Annual Conference of Indian Society of Information Theory and Application (ISITA)*, 22-28.

Gupta, D.& Sharma, S. (2011), "Minimizing rental cost under specified rental policy in two stage flow shop, the processing time associated with probabilities including break-down interval and job – block criteria" , *European Journal of Business and Management*, 3(2),85-103.

Johnson, S.M. (1954), "Optimal two and three stage production schedule with set up times included", *Naval Research Logistics Quart.*, 1(1), 61-68.

Khodadadi, A. (2008), "Development of a new heuristic for three machine flow-shop scheduling with transportation time of job", *World Applied Sciences Journal*, 5(5),598- 601.

Maggu, P.L. & Das, G. (1977), "Equivalent jobs for job block in job scheduling", *Opsearch*, 14(4), 277-281.

Maggu, P.L. & Das, G. (1980), "On n x 2 sequencing problem with transportation times of jobs", Pure and applied Mathematika Sciences, 12-16.

Maggu, P.L., Das, G. & Kumar, R.(1981), "On equivalent job-for-job block in 2 x n sequencing problem with transportation times", *Journal of OR Society of Japan*, 24, 136-146.

Narian,L. & Bagga, P.C. (1998), "Minimizing hiring cost of machines in n $\times$ 3 flow shop problem", *XXXI Annual ORSI Convention and International Conference on Operation Research and Industry*, Agra[India].

Narain, L. (2006), "Special models in flow shop sequencing problem", *Ph.D. Thesis*, University of Delhi, Delhi.

Panwalker, S. S. (1991), "Scheduling of a two machine flow-shop with travel time between machines", *J.Opl.Res.Soc*, 42(7), 609-613.

Smith, R.D. & Dudek, R.A. (1967), "**A** general algorithm for solution of the N-job, M-machine scheduling problem", *Opn. Res.*, 15(1), 71-82.

Sen, T. & Gupta, S.K. (1983), **"**A branch and bound procedure to solve a bicriteria scheduling problem", *AIIE Trans*., 15, 84-88.

Van Wassenhove, L.N. & Gelders, L.F. (1980), "Solving a bicriteria scheduling problem", *AIIE Tran 15s*. 84-88.

Tables

Table 3: The expected processing times for machines M_1, M_2 and M_3 are

Jobs	A_{i1}	$T_{i,1\to 2}$	A_{i2}	$T_{i,2\to 3}$	A_{i3}
1	1.8	2	0.8	2	1.3
2	3.6	1	1.2	1	2.4
3	4.2	3	1.0	2	1.6
4	2.6	2	0.8	2	0.8
5	1.5	4	1.2	1	1.8

Table 4: The In – Out table for the optimal sequence S is

Jobs	Machine M_1	$T_{i,1\to 2}$	Machine M_2	$T_{i,2\to 3}$	Machine M_3
i	In – Out		In – Out		In - Out
5	0.0 – 1.5	4	5.5 – 6.7	1	7.7 – 9.5
3	1.5 – 5.7	3	8.7 – 9.7	2	11.7 – 13.3
1	5.7 – 7.5	2	9.7 – 10.5	2	13.3 – 14.6
2	7.5 – 11.1	1	12.1 – 13.3	1	14.6 – 17.0
4	11.1 – 13.7	2	15.7 – 16.5	2	18.5 – 19.3

Table 5: The new reduced Bi-objective In – Out table is

Jobs	Machine M_1	$T_{i,1\to 2}$	Machine M_2	$T_{i,2\to 3}$	Machine M_3
i	In – Out		In – Out		In - Out
5	0.0 – 1.5	4	9.2 – 10.4	1	11.4 – 13.2
3	1.5 – 5.7	3	10.4 – 11.0	2	13.2 – 14.8
1	5.7 – 7.5	2	11.0 – 11.8	2	14.8 – 16.1
2	7.5 – 11.1	1	12.1 – 13.3	1	16.1 – 18.5
4	11.1 – 13.7	2	15.7 – 16.5	2	18.5 – 19.3