A Heuristic Approach for Two Stage Open Shop Scheduling with Transportation Time and Weightage of Jobs Including Job Block Criterion, the Processing Time Associated with Probabilities

Deepak Gupta, Renuka, Payal Singla

Department of Mathematics, M.M. University, Mullana, Ambala, India

Abstract- This Paper provides a heuristic algorithm for two stage open shop scheduling problem to minimize the makespan in which processing times are associated with their respective probabilities including the concepts of job block criteria and transportation time. Further weights are attached with jobs to indicate their relative importance. A computer programme followed by a numerical illustration is given to justify the proposed algorithm.

Index Terms- Open Shop Scheduling, Processing time, Makespan, Equivalent job, Job block, Transportation time, Weightage.

I. Introduction

Now a day, Scheduling has a significant role in the field of manufacturing, management, computer Science, production concern etc. An appropriate scheduling provides useful results to the Decision maker and Industrial Supervisor for minimizing the manufacturing cost and time. In flow shop scheduling problem, each job is processed on all the machines in the same order with the objective to optimize some well defined criteria while in Open shop scheduling, the order of the machines, on which the jobs are processed, is not fixed and not known in advance. Hence the order of the machines can be selected arbitrarily. For instant, in a large aircraft garage with specialized work center, an airplane may be requiring repairs in engine and electrical circuit. These two tasks may be carried out in any order but it is not possible to do tasks on the same plane simultaneously.

The research work, in most of the literature has been done in the field of flow shop scheduling problem. The research work on two machine flow shop problem was first of all introduced by Johnson [11] with the objective to minimize the makespan (i.e. total elapsed time).The work was extended by many researchers like Ignall and Schrage[10],Bagga [2], Compbell[3], Maggu and Das[12],Yoshida and Hitomi[15], Singh T.P. [13], Harbans and Maggu [9], ChandraShekhara [4], V.A.Strusevich[14], Anup[1],Gupta and Singh[5], Gupta and Sharma[6,7] by considering various parameters. The basic concept of equivalent job for a job block was investigated by Maggu & Das [12], in which priority to one job over another was considered in order to create a balance between a cost of providing priority in service and cost of giving service with non priority.

Gupta &Singh et.al[8] have studied the optimal two stage **open shop scheduling** problem with the concept of job block criteria in which they associated processing time with their respective probabilities. Here we have extended their work by introducing the concepts transportation time (which involves loading time, moving time and unloading time etc.) and weightage of jobs in order to indicate their relative importance. Thus the problem in the present paper becomes wider and practically more applicable. An algorithm has been developed to minimize the maximum completion time (makespan). The algorithm is demonstrated through a numerical example.

II. Practical Situation

Many applied and experimental situations occur in day to day working in factories and industrial concern. Open shop scheduling problem has wide engineering applications in manufacturing. The practical situation of open shop scheduling may be taken in automobile repair, quality control centers, semiconductor manufacturing, Satellite Communications, Class assignment etc. For example in a automobile garage with specialized shops. A Car may require the following work: Replacement of exhaust and muffler and alignment of wheels. These two tasks will be performed on two machines and may be carried out in any order on the given machines. Similarly an airplane may be requiring repairs in engine and electrical circuit. These two tasks may be carried out in any order. In many manufacturing companies different jobs are planted at different places then the transportation time has a significant role in production concern. Sometimes, the priority of one job over the other is preferred. It may be because of urgency or demand of its relative importance. Hence the job block criteria becomes significant

III. Notations

- A_i : Processing time of i^{th} job on machine A.
- B_i : Processing time of i^{th} job on machine B.
- A_i' : Expected Processing time of i^{th} job on machine A.
- B_i' : Expected Processing time of i^{th} job on machine B.

- p_i : Probability associated to the processing time A_i of ith job on machine A.
- q_i : Probability associated to the processing time B_i of i^{th} job on machine B.
- $T_{i,A\rightarrow B}$: Transportation time from machine A to B which is same as $T_{i,B\rightarrow A}$ Transportation time from machine B to A .
- w_i : Weight assigned to i^{th} job
- G_i : Processing time of i^{th} job on fictitious machine G
- H_i : Processing time of i^{th} job on fictitious machine H
- S_{AB} : Optimal/near optimal sequence for the machine order A to B
- S_{BA} : Optimal/near optimal sequence for the machine order B to A

IV. Problem Formulation

Let n jobs 1,2,3,…..n be processed through two machines A and B. Let A_i and B_i be the processing time of i^{th} job (i=1,2,3….n) on machine A and B respectively. Let pi and qi be the probabilities associated with processing time A_i and B_i respectively such that $0 \le p_i \le 1$, $\sum p_i=1$, $0 \le q_i \le 1$, $\sum q_i =1$. Let w_i be the weights of i^{th} job (i = 1,2,3…..n) and $T_{i,A\rightarrow B}$ be the transportation time of i^{th} job from machine A to B which is same as transportation time from machine B to A i.e. $T_{i,A\rightarrow B}$ is same as $T_{i,B\rightarrow A}$. Let α be an equivalent job for a job block (k,m) in which k is given priority over job m, where k and m are any jobs among the given n jobs.

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

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

Jobs i	Machine A		Transportation Time of i^{th} job from machine A to B or B to A	Machine B		w_i
	A_i	p_i		B_i	q_i	
1	A_1	p_1	$T_{1,A\rightarrow B}$ or $T_{1,B\rightarrow A}$	B_1	q_1	w_1
2	A_2	p_2	$T_{2,A\rightarrow B}$ or $T_{2,B\rightarrow A}$	B_2	q_2	w_2
3	A_3	p_3	$T_{3,A\rightarrow B}$ or $T_{3,B\rightarrow A}$	B_3	q_3	w_3
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
n	A_n	p_n	$T_{n,A\rightarrow B}$ or $T_{n,B\rightarrow A}$	B_n	q_n	w_n

Tableau-1

Our objective is to find the optimal or near optimal schedule of all the jobs which minimize the total elapsed time of each machine and weighted mean flow time whenever mean weighted production flow time is taken into consideration.

V. Assumptions

1. Each job once started, must be processed to completion.
2. Two jobs can't be processed on a single machine at a time.
3. Jobs are independent to each other
4. In a job block ($i_1,i_2\ldots\ldots\ldots i_k$), priority is given to the i_1 job over $i_2\ldots.i_k$
5. Transportation times from machine A to B and B to A are same.
6. Transporting device is always available.
7. $\sum p_i=1$, $\sum q_i=1$, $0\le p_i\le 1$, $0\le q_i\le 1$.

VI. Algorithm

The heuristic algorithm for the open shop scheduling can be decomposed in the following steps:

Step1: Define expected processing time A'_i and B'_i on machine A and B respectively as follows:-

a) $A'_i = A_i \times p_i$

b) $B'_i = B_i \times q_i$

Step2: Define two fictitious machines G and H with processing times G_i and H_i as follows:-

(a)If $\min(A'_i,\ B'_i) = A'_i$ then calculate

$G_i = A'_i + T_{i,A\rightarrow B} + w_i$ and $H_i = B'_i + T_{i,A\rightarrow B}$

(b)If $\min(A'_i,\ B'_i) = B'_i$ then calculate

ISSN 2250-3153

$G_i = A'_i + T_{i,A\to B}$ and $H_i = B'_i + T_{i,A\to B} + w_i$

Step3: Calculate G'_i and H'_i as follows:-

(a) $G'_i = G_i / w_i$

(b) $H'_i = H_i / w_i$

Step4: Calculate the expected processing time for the equivalent job α = (k, m) on fictitious machines G and H using[12]:-

For the machine order G→H

(a) $G'_\alpha = G'_k + G'_m - \min\left(G'_m, H'_k\right)$

(b) $H'_\alpha = H'_k + H'_m - \min\left(G'_m, H'_k\right)$

For the machine order H→G

(a) $H'_\alpha = H'_k + H'_m - \min\left(G'_k, H'_m\right)$

(b) $G'_\alpha = G'_k + G'_m - \min(G'_k, H'_m)$

Step5: Define a new reduced problem with processing times G'_i and H'_i as define in step 3 & 4

Step6: For the machine order G→H

Construct a set S_G of all the processing times G'_i where $G'_i \le H'_i$ and S'_G of all the processing times G'_i where $G'_i \ge H'_i$.

Step7: Let S_1 denote a sub optimal sequence of jobs corresponding to non decreasing times S_G and similarly a sequence S_2 corresponding to set S'_G.

Step8: The augmented ordered sequence $S_{AB} = (S_1, S_2)$ gives optimal or near optimal sequence of jobs for the machine order A to B

Step9: For the machine order H→G

Construct the set S_H and S'_H of processing times H'_i Where $H'_i \le G'_i$ and of processing times H'_i where $H'_i \ge G'_i$ respectively.

Step10: Let S_2 denote a sub optimal sequence of jobs corresponding to the non decreasing processing times in the set S_H. Similarly S'_2 corresponding to S'_H.

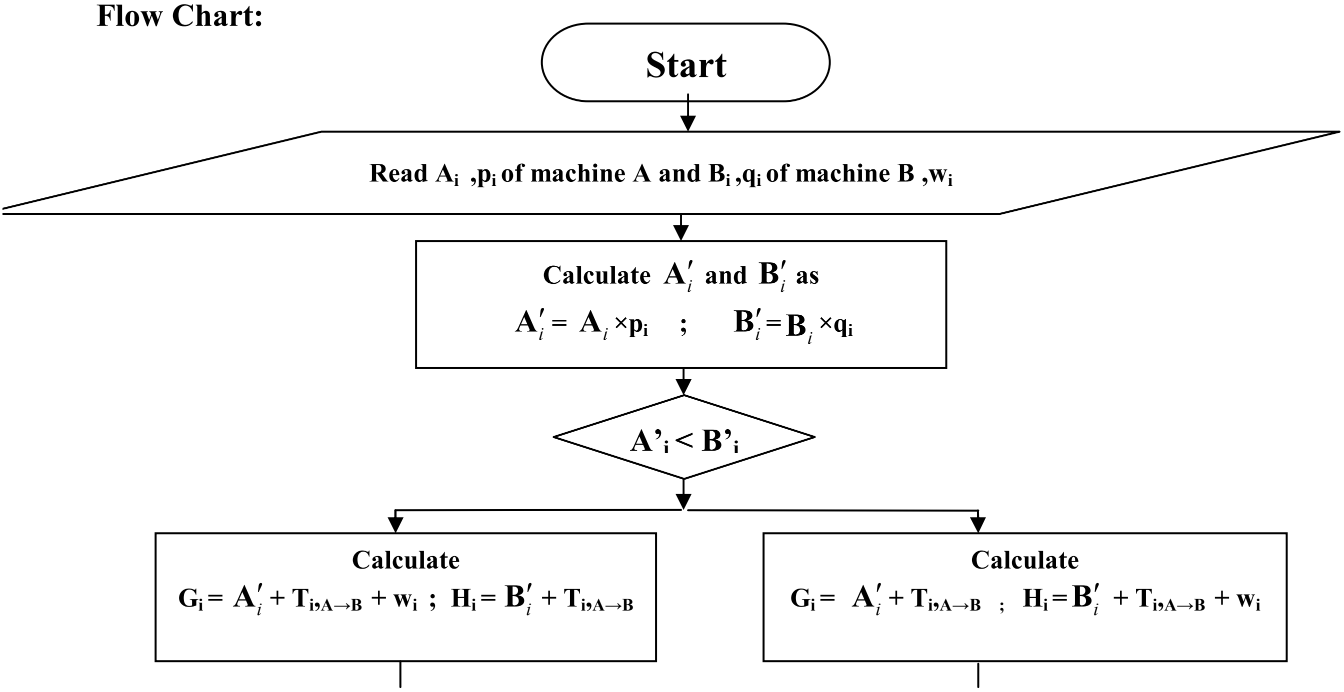
Step11: The augmented ordered sequence $S_{BA} = (S'_1, S'_2)$ gives the optimal or near optimal sequence of jobs for the machine order B to A.

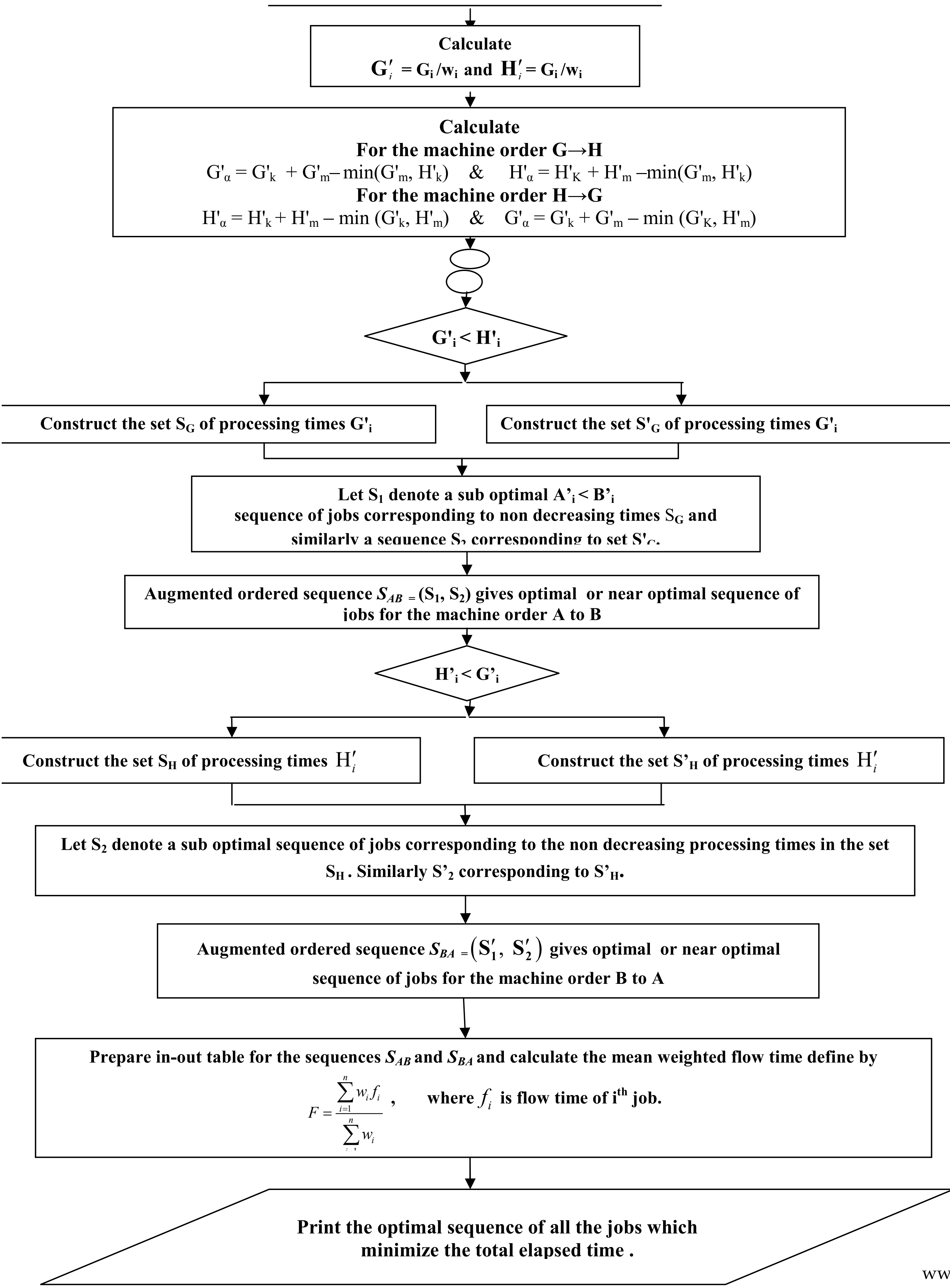
Step12: Prepare in-out table for the sequences $\boldsymbol{S_{AB}}$ and $\boldsymbol{S_{BA}}$ and calculate the mean weighted flow time F define by

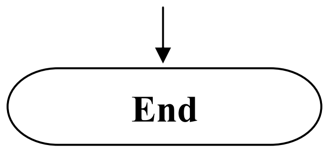
$$F = \frac{\sum_{i=1}^{n} w_i f_i}{\sum_{i=1}^{n} w_i}$$

, where f_i is flow time of i[th] job.

Flow Chart:




Calculate
$G'_i = G_i/w_i$ and $H'_i = G_i/w_i$
Calculate
For the machine order G→H
$G'_\alpha = G'_k + G'_m - \min(G'_m, H'_k)$ & $H'_\alpha = H'_K + H'_m - \min(G'_m, H'_k)$
For the machine order H→G
$H'_\alpha = H'_k + H'_m - \min(G'_k, H'_m)$ & $G'_\alpha = G'_k + G'_m - \min(G'_K, H'_m)$
$G'_i < H'_i$
Construct the set S_G of processing times G'_i
Construct the set S'_G of processing times G'_i
Let S_1 denote a sub optimal $A'_i < B'_i$
sequence of jobs corresponding to non decreasing times S_G and
similarly a sequence S_2 corresponding to set S'_G.
Augmented ordered sequence $S_{AB} = (S_1, S_2)$ gives optimal or near optimal sequence of jobs for the machine order A to B
$H'_i < G'_i$
Construct the set S_H of processing times H'_i
Construct the set S'_H of processing times H'_i
Let S_2 denote a sub optimal sequence of jobs corresponding to the non decreasing processing times in the set S_H. Similarly S'_2 corresponding to S'_H.
Augmented ordered sequence $S_{BA} = (S'_1, S'_2)$ gives optimal or near optimal sequence of jobs for the machine order B to A
Prepare in-out table for the sequences S_{AB} and S_{BA} and calculate the mean weighted flow time define by
$F = \frac{\sum_{i=1}^{n} w_i f_i}{\sum_{i=1}^{n} w_i}$, where f_i is flow time of i^{th} job.
Print the optimal sequence of all the jobs which minimize the total elapsed time .



VII. COMPUTER PROGRAM

```
#include<stdio.h>
#include<fstream.h>
#include<iostream.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#include<iomanip.h>
int noj;
int ajs[100],bjs[100];
float asg[100][3];
float asdg[100][3];
float ash[100][3];
float asdh[100][3];
float atb5[100][3];
float atb8[100][3];
int a1,a2;
int r1,r2,nbr,n=0;
struct ob
{
float aai,api,ti,bbi,bqi,aadi,abdi,agi,ahi,agdi,ahdi,ain,aout,badi,bbdi,bgdi,bhi,bhdi,bin,bout,wi;
int job;
}obv[100];
struct vector
{
float aai,api,ti,bbi,bqi,aadi,abdi,agi,ahi,agdi,ahdi,ain,aout,badi,bbdi,bgdi,bhi,bhdi,bin,bout,wi;
int job;
}v;
class A
{
public:
struct vector tempv[100];
void showmenu()
{
float ain1,aout1,bin1,bout1,ain2,aout2,bin2,bout2;
int a;
re:
clrscr();
cout<<"Select your options:"<<endl<<endl;
cout<<"1. Read Data:"<<endl;
cout<<"2. Show Data:"<<endl;
cout<<"3. Delete All Data:"<<endl;
```

ISSN 2250-3153

```
cout<<"4. Read the values of à:"<<endl;
cout<<"5. Generate Tables:"<<endl;
cout<<"6. Show the Final Result:"<<endl;
cout<<"7. Exit:"<<endl;
cin>>a;
switch(a)
{
case 1:
int jo;
float prot1=0,prot2=0;
clrscr();
cout<<"Enter the No of Jobs";
cin>>jo;
ofstream myf("data.txt",ios::app);
for(int u=1;u<=jo;u++)
{
v.job=u;
cout<<"Enter Ai for  Machine A:";
cin>>v.aai;
cout<<"Enter Pi for  Machine A:";
cin>>v.api;
cout<<"Enter Transportation Time Ti:";
cin>>v.ti;
prot1+=v.api;
cout<<"Enter Bi for  Machine B:";
cin>>v.bbi;
cout<<"Enter Qi for  Machine B:";
cin>>v.bqi;
cout<<"Enter wi ";
cin>>v.wi;

prot2+=v.bqi;
ofstream myf("data.txt",ios::app);
myf<<v.job<<endl;
myf<<v.aai<<endl;
myf<<v.api<<endl;
myf<<v.ti<<endl;
myf<<v.bbi<<endl;
myf<<v.bqi<<endl;
myf<<v.wi<<endl;

}
myf.close();
clrscr();
if(prot1!=1)
cout<<"Entered Pi is invalid\n\nDelete the data and Enter again..";
else if(prot2!=1)
cout<<"Entered Qi is invalid\n\nDelete the data and Enter again..";
else
cout<<"Data is stored";
getch();
```

```
goto re;
break;
case 2:
clrscr();
cout<<"The Data in the file is:\n\nJob\tAi\tPi\tTi\tBi\tQi\tWi\n-------------------------------------------------\n";
ifstream myf1("data.txt");
while(!myf1.eof())
{
myf1>>v.job>>v.aai>>v.api>>v.ti>>v.bbi>>v.bqi>>v.wi;
if(myf1.eof())
break;
cout<<v.job<<"\t";
cout<<v.aai<<"\t";
cout<<v.api<<"\t";
cout<<v.ti<<"\t";
cout<<v.bbi<<"\t";
cout<<v.bqi<<"\t";
cout<<v.wi<<endl;
}
myf1.close();
getch();
goto re;
break;
case 3:
clrscr();
remove("data.txt");
cout<<"Data is Deleted\n\nRun the Program again to Fill new Data\n\n Press any key to EXIT";
getch();
exit(0);
case 4:
clrscr();
rup:
cout<<"Enter the value of à 1: ";
cin>>a1;
cout<<"Enter the value of à 2: ";
cin>>a2;
cout<<"\nData is stored";
goto re;
break;
case 5:

        ain1=ain2=bin1=bin2=aout1=bout1=bout2=aout2=0;
        float tgdk,tgdm,thdk,tgda,thda,tgik,thik,tgim,thim,thdm;
        float btgdk,btgdm,bthdk,btgda,bthda,btgik,bthik,btgim,bthim,bthdm;
        if(a1==0 || a2==0)
        {
        cout<<"Enter the valid values of à & á first\n\n";
        goto rup;
        }
        else
        {
```

```
n=0;
ifstream myf2("data.txt");
noj=0;
        clrscr();
        cout<<"The given values are:\n---------------------------------------------------\n";
        cout<<"JOB\t Ai \t Pi \t Ti \t Bi \t Qi\n";

        while(!myf2.eof())
        {
                myf2>>v.job>>v.aai>>v.api>>v.ti>>v.bbi>>v.bqi>>v.wi;

                if(myf2.eof())
                        break;
        tempv[n]=v;
        cout<<v.job<<"\t"<<v.aai<<"\t"<<v.api<<"\t"<<v.ti<<"\t"<<v.bbi<<"\t"<<v.bqi<<endl;
        n++;
        noj++;
        }
myf2.close();
cout<<"Table No: 3\n------------------\n";
for(int a=0;a<n;a++)
{
tempv[a].aadi=tempv[a].aai*tempv[a].api;
tempv[a].badi=tempv[a].bbi*tempv[a].bqi;
tempv[a].agi=tempv[a].aadi+tempv[a].ti;
tempv[a].ahi=tempv[a].badi+tempv[a].ti;

cout<<setprecision(2)<<tempv[a].aadi<<"\t"<<tempv[a].badi<<endl;

}
cout<<"Table No: 4\n----------------------\n";
for(a=0;a<n;a++)
{
if(tempv[a].agi<tempv[a].ahi)
tempv[a].agi+=tempv[a].wi;
else
tempv[a].ahi+=tempv[a].wi;
if(tempv[a].job==a1)
{
 tgdk=tempv[a].agi/tempv[a].wi;
 thdk=tempv[a].ahi/tempv[a].wi;
}
if(tempv[a].job==a2)
{
 tgdm=tempv[a].agi/tempv[a].wi;
 thdm=tempv[a].ahi/tempv[a].wi;
}

cout<<setprecision(2)<<tempv[a].agi<<"\t"<<tempv[a].ahi<<endl;
}
 // cout<<tgdk<<tgdm<<thdk<<thdm;
```

```
//Step 5;
        int i=0;
        float toa1,toa2;
        int nosg=0,nosdg=0;
        int nosh=0,nosdh=0;
        for(a=0;a<n;a++)
        {
        tempv[a].agi/=tempv[a].wi;
        tempv[a].ahi/=tempv[a].wi;

        if(a1==tempv[a].job || a2==tempv[a].job)
        cout<<"";
        else
        {
        atb5[i][0]=tempv[a].job;
        atb5[i][1]=tempv[a].agi;
        atb5[i][2]=tempv[a].ahi;

        atb8[i][0]=tempv[a].job;
        atb8[i][2]=tempv[a].agi;
        atb8[i][1]=tempv[a].ahi;


        if(tempv[a].agi>tempv[a].ahi)
        {
        asg[nosg][0]=tempv[a].agi;
        asg[nosg][1]=tempv[a].job;
        nosg++;
        }
        else
        {
        asdg[nosdg][0]=tempv[a].agi;
        asdg[nosdg][1]=tempv[a].job;
        nosdg++;
        }
//b->a
        if(tempv[a].agi<tempv[a].ahi)
        {
        ash[nosh][0]=tempv[a].agi;
        ash[nosh][1]=tempv[a].job;
        nosh++;
        }
        else
        {
        asdh[nosdh][0]=tempv[a].agi;
        asdh[nosdh][1]=tempv[a].job;
        nosdh++;
        }
```

```
		i++;
		}
		}
		atb5[i][0]=-1;
		atb5[i][1]=tgda;
		atb5[i][2]=thda;
		atb8[i][0]=-2;
		atb8[i][1]=bthda;
		atb8[i][2]=btgda;
		i++;
		getch();
		clrscr();
//		cout<<btgdk<<"+ "<<btgdm<<"+ "<<bthdk<<"+ "<<bthdm<<"+ "<<endl;

		tgda=tgdk+tgdm-((thdk<tgdm)?thdk:tgdm);
		thda=thdk+thdm-((thdk<tgdm)?thdk:tgdm);
		bthda=thdk+thdm-((thdm<tgdk)?thdm:tgdk);
		btgda=tgdk+tgdm-((thdm<tgdk)?thdm:tgdk);
//		cout<<tgda<<" "<<thda<<" "<<bthda<<" "<<btgda<<" "<<endl;

//		cout<<bthda<<" -*- "<<btgda<<endl;
		cout<<"G\'à ="<<tgda<<"\tH\'à = "<<thda<<endl;
		cout<<"G\'à ="<<btgda<<"\tH\'à = "<<bthda<<endl;

		if(bthda<btgda)
		{
		 asdh[nosdh][0]=btgda;
		 asdh[nosdh][1]=-2;
		nosdh++;
		}
		else
		{
		 ash[nosh][0]=bthda;
		 ash[nosh][1]=-2;
		 nosh++;
		}
		if(tgda<thda)
		{
		 asdg[nosdg][0]=thda;
		 asdg[nosdg][1]=-1;
		 nosdg++;
		}
		else
		{
		 asg[nosg][0]=tgda;
		 asg[nosg][1]=-1;
		 nosg++;
		}

		cout<<"\nTable no 5:\n----------------\nJob\tG\'i\tH\'i\n";
		for(a=0;a<i;a++)
```

```
        {
        if(atb5[a][0]==-1)
        cout<<"à"<<"\t"<<setprecision(2)<<atb5[a][1]<<" \t "<<atb5[a][2]<<" \n";
        else
        cout<<atb5[a][0]<<" \t "<<atb5[a][1]<<" \t "<<atb5[a][2]<<" \n";
        }
        cout<<"\nTable no 10:\n----------------\nJob\tH\'i\tG\'i\n";
        for(a=0;a<i;a++)
        {
        if(atb8[a][0]==-2)
        cout<<"à"<<"\t"<<setprecision(2)<<atb8[a][1]<<" \t "<<atb8[a][2]<<" \n";
        else
        cout<<atb8[a][0]<<" \t "<<atb8[a][1]<<" \t "<<atb8[a][2]<<" \n";
        }
        getch();
        clrscr();
        float tp1,tp2;
        int b;
        for(a=0;a<nosdg-1;a++)
                for(b=0;b<nosdg-1-a;b++)
                        {
                        if(asdg[b][0]>asdg[b+1][0])
                                {
                                tp1=asdg[b][0];
                                tp2=asdg[b][1];
                                asdg[b][0]=asdg[b+1][0];
                                asdg[b][1]=asdg[b+1][1];
                                asdg[b+1][0]=tp1;
                                asdg[b+1][1]=tp2;
                                }
                        }
        for(a=0;a<nosg-1;a++)
                for(b=0;b<nosg-1-a;b++)
                        {
                        if(asg[b][0]>asg[b+1][0])
                                {
                                tp1=asg[b][0];
                                tp2=asg[b][1];
                                asg[b][0]=asg[b+1][0];
                                asg[b][1]=asg[b+1][1];
                                asg[b+1][0]=tp1;
                                asg[b+1][1]=tp2;
                                }
                        }
    //sorting b->a
        for(a=0;a<nosdh-1;a++)
                for(b=0;b<nosdh-1-a;b++)
                        {
                        if(asdh[b][0]>asdh[b+1][0])
                                {
                                tp1=asdh[b][0];
```

ISSN 2250-3153

```
                                        tp2=asdh[b][1];
                                        asdh[b][0]=asdh[b+1][0];
                                        asdh[b][1]=asdh[b+1][1];
                                        asdh[b+1][0]=tp1;
                                        asdh[b+1][1]=tp2;
                                        }
                                }
                for(a=0;a<nosh-1;a++)
                        for(b=0;b<nosh-1-a;b++)
                                {
                                 if(ash[b][0]>ash[b+1][0])
                                        {
                                        tp1=ash[b][0];
                                        tp2=ash[b][1];
                                        ash[b][0]=ash[b+1][0];
                                        ash[b][1]=ash[b+1][1];
                                        ash[b+1][0]=tp1;
                                        ash[b+1][1]=tp2;
                                        }
                                }

                i=0;
                for(a=0;a<nosdg;a++)
                {
                 if(asdg[a][1]==-1)
                  {
                   ajs[i]=a1;
                   i++;
                   ajs[i]=a2;
                   i++;
                  }
                 else
                  {
                 ajs[i]=(int)asdg[a][1];
                 i++;
                  }
            }
            for(a=0;a<nosg;a++)
                {
                 if(asg[a][1]==-1)
                  {
                   ajs[i]=a1;
                    i++;
                   ajs[i]=a2;
                   i++;
                  }
                 else
                  {
                 ajs[i]=(int)asg[a][1];
//               cout<<asg[a];
                 i++;
```

```
            }
        }
        for(a=0;a<i;a++)
        cout<<ajs[a]<<",";

            ain1=0;
            aout1=0;
            bin1=0;
            bout1=0;
            cout<<endl;
    float tti;
    clrscr();
    cout<<"In-Out table for the Order A->B\nJob\tA(in)\tA(out)\tB(in)\tB(out)\tWi\n-----------------------------------------------
\n";
    float tbadi,fl;
    float twi;
    float timt=0,btimt=0;
    float tottim=0,btottim=0;
            float temp=0;
             for(n=0;n<i;n++)
        {
                    for(a=0;a<i;a++)
                            if(tempv[a].job==ajs[n])
                                    {
                                    aout1=ain1+tempv[a].aadi;
                                    tti=tempv[a].ti;
                                    fl=tempv[a].badi;
                                    twi=tempv[a].wi;
                                    }
                            bin1=aout1+tti;
                            if(bin1<temp)
                            bin1=temp;
                            bout1=bin1+fl;
                            timt+=(bout1-ain1)*twi;
        cout<<setprecision(2)<<ajs[n]<<" \t "<<ain1<<"\t "<<aout1<<" \t "<<bin1<<"\t "<<bout1<<"\t"<<twi<<endl;
                            tottim+=twi;
        ain1=aout1;
        bin1=bout1;
        temp=bout1;
        }
            cout<<"Total Weighted Flow Time is: "<<timt/tottim;
            //B -> A
            i=0;
            for(a=0;a<nosdh;a++)
            {
             if(asdh[a][1]==-2)
              {
              bjs[i]=a1;
              i++;
              bjs[i]=a2;
              i++;
```

```
            }
            else
            {
            bjs[i]=(int)asdh[a][1];
            i++;
            }
        }
        for(a=0;a<nosh;a++)
            {
            if(ash[a][1]==-2)
            {
             bjs[i]=a1;
              i++;
             bjs[i]=a2;
             i++;
            }
            else
            {
            bjs[i]=(int)ash[a][1];
            i++;
            }
        }
        temp=0;
//      cout<<"---------------"<<endl;
//      for(a=0;a<i;a++)
//      cout<<bjs[a]<<",";
//      cout<<endl;
            ain2=0;
            aout2=0;
            bin2=0;
            bout2=0;
cout<<"\n\nIn-Out table for the Order B->A \nJob\tB(in)\tB(out)\tA(in)\tA(out)\tWi\n-----------------------------------------------
--\n";
for(n=0;n<i;n++)
        {
                for(a=0;a<i;a++)
                                if(tempv[a].job==bjs[n])
                                        {
//                                      aout2=ain2+tempv[a].aadi;
//                                      tti=tempv[a].ti;
//                                      fl=tempv[a].badi;
                                        aout2=ain2+tempv[a].badi;
                                        tti=tempv[a].ti;
                                        fl=tempv[a].aadi;
                                        twi=tempv[a].wi;
                                        }
                                bin2=aout2+tti;
                                if(bin2<temp)
                                bin2=temp;
                                bout2=bin2+fl;
                                btimt+=(bout2-ain2)*twi;
```

```
                    btottim+=twi;
//                  cout<<twi<<endl;
    cout<<setprecision(2)<<bjs[n]<<" \t "<<ain2<<"\t "<<aout2<<" \t "<<bin2<<"\t "<<bout2<<"\t"<<twi<<endl;
    ain2=aout2;
    bin2=bout2;
    temp=bout2;
    }
//     cout<<btimt<<"  "<<btottim<<endl;
        cout<<"Total Weighted Flow Time is: "<<btimt/btottim;

        }
getch();
goto re1;
break;
case 6:
re1:
        clrscr();
cout<<"\n\n\n\nThe Total Elapsed time for the the sequence (";
    for(a=0;a<noj;a++)
    cout<<ajs[a]<<",";
cout<<")  when the order is A-> B is: "<<bout1<<endl;

cout<<"\nThe Total Elapsed time for the the sequence (";
    for(a=0;a<noj;a++)
    cout<<bjs[a]<<",";
cout<<")  when the order is B-> A is: "<<bout2<<endl;
cout<<"\nSo.....\n\n";
if(bout1<bout2)
{
cout<<"\nHence the optimal sequence of all the job which minimise the total elapse time of each machine is (";
        for(a=0;a<noj;a++)
    cout<<ajs[a]<<",";
cout<<") for the order  A -> B .";
}
else
{
cout<<"\nHence the optimal sequence of all the job which minimize the total elapse time of each machine is (";
        for(a=0;a<noj;a++)
    cout<<bjs[a]<<",";
cout<<") for the order  B -> A .";

}

cout<<"\n\n\n Press Any key to exit..";
getch();
break;
}//end switch
}

};
void main()
```

```
    {
    clrscr();
    A ob;
    noj=0;
    ob.showmenu();
}
```

Numerical Illustration:

Consider 5 jobs, 2 machines open shop problem with processing time A_i and B_i associated with their respective probabilities p_i and q_i. Transportation time $T_{i,A\to B}$ or $T_{i,B\to A}$ and weights of each job are given in the following table. The jobs 3 and 5 are processed as a group job (3, 5).

Jobs	**Machine A**		**$T_{i,A\to B}$ or $T_{i,B\to A}$**	**Machine B**		**w_i**
I	A_i	p_i		B_i	q_i	
1	24	0.2	4	22	0.3	5
2	20	0.2	2	16	0.2	4
3	30	0.3	6	13	0.2	3
4	15	0.1	5	10	0.1	2
5	25	0.2	3	12	0.2	3

Tableau-2

Our objective is to find the optimal or near optimal sequence which minimizes the total elapsed time and mean weighted flow time.

Solution:-

As per Step1: Define expected processing time A'_i and B'_i on machine A and B respectively by following formulae:-

(a) $A'_i = A_i \times p_i$

(b) $B'_i = B_i \times q_i$

Jobs	A'_i	$T_{i,A\to B}$ or $T_{i,B\to A}$	B'_i	w_i
1	4.8	4	6.6	5
2	4.0	2	3.2	4
3	9.0	6	2.6	3
4	1.5	5	1.0	2
5	5.0	3	2.4	3

Tableau-3

As per Step2: Define two fictitious machines G and H with their processing times G_i and H_i using:

(a) If $\min(A'_i, B'_i) = A'_i$ then calculate $G_i = A'_i + T_{i,A\to B} + w_i$ and $H_i = B'_i + T_{i,A\to B}$

(b) If $\min(A'_i, B'_i) = B'_i$ then calculate $G_i = A'_i + T_{i,A\to B}$ and $H_i = B'_i + T_{i,A\to B} + w_i$

Jobs	G_i	H_i	w_i
1	13.8	10.6	5
2	6.0	9.2	4
3	15.0	11.6	3

4	6.5	8.0	2
5	8.0	8.4	3

Tableau-4

As per Step3: Calculate G'_i and H'_i where

(a) $G'_i = G_i / w_i$

(b) $H'_i = H_i / w_i$

Jobs	G'_i	H'_i
1	2.76	2.12
2	1.5	2.3
3	5.0	3.86
4	3.25	4.0
5	2.66	2.8

Tableau-5

As per Step4: For the machine order G→H

Calculate the expected processing time for the equivalent job α = (3, 5) on the fictitious machines G and H using:-

(a) $G'_\alpha = G'_k + G'_m - \min(H'_k, G'_m)$
$G'_\alpha = 5.0 + 2.66 - \min(3.86, 2.66)$
$= 5.0 + 2.66 - 2.66 = 5.0$

(b) $H'_\alpha = H'_k + H'_m - \min(H'_k, G'_m)$
$H'_\alpha = 3.86 + 2.8 - \min(3.86, 2.66)$
$= 6.66 - 2.66 = 4.0$

As per Step5: Define a new reduced problem with processing times G'_i and H'_i as define in step 3 & 4

Jobs	G'_i	H'_i
1	2.76	2.12
2	1.5	2.3
α	5.0	4.0
4	3.25	4.0

Tableau-6

As per Step6: Construct a set S_G and S'_G
$S_G = \{1.5, 3.25\}$; $S'_G = \{2.76, 5.0\}$

As per Step7: $S_1 = \{2, 4\}$; $S_2 = \{1,\alpha\}$

As per Step8: Augmented ordered sequence $S_{AB} = \{S_1, S_2\}$ i.e. {2,4,1,3,5} gives optimal/near optimal sequence for the machine order A to B

For the machine order H→G; As per step1

Jobs	B'_i	$T_{i,A\to B}$ or $T_{i,B\to A}$	A	w_i
1	6.6	4	4.8	5
2	3.2	2	4.0	4
3	2.6	6	9.0	3
4	1.0	5	1.5	2
5	2.4	3	5.0	3

Tableau-7

As per Step2: Define two fictitious machines H and G with processing times H_i and G_i for job i on machines H and G respectively.

Jobs	H_i	G_i	w_i
1	10.6	13.8	5
2	9.2	6.0	4
3	11.6	15.0	3
4	8.0	6.5	2
5	8.4	8.0	3

Tableau-8

As per Step3:

Jobs	H'_i	G'_i
1	2.12	2.76
2	2.3	1.5
3	3.86	5.0
4	4.0	3.25
5	2.8	2.66

Tableau-9

As per Step4: Calculate processing time for equivalent job α = (3,5) as define in step3&4

$H'_\alpha = H'_k + H'_m - \min(H'_m, G'_k)$
$=3.86 + 2.8 - \min(5.0, 2.8)$
$= 3.86+ 2.8 -2.8 = 3.86$

$G'_\alpha = G'_k + G'_m - \min(H'_m, G'_k)$
$=5.0 + 2.66 - \min(5.0, 2.8)$
$=7.66 -2.8 =4.86$

As per Step8: For the machine order H→G construct the set S_H and S'_H:

Jobs	H'_i	G'_i
1	2.12	2.76
2	2.3	1.5
α	3.86	4.86
4	4.0	3.25

Tableau-10

S_H = {2.12,3.86} ; S'_H = {2.3,4.0}

As per Step9: S'_1 = {1,α } ; S'_2 = {2,4}

As per Step10: Augmented ordered sequence S_{BA} = { S'_1, S'_2 } i.e. {1,3,5,2,4} gives the optimal / near optimal sequence for machine order B to A

Now calculate the total production time for the sequence S_{AB} and S_{BA}

As per Step11: In – Out table for the sequence S_{AB}

Jobs i	Machine A	Machine B
	In Out	In Out
2	0.0 - 4.0	6.0 - 9.2
4	4.0 - 5.5	10.5 - 11.5
1	5.5 - 10.3	14.3 - 20.9
3	10.3 - 19.3	25.3 - 27.9
5	19.3 - 24.3	27.9 - 30.3

Tableau-11

The mean weighted flow time for the machine order A to B

$$= \frac{[9.2\times 4+(11.5 - 4.0)\times 2+(20.9 - 5.5)\times 5+(27.9 - 10.3)\times 3+(30.3-19.3)\times 3]}{4+2+5+3+3}$$

=12.62 hrs.

In – Out table for the sequence S_{BA}

Jobs i	Machine B	Machine A
	In Out	In Out
1	0.0 - 6.6	10.6 - 15.4
3	6.6 - 9.2	15.4 - 24.4
5	9. 2 - 11.6	24.4 - 29.4
2	11.6 - 14.8	29.4 - 33.4
4	14.8 - 15.8	33.4 - 34.9

Tableau-12

The mean weighted flow time for the machine order B to A

$$=\frac{\mathbf{15.4\times5+(24.4-6.6)\times3+(29.4-9.2)\times3+(33.4-11.6)\times4+(34.9-14.8)\times2}}{5+3+3+4+2}$$

=18.72 hrs

The total expected elapsed time when the order is from A to B for the sequence (2, 4, 1, 3, 5) is 30.3. units and for the sequence (1, 3, 5, 2, 4) is 34.9 units when order is B to A. Hence the optimal sequence of all the jobs which minimize the total elapsed time of each machine is (2, 4, 1, 3, 5) for the order A→B.

VIII. Remarks

[1] If the probabilities are not associated with processing time then the results tally with[9].

[2] If probabilities are associate with processing time then the results tally with[8].

[3] If both probabilities and transportation time are not included then results tally with[9].

[4] Equivalent job formulation is associative and non commutative in nature.

[5] The study on n×2 open shop scheduling may be further extended by including parameter such as break down interval, etc.

[6] The study may further be extended for 3 machines, also by considering various parameters such as Transportation time, break down interval, weightage of jobs etc.

References

[1] **Anup(2002)** , "On two machine flow shop problem in which processing time assume Probabilities and there exists equivalent for an ordered job block", *JISSO,VOL.XXIII,* No.1-4,pp.41-44.

[2] **Bagga,P.C.(1969)**, "Sequencing in rental situations", *Canadian operation Research Society* 7,pp.152-153.

[3] **Campbell,H.A, Dular,R.A. and Smith, M.L(1970)**, "A heuristic algorithm for the n-job, m-machine sequencing problem", *Management science*, 16,B630-B637.

[4] **Chandrashekharan Rajendaran(1992)**, "Two stage flow shop scheduling Problem with Bicriteria",*O.R.Soc*.,Vol.43, No9,pp.871-84.

[5] **Gupta Deepak & Singh.T.P.(2004)** , " Optimal two stage production schedule with group job -restrictions having set up times separated from processing time, associated with probabilities", *Des ERA JMS* VOL.1,pp 53-70.

[6] **Gupta, D, Sharma, S. and Seema(2011)**, "Bicriteria in n*3 flow shop scheduling under specified rental policy, processing time associated with probabilities including transportation time, weightage and job block criteria.", *International Academic Research*,Vol.11,No.3,pp.66-83.

[7] **Gupta, D, Singla, P. and Sharma, S.(2011)**, "Heuristic Approach for n-jobs, 3-Machines flow shop scheduling problem, processing time associated with probabilities involving transportation time, break –down interval, weightage of jobs and job block criteria.", *Mathematical Theory and Modeling,*Vol.1,No.1,pp.30-36.

[8] **Gupta Deepak, Singh T.P et.al(2005)**, "On job block open shop scheduling ,the processing time associated with probability."*J. Indian Soc. Stat. Opers. Res.* Vol.XXVI,No.1-4.pp. 91-96.

[9] **Harbans Lal and Maggu.P.L. (1989)**, "On job block open shop scheduling problem", *PAMS,* Vol.XXIX, pp. 45-51.

[10] **Ignall,E. and Schrage L.(1965)**, "Application on Branch and Bound technique to some flowshop scheduling problems, *Operation Research*,Vol.13,pp.400-412.

[11] **Johnson,S.M.(1954)**, "Optimal two and three stage production schedule with set up times included" *Nay Res Log Quart* Vol 1,pp.61-68.

[12] **Maggu,P.L.and Das G.(1977)**, "Equivalent jobs for job block in job sequencing, *Opsearch*, Vol 14,No.4,pp.277-281.

[13] **Singh, T.P.(1985)**, "On n*2 open shop problem involving job block, Transportation time and Break down Machinetimes",*PAMS*,Vol.XX1,pp.12.

[14] **V.A.Strusevich(1999)**, "A heuristic for the two–machine open shop heduling problem with transportation times", *Discrete Applied Mathematics* 93,pp.287-304

[15] **Yoshida and Hitomi(1979)**, "Optimal two stage production schedule with set up time Separated", *AIIE Transactions*.Vol.II,pp261-263.

Authors

First Author – Deepak Gupta, Prof.& Head, Department of Mathematics M.M. University, Mullana, Ambala, India, Email: guptadeepak2003@yahoo.co.in
Second Author – Renuka, Department of Mathematics, M.M. University, Mullana, Ambala, India, Email: renukajain7@gmail.com
Third Author – Payal Singla, Department of Mathematics, M.M. University, Mullana, Ambala, India, Email: payalsingla86@gmail.com