International Journal of Mathematical Archive-3(5), 2012, 1859-1872
IJMA Available online through www.ijma.info ISSN 2229 – 5046

OPTIMAL TWO STAGE OPEN SHOP SCHEDULING, PROCESSING TIME ASSOCIATED WITH PROBABILITIES INCLUDING TRANSPORTATION TIME AND JOB BLOCK CRITERIA

Deepak Gupta
Prof. & Head, Department of Mathematics,
Maharishi Markandeshwar University, Mullana, Haryana, India

Renuka
Research Scholar, Department of Mathematics
Maharishi Markandeshwar University, Mullana, Haryana, India

Payal Singla*
Research Scholar, Department of Mathematics,
Maharishi Markandeshwar University, Mullana, Haryana, India

(Received on: 27-04-12; Accepted on: 14-05-12)

ABSTRACT

The present paper is an attempt through heuristic method to obtain the optimal sequence for n jobs two stage open shop problem in which the processing times are associated with probabilities. The concepts of equivalent job for a job block and transportation time from one machine to another are also taken into consideration. The algorithm developed in this paper is very simple and easy to understand. A computer program followed by numerical illustration is given to clarify the algorithm.

Keywords: *Open shop problem, Equivalent job, Elapsed time, Job block, Transportation time.*

1. INTRODUCTION

Job sequence and scheduling problem has a wider application in the industrial areas, which minimizing the manufacturing cost and time or optimizes effectiveness by selecting the most suitable sequence. The earliest results for flow shop scheduling problem was introduced by Johnson [7] in order to minimize the total idle time of the machines. Further the work was developed by Jackson.J.R.[8], Maggu and Das[9], Harbans[6], Yoshida and Hitomi[13], M.Deel'sAmico[10], D.Rebaine[2], Anup[1], Gupta and Singh [3,5,11], Rastogi and Singh [12], Gupta D; Sharma S.[4] by introducing different parameters such as transportation time ,break down interval, weightage etc. Maggu & Das[9] introduced the concept of equivalent job for a job block in which the priority to one job over another was taken into account. Anup[1] extended the study by associating probabilities with processing time as the processing time are always not exact. Open shop scheduling differ from flow shop in the sense that there are no restrictions placed on the order of the machines i.e. operations can be performed in any order A to B or B to A and not known in the advance. Gupta and Singh [5] studied the n*2 open shop problem to minimize the total idle time of the machines in which the probabilities associated with processing time including job block criteria. In this paper we have extended the study made by Gupta and Singh [5] by including the concept of transportation time as there are many situations where the transportation times are quite significant and can not be simply neglected. Thus the problem in the present paper has wider and practically more applicable and provides suitable results. An algorithm has been developed to minimize the maximum completion time (makespan). The algorithm is demonstrated through numerical example.

2. PROBLEM FORMULATION

Let n jobs 1, 2, 3,…..n be processed through two machines A and B. Let A_i be the processing time of i^{th} job (i=1,2,3….n) on machine A and p_i be the probabilities associated with processing time A_i such that $0 \le p_i \le 1$ and $\sum p_i=1$ and B_i be the processing time of i^{th} job on machine B and q_i be the probabilities associated with processing time B_i such that $0 \le q_i \le 1$ and $\sum q_i =1$.Let $T_{i,1\to 2}$be the transportation time of i^{th} job from 1^{st} machines to the 2^{nd} machine which is same as transportation time from 2^{nd} to 1^{st} i.e. $T_{i,1\to 2}$ is same as$T_{i,2\to 1}$.

Let α be an equivalent job for 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.

Corresponding author: *Payal Singla**
Research Scholar, Department of Mathematics, Maharishi Markandeshwar University, Mullana, Haryana, India

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

Jobs	Machine A		$T_{i,1\to2}$	Machine B	
i	A_i	p_i	$T_{i,2\to1}$	B_i	q_i
1	A_1	p_1	$T_{1,1\to2}$	B_1	q_1
2	A_2	p_2	$T_{2,1\to2}$	B_2	q_2
3	A_3	p_3	$T_{3,1\to2}$	B_3	q_3
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
n	A_n	p_n	$T_{n,1\to2}$	B_n	q_n

Tableau-1

Our objective is to find the optimal schedule of all the jobs which minimize the total elapsed time of each machine, expected processing time may be in the terms of cost.

3. ASSUMPTIONS

1. Two jobs can't be processed on a single machine at a time.
2. Transportation time from machine first to second and second to first are same.
3. Priority is given to the i_1 job over $i_2 \ldots . i_k$ in job block $(i_1, i_2 \ldots\ldots\ldots i_k)$.
4. The second job will be processed on a machine when first job is completed.
5. Transporting device is always available.
6. $\sum p_i=1$, $\sum q_i=1$, $0\leq p_i\leq 1$, $0\leq q_i\leq 1$

4. ALGORITHM

The heuristic algorithm for the problem discussed here, is as follow as:

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

(a) $A'_i = A_i * p_i$

(b) $B'_i = B_i * q_i$

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

$G_i = A'_i + T_{i,1\to2}$; $Hi = B'i + Ti,2\to1$

Step3: Define expected processing time for the equivalent job $\alpha = (k,m)$ as follows:-

For the order $G \to H$

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

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

For the order $H\to G$

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

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

Step4: Represent a new reduced problem with processing time G'_i and H'_i as per step 2&3

Step5: For the order $G\to H$

Construct a set S_G of all the processing time G'_I where $G'_i \leq H'_i$ and S'_G of all the processing time G'_i where $G'_i \geq H'_i$.

Step6: Let S_1 denote a sub optimal sequence of jobs corresponding to non decreasing times SG and similarly a sequence S_2 corresponding to set S'_G.

Step7: The augmented ordered sequence (S_1, S_2) gives optimal or near optimal sequence for processing the jobs on machine A for the given problem.

Step8: For the order $H\to G$

Construct the set S_H and S'_H of processing times H'_i Where $H'_i \leq G'_i$ and of processing times H'_i where

$H'_i \geq G'_i$ respectively .

Step9: Let S_2 denote a sub optimal sequence of jobs corresponding to the non decreasing processing times in the set S_H. similarlly S'_2 corresponding to S'_H.

Step10: The augmented ordered sequence (S'_1, S'_2) gives the optimal or near optimal sequence for processing the jobs on the machine B for the given problem.

Step11: Prepare in-out tables of sequences (S_1, S_2) and (S'_1, S'_2) in the order A →B and B→A respectively by including transportation time.

5. 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;
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;
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;
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 Transfer 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;
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.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\t\n--------------------------------------------------\n";
ifstream myf1("data.txt");
while(!myf1.eof())
{
myf1>>v.job>>v.aai>>v.api>>v.ti>>v.bbi>>v.bqi;
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<<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;

                        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;

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

```
            {
            cout<<setprecision(2)<<tempv[a].agi<<"\t"<<tempv[a].ahi<<endl;
            }
// cout<<tgdk<<tgdm<<thdk<<thdm;
            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<<bthda<<" -*- "<<btgda<<endl;
            cout<<"G\'à ="<<tgda<<"\tH\'à = "<<thda<<endl;

//Step 5;
            int i=0;
            float toa1,toa2;
            int nosg=0,nosdg=0;
            int nosh=0,nosdh=0;
            for(a=0;a<n;a++)
            {
            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();
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 "<<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 8:\n----------------\nJob\tH\'i\tG\'i\n";
for(a=0;a<i;a++)
{
if(atb8[a][0]==-2)
cout<<"à"<<" \t "<<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];
                                  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)\n-------------------------------------------------
\n";
float tbadi,fl;
           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;

                              }
                        bin1=aout1+tti;
                        if(bin1<temp)
                        bin1=temp;
                        bout1=bin1+fl;
      cout<<setprecision(2)<<ajs[n]<<" \t "<<ain1<<"\t "<<aout1<<" \t "<<bin1<<"\t "<<bout1<<endl;
      ain1=aout1;
      bin1=bout1;
      temp=bout1;
      }
           //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)\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;

                                          }
                              bin2=aout2+tti;
                              if(bin2<temp)
                              bin2=temp;
                              bout2=bin2+fl;
      cout<<setprecision(2)<<bjs[n]<<" \t "<<ain2<<"\t "<<aout2<<" \t "<<bin2<<"\t "<<bout2<<endl;
      ain2=aout2;
      bin2=bout2;
      temp=bout2;
      }
            }

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 minimise 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 and transportation time $T_{i,1\to 2}$ or $T_{i,2\to 1}$ given in the following table. The jobs 2 and 4 are processed as a group job (2,4).

jobs	Machine A		$T_{i,1\to 2}$	Machine B	
i	A_i	P_i	$T_{i,2\to 1}$	B_i	q_i
1	21	0.2	5	14	0.1
2	16	0.1	3	15	0.2
3	13	0.3	4	18	0.3
4	17	0.2	2	10	0.1
5	11	0.2	1	18	0.3

Tableau-2

Our objective is to find the optimal/ near optimal sequence which minimizes the total elapsed time.

Solution:-

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

(a) $A'_i = A_i * p_i$

(b) $B'_i = B_i * q_i$

Jobs	A′i	B'_i
1	4.2	1.4
2	1.6	3.0
3	3.9	5.4
4	3.4	1.0
5	2.2	5.4

Tableau-3

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

$G_i = A'_i + T_{i,1\to 2}$; $H_i = B'_i + T_{i,2\to 1}$

Jobs	G_i	H_i
1	9.2	6.4

2	4.6	6.0
3	7.9	9.4
4	5.4	3.0
5	3.2	6.4

Tableau-4

As per Step3: For the order G→H

Define processing time for the equivalent job $\alpha = (2, 4)$ using:

(a) $G'_\alpha = G'_k + G'_m - \min(H'_k, G'_m)$
$G'_\alpha = 4.6 + 5.4 - \min(5.4, 6.0) = 10.0 - 5.4 = 4.6$

(b) $H'_\alpha = H'_k + H'_m - \min(H'_k, G'_m)$
$H'_\alpha = 6.0 + 3.0 - \min(5.4, 6.0) = 9.0 - 5.4 = 3.6$

As per Step4: Reduced processing times G'_i and H'_i are as follows :

Jobs	G'_i	H'_i
1	9.2	6.4
α	4.6	3.6
3	7.9	9.4
5	3.2	6.4

Tableau-5

As per Step5: Construct a set S_G and S'_G $S_G = \{7.9, 3.2\}$ $S'_G = \{9.2, 4.6\}$

As per Step6: $S_1 = \{5, 3\}$ $S'_1 = \{\alpha, 1\}$

As per Step7: Augmented ordered sequence = {5, 3, α, 1} i.e. {5 ,3, 2, 4, 1}

For the order H→G : As per step1

Jobs	B'_i	$T_{i,2\to1}$	A'_i
1	1.4	5	4.2
2	3.0	3	1.6
3	5.4	4	3.9
4	1.0	2	3.4
5	5.4	1	2.2

Tableau-6

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 as:
$H_i = B'_i + T_{i,2\to1}$; $G_i = A'_i + T_{i,1\to2}$

Jobs	H_i	G_i
1	6.4	9.2
2	6.0	4.6
3	9.4	7.9
4	3.0	5.4
5	6.4	3.2

Tableau-7

As per Step3: Define processing time for equivalent job $\alpha = (2,4)$
$H'_\alpha = H'_k + H'_m - \min(H'_m, G'_k)$
$= 6.0 + 3.0 - \min(4.6, 3.0)$
$= 9.0 - 3.0 = 6.0$

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

=4.6 + 5.4 –min (4.6, 3.0) =10.0 -3.0 =7.0

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

Jobs	H'_i	G'_i
1	6.4	9.2
α	6.0	7.0
3	9.4	7.9
5	6.4	3.2

Tableau-8

S_H = {6.4,6.0} S'_H = {9.4,6.4}

As per Step9: S_2 = {α, 1}S'_2 = {5,3}

As per Step10: Augmented ordered sequence is = {α,1,5,3} i.e. {2,4,1,5,3}

This gives the exact or near optimal sequence of jobs when processed from A to B

Now we calculate the total production times on machine A and B for the sequence (5,3,2,4,1) and(2,4,1,5,3)respectively.

As per Step11: In – Out table for the order A→B

For the order A→B		
Jobs	Machine A	Machine B
	In Out	In Out
5	0 - 2.2	3.2 - 8.6
3	2.2 - 6.1	10.1 - 15.5
2	6.1 - 7.7	15.5 - 18.5
4	7.7 - 11.1	18.5 - 19.5
1	11.1 - 15.3	20.3 - 21.7

Tableau-9

In – Out table for the order B→A

For the order B→A		
jobs	Machine B	Machine A
	In Out	In Out
2	0.0 - 3.0	6.0 - 7.6
4	3.0 - 4.0	7.6 - 11.0
1	4.0 - 5.4	11.0 - 15.2
5	5.4 -10.8	15.2 - 17.4
3	10.8 - 16.2	20.2 - 24.1

Tableau-10

The total expected elapsed time when the order is from A to B for the sequence (5, 3, 2, 4, 1) is 21.7 units and for the sequence (2, 4, 1, 5, 3) is 24.1 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 (5, 3, 2, 4, 1) for the order A→B.

REMARKS

1. If the probabilities are not associated with processing time then the results tally with [6].
2. If probabilities are associate with processing time then the results tally with[5].
3. If both probabilities and transportation time are not included then results tally with[6].
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 various parameters such as break down intervals, weightage of jobs etc.

6. The study may further be extended for n jobs 3 machine open shop problem.

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] **D. Rebaine, V. A. Strusevich(1998)**,"Two machine open shop scheduling with special transportation times",CASSM R&D paper 15,University of Greenwich, London, UK.

[3] **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", Presented and Published in the Proceedings of International conference on Industrial & Applied Mathematics at India International and published in the journal reflections des ERA JMS VOL 1 FEB 2006,pp 53-70.

[4] **Gupta D; Singla P; Sharma S et.al(2011)** , **"**Minimizing Rental Cost for n-jobs 2-Machines flow shop scheduling , processing time association with probabilities including Transportation time and job block criteria", Information and Knowledge Management, ISSN 2224 5758(paper) ISSN 2224- 896x(online),Vol 1,No.3,2011.

[5] **Gupta Deepak, Singh T.P (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.

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

[7] **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.

[8] **Jackson, J.R.(1956)** , "An extention of Johnson's result on job scheduling", Nav.Res.Log.Quar.,3, 201-203.

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

[10] **M. Dell'Amico(1996)** , " Shop problems with two machines and time lags",Oper.Res.44,pp 777-787.

[11] **Singh T.P (1985)**,"On n*2 shop problem involving job block, transportation time & break down machine times", PAMS Vol. XXI, pp 1-2.

[12] **Singh T.P. & Rastogi V.K. (1998),** "Group job –restrictions in a flowshop problem involvingseparated sequences independent set-up times and transportation time." The Mathematics Education Vol.XXXII.No.1.pp 12-16.

[13] **Yoshida and Hitomi(1979)** , "Optimal two stage production schedule with set up time Separated", AIIE Transactions. Vol. II. Pp261-263.

Source of support: Nil, Conflict of interest: None Declared