Journal of Natural Sciences Research www.iiste.org
ISSN 2224-3186 (Paper) ISSN 2225-0921 (Online)
Vol.1, No.3, 2011

Optimal Three Stage Flow Shop Scheduling in which Processing Time, Set Up Time, Each Associated With Probabilities Including Transportation Time and Jobs in a String of Disjoint Job-Blocks

Sameer Sharma*

Department of Mathematics, Maharishi Markandeshwar University, Mullana, Ambala, India

* E-mail of the corresponding author: samsharma31@yahoo.com

Abstract

The paper deals with n jobs, 3 machines flow shop production scheduling in which processing times and set up times are associated with their respective probabilities involving transportation time and jobs are processed in two disjoint job blocks in a string. A heuristic method with an objective to minimize the total time elapsed time/idle time of the jobs/machines is discussed. A computer program followed by numerical illustration is given to clarify the algorithm.

Keywords: Disjoint job-block, Set up time, Processing time, Transportation time, Equivalent job

1. Introduction

In today's world of global competition, scheduling problem has become vital in order to meet customer requirements as promptly as possible while maximizing the profits. A scheduling problem is to find sequences of jobs on given machines with the objective of minimizing some function of the job completion times. In flow shop scheduling all jobs passes through all machines in the same order without any pre-emission. Johnson (1954) gave a heuristic technique for production schedule in which n jobs are processed on two or three machines in an ordered manner to minimize the total idle time of machines. The work was developed by Ignall & Scharge (1965), Camphell (1970) , Maggu & Das (1977) , Yoshida & Hitomi (1979) , Singh (1985) , Anup (2002) ,Chandramouli (2005) , Khodadadi (2008), Pandian & Rajenderan (2010) by considering various parameters. Heydari (2003) dealt with a flow shop scheduling problem where the jobs are processed in two disjoint job blocks in a string consists of one block in which order of jobs is fixed & other block in which order of job is arbitrary.

Gupta, Sharma & Gulati (2011) studied n×3 machine flow shop schedule in which processing time, set up time, each associated with probabilities along with jobs in a string of disjoint job-blocks. Most machine scheduling models assume that jobs are delivered instantaneously from one location to another without considering significant transportation time. However, there are many situations where the transportation times are quite significant and can not be simply neglected. As example, when the machines on which jobs are to be processed are planted at different stations and these jobs require form of loading-time of jobs, moving time and then unloading-time of jobs. In this paper, we have studied machines scheduling problems with explicit transportation considerations. We have extended the study made by Gupta, Sharma & Gulati (2011) by introducing the concept of transportation time. The problem discussed here is wider and practically more applicable and has significant results in process industries

2. Practical Situation

Many applied and experimental situations exist in our day-to-day working in factories and industrial production concerns. In many manufacturing / production 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 includes loading time, moving time and unloading time etc.) has a significant role in production concern. Setup includes work to prepare the machine, process or bench for product parts or the cycle. This includes obtaining tools, positioning work-in-process material, return tooling, cleaning up, setting the required jigs and fixtures, adjusting tools and inspecting material and hence significant. 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, .i.e. how much is to be charged from the priority customer(s) as compared to non priority customer(s).

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}

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

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

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

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

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

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

4. Assumptions

1. n jobs be processed through three machines M_1, M_2 & M_3 in the order $M_1M_2M_3$ i.e. no passing is allowed.
2. A sequence of *k* jobs $i_1, i_2 \ldots\ldots i_k$ as a block or group-job in the *order* $(i_1, i_2 \ldots\ldots i_k)$ shows priority of job i_1 over i_2, etc.
3. Jobs may be held in inventory before going to a machine.
4. The storage space is available and the cost of holding inventory for each job is either same or negligible.
5. Time intervals for processing are independent of the order in which operations are performed.
6. A job is an entity i.e. even though the job represents a lot of individual part; no job may be processed by more than one machine at a time.
7. Each operation once started must performed till completion.

4. Problem Formulation

Let some job *i* (i = 1,2,……..,n) is to be processed on three machines M_j (j = 1,2,3) in the way such that no passing is allowed. Let a_{ij} be the processing time of i^{th} job on j^{th} machine with probabilities p_{ij} and s_{ij} be the setup time of i^{th} job on j^{th} machine with probabilities q_{ij}. Let A_{ij} be the expected processing time and $S_{i,j}$ be the expected setup time of i^{th} job on j^{th} machine. Let $T_{i,j\rightarrow k}$ be the transportation time of i^{th} job from j^{th} machine to k^{th} machine..Let $\alpha = (i_k, i_m)$ be an equivalent job for job block in which job i_k is given priority over job i_m Take two job blocks α and β such that block α consists of *m* jobs out of *n* jobs in which the order of jobs is fixed and β consists of *r* jobs out of *n* in which order of jobs is arbitrary such that $m + r = n$. let $\alpha \cap \beta = \Phi$ i.e. the two job blocks α & β form a disjoint set in the sense that the two blocks have no job in common. A string *S* of job blocks α and β is defined as $S = (\alpha, \beta)$. Our objective is to find an optimal

Journal of Natural Sciences Research www.iiste.org
ISSN 2224-3186 (Paper) ISSN 2225-0921 (Online)
Vol.1, No.3, 2011
schedule for all the jobs minimizing the total elapsed time.

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

Jobs	Machine A				$T_{i,1\to2}$	Machine B				$T_{i,2\to3}$	Machine C			
i	a_{i1}	p_{i1}	s_{i1}	q_{i1}		a_{i2}	p_{i2}	s_{i2}	q_{i2}		a_{i3}	p_{i3}	s_{i3}	q_{i3}
1	a_{11}	p_{11}	s_{11}	q_{11}	$T_{1,1\to2}$	a_{12}	p_{12}	s_{12}	q_{12}	$T_{1,2\to3}$	a_{13}	p_{13}	s_{13}	q_{13}
2	a_{21}	p_{21}	s_{21}	q_{21}	$T_{2,1\to2}$	a_{22}	p_{22}	s_{22}	q_{22}	$T_{2,2\to3}$	a_{23}	p_{23}	s_{23}	q_{23}
3	a_{31}	p_{31}	s_{31}	q_{31}	$T_{3,1\to2}$	a_{32}	p_{32}	s_{32}	q_{32}	$T_{3,2\to3}$	a_{33}	p_{33}	s_{33}	q_{33}
4	a_{41}	p_{41}	s_{41}	q_{41}	$T_{4,1\to2}$	a_{42}	p_{42}	s_{42}	q_{42}	$T_{4,2\to3}$	a_{43}	p_{43}	s_{43}	q_{43}
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n	a_{n1}	p_{n1}	s_{n1}	q_{n1}	$T_{n,1\to2}$	a_{n2}	p_{n2}	s_{n2}	q_{n2}	$T_{n,2\to3}$	a_{n3}	p_{n3}	s_{n3}	q_{n3}

(Tableau 1)

5. Algorithm:

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

$$A_{ij} = a_{ij} \times p_{ij} \text{ and } S_{ij} = s_{ij} \times q_{ij} \quad \forall i, j = 1,2,3$$

Also we consider the following structure relation holds good if

Step 2: Check the condition

Either Min $\{A_{i1} + T_{i,1\to2} - S_{i2}\} \geq$ Max$\{A_{i2} + T_{i,1\to2} - S_{i1}\}$

or Min$\{A_{i3} + T_{i,2\to3} - S_{i2}\} \geq$ Max$\{A_{i2} + T_{i,2\to3} - S_{i3}\}$ 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

$G_i = A_{i1} + A_{i2} + \max(S_{i1}, S_{i2}) + T_{i,1\to2}$ and $H_i = A_{i2} + A_{i3} - S_{i3} + T_{i,2\to3}$

Step 4: Take equivalent job $\alpha = (i_k, i_m)$ for the given job block (i_k, i_m) and define its processing time on the lines of Maggu & Das (1977) defined as follows:

$G_\alpha = G_k + G_m - min(G_m, H_k)$

$H_\alpha = H_k + H_m - min(G_m, H_k)$

Step 5: Obtain the order of jobs in the job block β in an optimal manner using Johnson's (1954) technique by treating job block β as sub flow shop scheduling problem of the main problem. Let β' be the new job block. Define its processing time $G_{\beta'}$ & $H_{\beta'}$ on the lines of Maggu & Das (1977) as defined in step 4.

Now, the given problem reduce into new problem replacing m jobs by job block α with processing times G_α & H_α on machine G & H respectively as defined in step 4 and r jobs of job block β by β' with processing times $G_{\beta'}$ & $H_{\beta'}$ on machine G & H respectively as defined in step 5.

The new problem can be represented as –

Jobs (i)	Machine G (G_i)	Machine H (H_i)
α	G_α	H_α
β'	$G_{\beta'}$	$H_{\beta'}$

(Tableau – 2)

Step 6: Consider S_1 of all the processing time G_i when $G_i \leq H_i$ and let S_2 denote the set of processing times which are not covered in set S_1.

Step 7: Let S'_1 denote a suboptimal sequence of jobs corresponding to non decreasing times in set S_1 & let S'_2 denote a suboptimal sequence of jobs corresponding to non-decreasing times in set S_2.

Step 8: The augmented ordered sequence (S'_1, S'_2) gives optimal sequence for processing the jobs for the original problem.

6. Programme

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

int n;
float a1[16],b1[16],c1[16],g[16],h[16],sa1[16],sb1[16],sc1[16];
float macha[16],machb[16],machc[16];
int e;int group[16];//variables to store two job blocks
float minval;int gg=0;float gcal;float hcal;float gbeta=0.0,hbeta=0.0;float galfa=0.0,halfa=0.0;
char s1[5];char s2[5];
void ghcal(float k,float m)
{
float minv;
if(g[m]>h[k])
minv=h[k];
else
minv=g[m];gcal=g[k]+g[m]-minv;hcal=h[k]+h[m]-minv;
//return(c);}
void main()
{
        clrscr();
        int a[16],b[16],c[16],sa[16],sb[16],sc[16],T12[16],T23[16];
        float p[16],q[16],r[16],u[16],v[16],w[16];float maxv;
        cout<<"How many Jobs (<=15) : ";cin>>n;
        if(n<1 || n>15)
        {       cout<<endl<<"Wrong input, No. of jobs should be less than 15..\n Exitting";     getch();
                exit(0);}
        for(int i=1;i<=n;i++)
                {cout<<"\nEnter the processing time and its probability, Setup time and its probability of
"<<i<<" job for machine A and Transportation time from A to B : ";
        cin>>a[i]>>p[i]>>sa[i]>>u[i]>>T12[i];
                cout<<"\nEnter the processing time and its probability, Setup time and its probability of
"<<i<<" job for machine B and Transportation time from B to C : ";
        cin>>b[i]>>q[i]>>sb[i]>>v[i]>>T23[i];
                cout<<"\nEnter the processing time and its probability, Setup time and its probability of
"<<i<<"job for machine C: ";cin>>c[i]>>r[i]>>sc[i]>>w[i];
```

```
//Calculate the expected processing times of the jobs for the machines:
                    a1[i] = a[i]*p[i];  b1[i] = b[i]*q[i];  c1[i] = c[i]*r[i];
//Calculate the expected setup times of the jobs for the machines:
                    sa1[i] = sa[i]*u[i];sb1[i] = sb[i]*v[i];sc1[i] = sc[i]*w[i];}
          cout<<endl<<"Expected processing time of machine A, B and C: \n";
          for(i=1;i<=n;i++)
          {cout<<a1[i]<<"\t"<<sa1[i]<<"\t"<<T12[i]<<"\t"<<b1[i]<<"\t"<<sb1[i]<<"\t"<<T23[i]<<"\t"<<c1
[i]<<"\t"<<sc1[i];cout<<endl;}
//Finding smallest in a1
          float mina1;mina1=a1[1]+T12[1]-sb1[1];
          for(i=2;i<n;i++)
          {if((a1[i]+T12[i]-sb1[i])<mina1)
                    mina1=a1[i]+T12[i]-sb1[i];}
          //For finding largest in b1
          float maxb1;maxb1=b1[1]+T12[1]-sa1[1];
          for(i=2;i<n;i++)
          {         if(b1[i]+T12[i]-sa1[i]>maxb1)
                    maxb1=b1[i]+T12[i]-sa1[i];}
          float maxb2;maxb2=b1[1]+T23[1]-sc1[i];
          for(i=2;i<n;i++)
          {if((b1[i]+T23[i]-sc1[i])>maxb2)
                    maxb2=b1[i]+T23[i]-sc1[i];}
//Finding smallest in c1
          float minc1;minc1=c1[1]+T23[1]-sb1[i];
          for(i=2;i<n;i++)
          {if((c1[i]+T23[i]-sb1[i])<minc1)
                    minc1=c1[i]+T23[i]-sb1[i];}
          if(mina1<=maxb1||minc1<=maxb2)
          {g[i]=a1[i]+b1[i]+maxv;h[i]=b1[i]+c1[i]-sc1[i];}
else  {cout<<"\n data is not in Standard Form...\nExitting";getch();exit(0);}
//Function for two ficticious machine G and H
          for(i=1;i<=n;i++)
          {if(sa1[i]>sb1[i])
          {maxv= sa1[i];}
else      {maxv=sb1[i];}
          g[i]=a1[i]+b1[i]+maxv+T12[i];h[i]=b1[i]+c1[i]-sc1[i]+T23[i]; }
cout<<endl<<"Expected processing time for two fictious machines G and H: \n";
 for(i=1;i<=n;i++)
 { cout<<endl;cout<<g[i]<<"\t"<<h[i];cout<<endl;}
          cout<<"\nEnter the number of fixed jobs in job block alpha <="<<n<<":";cin>>e;
          cout<<"\nEnter the fixed job blocks ("<<e<<" numbers from 1 to "<<n<<") alpha : ";
          for(int y=1;y<=e;y++)
          {cin>>group[y];}
```

```
            cout<<"\nEnter the jobs having disjoint job block ( numbers from 1 to "<<n<<" other than the
fixed job block) beta:";
               for(int j=e+1;j<=n;j++)
              {cin>>group[j];}
 float btj[16],btg[16],bth[16];
cout<<"Expected processing time for two fictious machines G and H for Beta: \n";
for(i=1,j=e+1;j<=n;i++,j++)
            {btj[i]=group[j];btg[i]=g[group[j]];bth[i]=h[group[j]];
            cout<<endl<<btj[i]<<"\t"<<btg[i]<<"\t"<<bth[i];}
float mingh[16];char ch[16];
  for(i=1;i<=n-e;i++)
  {
            if(btg[i]<bth[i])
                         {
                                    mingh[i]=btg[i];ch[i]='g';
                         }
            else
                        {
                                    mingh[i]=bth[i];ch[i]='h';
                         }}
                        for(i=1;i<=n-e;i++)
            {for(j=1;j<=n-e;j++)
                                                if(mingh[i]<mingh[j])
                                                            {float temp=mingh[i]; int temp1=btj[i]; char d=ch[i];
                                                                        mingh[i]=mingh[j]; btj[i]=btj[j]; ch[i]=ch[j];
                                                                        mingh[j]=temp; btj[j]=temp1; ch[j]=d;}}
            // calculate beta scheduling
float sbeta[16];int t=1,s=0;
for(i=1;i<=n-e;i++)
{if(ch[i]=='h')
            { sbeta[(n-s-e)]=btj[i];s++;}
else if(ch[i]=='g')
            {sbeta[t]=btj[i];t++;}
}
cout<<endl<<endl<<"Beta Scheduling:"<<"\t";
for(i=1;i<=n-e;i++)
{cout<<sbeta[i]<<"  ";}
//calculate G_Alfa and H_Alfa
ghcal(group[1],group[2]);
galfa=gcal;halfa=hcal;i=3;
while(i<=e)
{
if(i>e)
```

```
break;
else
{
if(g[group[i]]<halfa)
minval=g[group[i]];
else
minval=halfa;galfa=galfa+g[group[i]]-minval;halfa=halfa+h[group[i]]-minval;
}
i++;
}
cout<<endl<<endl<<"G_Alfa="<<galfa;cout<<endl<<"H_Alfa="<<halfa;
//calculate G_Beta and H_Beta
ghcal(sbeta[1],sbeta[2]);gbeta=gcal;hbeta=hcal;i=3;
while(i<=(n-e))
{
if(i>(n-e))
break;
else
{
if(g[sbeta[i]]<hbeta)
minval=g[sbeta[i]];
else
minval=hbeta;gbeta=gbeta+g[sbeta[i]]-minval;hbeta=hbeta+h[sbeta[i]]-minval;}
i++;}
cout<<endl<<endl<<"G_Beta="<<gbeta;cout<<endl<<"H_Beta="<<hbeta;
//calculate optimal sequence
if(galfa<=halfa)
{
s1[1]='a';s2[1]='\0';
}
else
{
s2[1]='a';s1[1]='\0';
}
if(gbeta<=hbeta)
{
s1[2]='b';s2[2]='\0';
}
else
{
s2[2]='b';s1[2]='\0';
}
//cout<<endl<<endl<<"Optimal Sequence:"<<"\t";
```

```
int arr[16];
if(s1[1]=='a')
		{
		//cout<<"\n a";
		for(i=1;i<=e;i++)
				{
				//cout<<group[i]<<"\t";
				arr[i]=group[i];
				}
				gg=gg+e;
		}
if(s1[2]=='b')
		{
		//cout<<"\n b";
		for(i=1;i<=n-e;i++)
				{
				//cout<<endl<<sbeta[i]<<"\t";
				arr[i+gg]=sbeta[i];
				}
				gg=gg+(n-e)+1;
		}
if(s2[1]=='a')
		{
		//cout<<"\n a";
		for(i=1;i<=e;i++)
				{
				//cout<<endl<<group[i]<<"\t";
				arr[i+gg]=group[i];
				}
				gg=gg+e;
		}
if(s2[2]=='b')
		{
		//cout<<"\n b";
		for(i=1;i<=(n-e);i++)
				{
				//cout<<sbeta[i]<<"\t";
				arr[i+gg]=sbeta[i];
				}}
//calculating total computation sequence;
float time=0.0,macha1[16] ;float maxv1[16],maxv2[16];
		macha[1]=time+a1[arr[1]];
for(i=2;i<=n;i++)
```

```
            {macha1[i]=macha[i-1]+sa1[arr[i-1]];macha[i]=macha1[i]+a1[arr[i]];}
            machb[1]=macha[1]+b1[arr[1]]+T12[arr[1]];
for(i=2;i<=n;i++)
            {if((machb[i-1]+sb1[arr[i-1]])>(macha[i]+T12[arr[i]]))
      {maxv1[i]=machb[i-1]+sb1[arr[i-1]];}
  else
      {maxv1[i]=macha[i]+T12[arr[i]];}machb[i]=maxv1[i]+b1[arr[i]];}
            machc[1]=machb[1]+c1[arr[1]]+T23[arr[1]];
for(i=2;i<=n;i++)
            {if((machc[i-1]+sc1[arr[i-1]])>(machb[i]+T23[arr[i]]))
            maxv2[i]=machc[i-1]+sc1[arr[i-1]];
else
            maxv2[i]=machb[i]+T23[arr[i]];machc[i]=maxv2[i]+c1[arr[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<<" "<<arr[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<<arr[1]<<"\t"<<time<<"--"<<macha[1]<<"    \t"<<"\t"<<macha[1]+T12[arr[1]]<<"--"<<machb[1]<<"
\t"<<"\t"<<machb[1]+T23[arr[1]]<<"--"<<machc[1]<<endl;
for(i=2;i<=n;i++)
            {cout<<arr[i]<<"\t"<<macha1[i]<<"--"<<macha[i]<<"    "<<"\t"<<maxv1[i]<<"--"<<machb[i]<<"
"<<"\t"<<maxv2[i]<<"--"<<machc[i]<<endl;}
cout<<"\n\n\nTotal Computation Time (T) = "<<machc[n];
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];}
float tt=machc[n];
cout<<endl<<endl<<endl<<"Total Expected Idle Time on Machine A: "<<(tt-sum1);
cout<<endl<<"Total Expected Idle Time on Machine B: "<<(tt-sum2);
cout<<endl<<"Total Expected Idle Time on Machine C: "<<(tt-sum3);
cout<<"\n\n\t*******************************************************************";
getch();
}
```

7. Numerical Illustration

Consider 5 jobs, 3 machine flow shop problem with processing time ,setup time associated with their respective probabilities and transportation time as given in table

Jobs	Machine M_1				$T_{i,1\to2}$	Machine M_2				$T_{i,2\to3}$	Machine M_3			
i	a_{i1}	p_{i1}	s_{i1}	q_{i1}		a_{i2}	p_{i2}	s_{i2}	q_{i2}		a_{i3}	p_{i3}	s_{i3}	q_{i3}

1	30	0.2	3	0.3	2	10	0.3	3	0.2	2	20	0.2	4	0.2
2	32	0.2	2	0.1	1	22	0.2	2	0.2	1	19	0.3	3	0.2
3	43	0.1	2	0.3	2	22	0.2	1	0.3	2	15	0.2	3	0.3
4	25	0.2	4	0.1	3	25	0.1	3	0.1	3	24	0.1	4	0.2
5	23	0.3	2	0.2	4	10	0.2	2	0.2	1	26	0.2	5	0.1

(Tableau 3)

Our objective is to obtain an optimal schedule for above said problem to minimize the total production time / total elapsed time in which jobs 2,5 are to be processed as a group job in a fixed order and remaining jobs as a disjoint string in any random order.

Solution: As per Step 1: the expected processing times and expected setup times for machines M_1, M_2 and M_3 are as shown in table 4.

As per step 2: The expected processing time for two fictitious machine G & H is as shown in table 5.

As per step 3: Here $\beta = (1, 3, 4)$

Now, using Johnson (1954) technique by treating job block β as sub flow shop scheduling problem of the main problem. Let β' be the new job block. Here we get $\beta' = (3, 1, 4)$

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

Therefore, $G_\alpha = 12.2 + 13.3 - 10.5 = 15$ and $H_\alpha = 10.5 + 7.7 - 10.5 = 7.7$

Also $\beta' = (3,1,4) = ((3, 1), 4) = (\alpha', 4)$, where $\alpha' = (3, 1)$

Therefore, $G_{\alpha'} = 11.3 + 11.9 - 8.5 = 14.7$ and $H_{\alpha'} = 8.5 + 8.2 - 8.5 = 8.2$

$G_{\beta'} = 14.7 + 10.9 - 8.2 = 17.4$ and $H_{\beta'} = 8.2 + 7.1 - 8.2 = 7.1$

Now problem reduces to jobs α *and* β' as shown in table 6

As per step 5: $S_1 = \varphi$, $S_2 = [15, 17.4]$

As per step 6: $S'_1 = \varphi$, $S'_2 = (\alpha, \beta')$

As per step 7: The Optimal sequence is S = 2 – 5 – 3 – 1 – 4 .The In-Out flow table for the optimal sequence S is as shown in table 7.

Total expected idle time on machine A = 0.2+0.4+0.6+0.9+10.9 = 13hrs

Total expected idle time on machine B = 7.4+5.7+0.7+2.2+3.9+5.4= 25.3 hrs

Total expected idle time on machine C = 12.8+2+0.9+2.2+3.4 = 21.3 hrs

8. Conclusions

This paper provides a new heuristic method with an objective to minimize the total time elapsed time / idle time of the jobs / machines for n jobs, 3 machines flow shop production scheduling in which processing times and set up times are associated with their respective probabilities involving transportation time and jobs are processed in two disjoint job blocks in a string. The study may further be extended by considering various parameters such as weights in jobs, arbitrary time, break down interval etc.

References

Anup and Maggu P.L(2002),, "On an optimal schedule procedure for a n x 2 flow shop scheduling problem involving processing time, set up times, transportation times with their respective probabilities and an equivalent job for a job block", *PAMS*, **56**(1-2) , 88-93.

Ahmad Pour Darvish Heydari,(2003), "On flow shop scheduling problem with processing of jobs in a string of disjoint job blocks: fixed order jobs and arbitrary order jobs", *JISSOR*, XXIV, , 39- 43.

Campbell, H.A, Duder, R.A & Smith, M.L.(1970), "A heuristic algorithm for the n-job, m-machine sequencing problem", *Management Science*, **16**, B630-B637.

Chandramouli,A.B. (2005), "Heuristic approach for n-jobs, 3-machine flow shop scheduling problem involving transportation time, break down time and weights of jobs", *Mathematical and Computational Applications*,**10**(2), 301-305.

Gupta D., Sharma S.& Gulati, N.(2011), "n×3 flow shop production schedule, processing time, setup time each associated with probabilities along with jobs in string of disjoint job-blocks", *Antartica J. Math*,8(5),, 443 – 457.

Ignall & Schrage,L.(1965), "Application of the branch-and bound technique to some flow shop scheduling problems", *Operation Research* , **13**, 400-412.

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

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

Maggu, P.L, and Das, G,(1977), "Equivalent jobs for job block in job sequencing", *Operation Research*, **14**(4), 277-281.

Rajenderan, C.(1993), "Heuristic algorithm for scheduling in flow shop to minimize the total flow time", *International Journal of Production Economics*, **29**(1), 65-73.

Singh, T.P.(1985), "On n x 2 flow shop problem involving job block, transportation times & break- down machine times", *PAMS XXI*, 1-2.

Singh, T P, Kumar, R. & Gupta, D. (2005), "Optimal three stage production schedule, the processing and set up times associated with probabilities including job block criteria", *Proceedings of National Conference on FACM*, 463-470.

Pandian, P. &.Rajendran, P.(2010), "Solving constrained flow-shop scheduling problems with three machines", *Int. J. Contemp. Math. Sciences*, **5**(19), 921-929.

Yoshida & Hitomi (1979), "Optimal two stage production scheduling with set up times separated", *AIIE Transactions*, Vol . II, 261-263

Tables

Table 4: The expected processing times and expected setup times for machines M_1, M_2 and M_3 are

Jobs	A_{i1}	S_{i1}	$T_{i,1\to2}$	A_{i2}	S_{i2}	$T_{i,2\to3}$	A_{i3}	S_{i3}
1	6.0	0.9	2	3.0	0.6	2	4.0	0.8
2	6.4	0.2	1	4.4	0.4	1	5.7	0.6
3	4.3	0.6	2	4.4	0.3	2	3.0	0.9
4	5.0	0.4	3	2.5	0.3	3	2.4	0.8
5	6.9	0.4	4	2.0	0.4	1	5.2	0.5

Table 5: The expected processing time for two fictitious machine G & H is

Jobs	G_i	H_i
1	11.9	8.2
2	12.2	10.5
3	11.3	8.5
4	10.9	7.1
5	13.3	7.7

Table 6: The new reduced problem is

Jobs(i)	Machine G(G_i)	Machine H(H_i)
α'	15	7.7
β'	17.4	7.1

Table 7: The In-Out flow table for the optimal sequence S is

Jobs	Machine M_1	$T_{i,1\to2}$	Machine M_2	$T_{i,2\to3}$	Machine M_3
i	In – Out		In – Out		In - Out
2	0 – 6.4	1	7.4 – 11.8	1	12.8 – 18.5
5	6.6 – 13.5	4	17.5 – 19.5	1	20.5 – 25.7
3	13.9 – 18.2	2	20.2 – 24.6	2	26.6 – 29.6
1	18.8 – 24.8	2	26.8 – 29.8	2	31.8 – 35.8
4	25.7 – 30.7	3	33.7 – 36.2	3	39.2 – 41.6