Industrial Engineering Letters www.iiste.org
ISSN 2224-6096 (print) ISSN 2225-0581 (online)
Vol 1, No.2, 2011

Constrained Flow Shop Scheduling with n-Jobs, 3-Machines, Processing Time Associated with Probability involving Transportation Time, Breakdown Interval and Weightage of Jobs

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

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

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

Geeta Bhalla
Department of Mathematics
Sant Baba Bhag Singh Institute of Engineering & Tech.
Padhiana, Jalandhar, Punjab, India
geet14ar@gmail.com

Abstract

This paper is an attempt to find simple heuristic algorithm for n jobs, 3 machines flow shop scheduling problem in which processing times are associated with probabilities involving transportation time and break down interval. Further jobs are attached with weights to indicate their relative importance. A simple heuristic approach to find optimal or near optimal sequence minimizing the total elapsed time whenever mean weighted production flow time is taken into consideration. The proposed method is very easy to understand and, also provide an important tool for the decision makers. A computer programme followed by a numerical illustration is also given to clarify the algorithm.

Keywords: Flow shop scheduling, Processing time, Transportation time, Breakdown interval, Weights of job, Optimal sequence

1. Introduction

Flow Shop scheduling is a typical combinatorial optimization problem, where each job has to go through the processing on each and every machine in the shop floor. Each machine has same sequence of jobs. The jobs have different processing time for different machines. So in this case we arrange the jobs in a particular order and get many combinations and choose that combination where we get the minimum make span. It is an important process widely used in manufacturing, production, management, computer science, and so on. Appropriate scheduling not only reduces manufacturing costs but also reduces possibilities for violating the due dates. Finding good schedules for given sets of jobs can thus help factory supervisors effectively to control the job flow and provide solutions for job sequencing. In flow shop scheduling problems, the objective is to obtain a sequence of jobs which when processed on the machine will optimize some well defined criteria, The number of possible schedules of the flow shop scheduling problem involving n-jobs and m-machines is $(n!)^m$. The scheduling problem practically depends upon the three important factors Job Transportation which includes loading time, moving time and unloading time etc., Weightage of a job which is due to the relative importance of a job as compared with other jobs and machine Breakdown due to failure of a component of machine for a certain interval of time or the machines are supposed to stop their working for a certain interval of time due to some external imposed policy such as non supply of electric current to the machines may be a government policy due to shortage of electricity production. These concepts were separately studied by various researchers Johnson(1954), Jakson(1956), Belman(1956), Baker(1974), Maggu and Das (1981), Nawaz et al.(1983), Miyazaki and Nishiyama (1980), Parker(1995), Narain and Bagga(1998), Singh,T.P. (1985), Chandramouli(2005), Belwal and Mittal (2008), khodadadi (2008), Pandian and Rajendran (2010), Gupta and Sharma (2011).

Pandian and Rajendran(2010) proposed a heuristic algorithm for solving constrained flow shop scheduling problems with three machines. In practical situations, the processing time are always not be exact as has been taken by most of researchers, hence, we made an attempt to associate probabilities with processing time. In this paper, we propose a new simple heuristic approach to obtain an optimal sequence with three machines in which probabilities are associated with processing time involving transportation time, breakdown interval and weights of jobs. We have obtained an algorithm which minimizing the total elapsed time whenever means weighted production flow time is taken into consideration. Thus the problem discussed here is wider and practically more applicable and will have significant results in the process industry.

2. Practical Situation

Many applied and experimental situations exist in our day-to-day working in factories and industrial production concerns etc. The practical situation may be taken in a paper mill, sugar factory and oil refinery etc. where various qualities of paper, sugar and oil are produced with relative importance i.e. weight in jobs. 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 includes loading time, moving time and unloading time etc.) has a significant role in production concern. The break down of the machines (due to delay in material, changes in release and tails date, tool unavailability, failure of electric current, the shift pattern of the facility, fluctuation in processing times, some technical interruption etc.) have significant role in the production concern.

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

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

A'_{ij} : Expected processing time of i^{th} job after break-down effect on j^{th} machine

$I_{ij}(S_k)$: 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

W_i : Weight assigned to i^{th} job

L : Length of break down interval

4. Problem Formulation

Let some job i (i = 1,2,……..,n) are to be processed on three machines M_j (j = 1,2,3). Let a_{ij} be the processing time of i^{th} job on j^{th} machine and p_{ij} be the probabilities associated with a_{ij}. Let $T_{i,j\to k}$ be the transportation time of i^{th} job from j^{th} machine to k^{th} machine. Let w_i be the weights assigned to the ith job.. Our aim is to find the sequence $\{S_k\}$ of the jobs which minimize the total elapsed time, whenever mean weighted production flow time is taken into consideration.. The mathematical model of the given problem P in matrix form can be stated as:

Jobs	Machine A		$T_{i,1\to 2}$	Machine B		$T_{i,2\to 3}$	Machine C		Weights of Jobs
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}	w_1
2	a_{21}	p_{21}	$T_{2,1\to 2}$	a_{22}	p_{22}	$T_{2,2\to 3}$	a_{23}	p_{23}	w_2
3	a_{31}	p_{31}	$T_{3,1\to 2}$	a_{32}	p_{32}	$T_{3,2\to 3}$	a_{33}	p_{33}	w_3
4	a_{41}	p_{41}	$T_{4,1\to 2}$	a_{42}	p_{42}	$T_{4,2\to 3}$	a_{43}	p_{43}	w_4
-	-	-		-	-	-	-	-	-
n	a_{n1}	p_{n1}	$T_{n,1\to 2}$	a_{n2}	p_{n2}	$T_{n,2\to 3}$	a_{n3}	p_{n3}	w_n

5. Algorithm

The following algorithm provides the procedure to determine an optimal sequence to the problem P:

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

Step 2 : Check the structural condition

Max $\{A_{i1} + T_{i.1\to 2}\} \geq$ Min $\{A_{i2} + T_{i,1\to 2}\}$

or Max $\{A_{i3} + T_{i,2\to 3}\} \geq$ Min $\{A_{i2} + T_{i,2\to 3}\}$, or both.

If these structural conditions satisfied then go to step 3 else the data is not in standard form.

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

$G_i = |A_{i1} - A_{i2} - T_{i,1\to 2} - T_{i,2\to 3}|$ and $H_i = |A_{i3} - A_{i2} - T_{i,1\to 2} - T_{i,2\to 3}|$.

Step 4 : Compute *Minimum* (G_i ,H_i)

- If *Min* $(G_i, H_i)=G_i$ then define $G_i^{'}=G_i+w_i$ and $H_i^{'}=H_i$.
- If *Min* $(G_i, H_i)=H_i$ then define $G_i^{'}=G_i$ and $H_i^{'}=H_i+w_i$.
- If *Min* $(G_i, H_i)=H_i=G_i$ then define $G_i^{'}=G_i$ and $H_i^{'}=H_i+w_i$ or $G_i^{'}=G_i+w_i$ and $H_i^{'}=H_i$ arbitrarily (with minimum total elapsed time)

Step 5 : Define a new reduced problem with $G_i^{''}$ and $H_i^{''}$ where

$G_i^{''} = G_i^{'}/w_i, H_i^{''} = H_i^{'}/w_i \quad \forall i = 1,2,3.....,n$

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 the sequences $S_1, S_2, ..., S_r$ obtained in step 6. Let the mean flow time is minimum for the sequence S_k. Now, read the effect of break down interval (a, b) on different jobs on the lines of *Singh T.P.*[17] for the sequence S_k.

Step 8 : Form a modified problem with processing time A'_{ij} ; *i = 1, 2, 3, --- n; j= 1, 2,3.*

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

$A'_{ij} = A_{ij} + L$; Where L = b – a, the length of break-down interval

If the break-down interval (a, b) has no effect on i[th] job then

$A'_{ij} = A_{ij}$.

Step 9 : Repeat the procedure to get the optimal sequence for the modified scheduling problem using steps 2 to 6.Determine the total elapsed time.

Step 10 : Find the performance measure studied in weighted mean flow time defined as

$F = \sum_{i=1}^{n} w_i f_i \Big/ \sum_{i=1}^{n} f_i$, where f_i is flow time of i^{th} job.

6. Programme

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<math.h>
int n,j;
float a1[16],b1[16],c1[16],a11[16],b11[16],c11[16],g[16],h[16],T12[16],T23[16];
float macha[16],machb[16],machc[16],macha1[16],machb1[16],machc1[16];
int f=1;
float minval,minv,maxv1[16],maxv2[16], w[16];
int bd1,bd2;// Breakdown interval
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];
cout<<"\nEnter the weightage of "<<i<<"job:"; cin>>w[i];
//Calculate the expected processing times of the jobs for the machines:
    a1[i] = a[i]*p[i]; b1[i] = b[i]*q[i]; c1[i] = c[i]*r[i];}
    cout<<endl<<"Expected processing time of machine A, B and C with weightage: \n";
    for(i=1;i<=n;i++){cout<<j[i]<<"\t"<<a1[i]<<"\t"<<b1[i]<<"\t"<<c1[i]<<"\t"<<w[i]; cout<<endl;}
```

Industrial Engineering Letters www.iiste.org
ISSN 2224-6096 (print) ISSN 2225-0581 (online)
Vol 1, No.2, 2011

```
        cout<<"\nEnter the two breakdown interval:"; cin>>bd1>>bd2;
        //Function for two ficticious machine G and H //Finding largest in a1
        float maxa1; maxa1=a1[1]+T12[1];for(i=2;i<n;i++)
        { if(a1[i]+T12[i]>maxa1)
        maxa1=a1[i]+T12[i];}
//For finding smallest in b1
        float minb1; minb1=b1[1]+T23[1];or(i=2;i<n;i++)
        {if(b1[i]+T23[i]<minb1)
        minb1=b1[i]+T23[i];}
        float minb2;minb2=b1[1]+T12[1]; for(i=2;i<n;i++)
        {if(b1[i]+T12[i]<minb2)minb2=b1[i]+T12[i];}
//Finding largest in c1
        float maxc1;    maxc1=c1[1]+T23[1];for(i=2;i<n;i++)
        {if(c1[i]+T23[i]>maxc1)maxc1=c1[i]+T23[i];}
        if(maxa1>=minb2||maxc1>=minb1)
        {for(i=1;i<=n;i++)
        {g[i]=abs(a1[i]-T12[i]-b1[i]-T23[i]);   h[i]=abs(c1[i]-T12[i]-b1[i]-T23[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]<<"\t"<<w[i]; cout<<endl;}
//To find minimum of G & H
        float g1[16],h1[16];for (i=1;i<=n;i++)
        if(g[i]<=h[i])
        { g1[i]=g[i]+w[i]; h1[i]=h[i];}
else{g1[i]=g[i]; h1[i]=h[i]+w[i]; }
float g2[16],h2[16];
        for(i=1;i<=n;i++){g2[i]=g1[i]/w[i]; h2[i]=h1[i]/w[i];}
        cout<<endl<<endl<<"displaying original scheduling table"<<endl;
for(i=1;i<=n;i++)
        {cout<<j[i]<<"\t"<<g2[i]<<"\t"<<h2[i]<<endl;}
        float mingh[16]; char ch[16]; for(i=1;i<=n;i++)
        {if(g2[i]<h2[i])
          {mingh[i]=g2[i]; ch[i]='g';}
else{ mingh[i]=h2[i]; ch[i]='h'; }}
for(i=1;i<=n;i++)
        {cout<<endl<<mingh[i]<<"\t"<<ch[i];}
for(i=1;i<=n;i++)
        {for(int k=1;k<=n;k++)
        if(mingh[i]<mingh[k])
```

```
{float temp=mingh[i]; int temp1=j[i]; char d=ch[i];mingh[i]=mingh[k]; j[i]=j[k]; ch[i]=ch[k];
mingh[k]=temp; j[k]=temp1; ch[k]=d;}}
for(i=1;i<=n;i++)
{ cout<<endl<<endl<<j[i]<<"\t"<<mingh[i]<<"\t"<<ch[i]<<"\n"; }
// calculate scheduling
float sbeta[16]; int t=1,s=0;for(i=1;i<=n;i++)
{if(ch[i]=='h')
{sbeta[(n-s)]=j[i]; s++;}
else if(ch[i]=='g')
{sbeta[t]=j[i];t++;}}
int arr1[16], m=1; cout<<endl<<endl<<"Job Scheduling:"<<"\t";
for(i=1;i<=n;i++){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++)
{f((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]];}
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;
if(time<=bd1 && macha[1]<=bd1||time>=bd2 && macha[1]>=bd2)
{a1[arr1[1]]=a1[arr1[1]];}
else{a1[arr1[1]]+=(bd2-bd1);}
if((macha[1]+T12[arr1[1]])<=bd1&&machb[1]<=bd1||(macha[1]+T12[arr1[1]])>=bd2&&machb[1]>=bd2)
{b1[arr1[1]]=b1[arr1[1]];}
else{b1[arr1[1]]+=(bd2-bd1);}
if((machb[1]+T23[arr1[1]])<=bd1&&machc[1]<=bd1||(machb[1]+T23[arr1[1]])>=bd2&&machc[1]>=bd2)
{c1[arr1[1]]=c1[arr1[1]];}
```

```
else{c1[arr1[1]]+=(bd2-bd1);}
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;
if(macha[i-1]<=bd1 && macha[i]<=bd1 || macha[i-1]>=bd2 && macha[i]>=bd2)
    {a1[arr1[i]]=a1[arr1[i]];}
else{a1[arr1[i]]+=(bd2-bd1);}
if(maxv1[i]<=bd1 && machb[i]<=bd1 || maxv1[i]>=bd2 && machb[i]>=bd2)
    {b1[arr1[i]]=b1[arr1[i]];}
else{b1[arr1[i]]+=(bd2-bd1);}
    if(maxv2[i]<=bd1 && machc[i]<=bd1 || maxv2[i]>=bd2 && machc[i]>=bd2)
    {c1[arr1[i]]=c1[arr1[i]];}
else{c1[arr1[i]]+=(bd2-bd1);}}
cout<<"\n\n\nTotal Elapsed Time (T) = "<<machc[n]; int j1[16];
for(i=1;i<=n;i++)
    { j1[i]=i;a11[arr1[i]]=a1[arr1[i]];b11[arr1[i]]=b1[arr1[i]];c11[arr1[i]]=c1[arr1[i]];}
cout<<endl<<"Modified Processing time after breakdown for the machines is:\n";
cout<<"Jobs"<<"\t"<<"Machine     M1"<<"\t"<<"\t"<<"Machine     M2"     <<"\t"<<"\t"<<"Machine
M3"<<"\t"<<"Weightage"<<endl;
for(i=1;i<=n;i++)
    {cout<<endl;cout<<j1[i]<<"\t"<<a11[i]<<"\t"<<b11[i]<<"\t"<<c11[i]<<"\t"<<w[i];cout<<endl;}
float maxa12,minb12,minb22,maxc12;float g12[16],h12[16];
//Function for two ficticious machine G and H //Finding largest in a11
    maxa12=a11[1]+T12[1];
for(i=2;i<n;i++)
    {if(a11[i]+T12[i]>maxa12)
    maxa12=a11[i]+T12[i];}
//For finding smallest in b11
    minb12=b11[1]+T23[1];
    for(i=2;i<n;i++)
    {if(b11[i]+T23[i]<minb12)
    minb12=b11[i]+T23[i];} minb22=b11[1]+T12[1];
    for(i=2;i<n;i++)
    {if(b11[i]+T12[i]<minb22)
    minb22=b11[i]+T12[i];}
//Finding largest in c12
    maxc12=c11[1]+T23[1];
    for(i=2;i<n;i++)
    {if(c11[i]+T23[i]>maxc12)
```

```
        maxc12=c11[i]+T23[i];}
        if(maxa12>=minb22||maxc12>=minb12)
        {for(i=1;i<=n;i++)
        { g12[i]=abs(a11[i]-T12[i]-b11[i]-T23[i]);h12[i]=abs(c11[i]-T12[i]-b11[i]-T23[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<<j1[i]<<"\t"<<g12[i]<<"\t"<<h12[i]<<"\t"<<w[i];cout<<endl;}
//To find minimum of G & H
        float g11[16],h11[16];
        for (i=1;i<=n;i++)
        if(g12[i]<=h12[i])
        {g11[i]=g12[i]+w[i];h11[i]=h12[i];}
else{g11[i]=g12[i];h11[i]=h12[i]+w[i];}
        float g21[16],h21[16];
for(i=1;i<=n;i++)
        {g21[i]=g11[i]/w[i];h21[i]=h11[i]/w[i];}
cout<<endl<<endl<<"displaying original scheduling table"<<endl;
        for(i=1;i<=n;i++)
        {cout<<j1[i]<<"\t"<<g21[i]<<"\t"<<h21[i]<<endl;}
        float mingh1[16];char ch1[16];
for(i=1;i<=n;i++)
        {if(g21[i]<h21[i])
          {mingh1[i]=g21[i];ch1[i]='g'; }
else{mingh1[i]=h21[i];ch1[i]='h';}}
        for(i=1;i<=n;i++)
        {cout<<endl<<i<<"\t"<<mingh1[i]<<"\t"<<ch1[i];}
        for(i=1;i<=n;i++)
        {for(int k=1;k<=n;k++)
        if(mingh1[i]<mingh1[k])
        {float temp=mingh1[i]; int temp1=j1[i]; char d=ch1[i];mingh1[i]=mingh1[k]; j1[i]=j1[k];
        ch1[i]=ch1[k];mingh1[k]=temp; j1[k]=temp1; ch1[k]=d;}}
for(i=1;i<=n;i++)
        {cout<<endl<<endl<<j1[i]<<"\t"<<mingh1[i]<<"\t"<<ch1[i]<<"\n";}
  // calculate   scheduling
        float sch[16];int d=1,f=0;
for(i=1;i<=n;i++)
        {if(ch1[i]=='h')
        {sch[(n-f)]=j1[i];f++;}
```

```
else  if(ch1[i]=='g')
        {sch[d]=j1[i];d++;}}
        int arr2[16], y=1;cout<<endl<<endl<<"Job Scheduling:"<<"\t";
for(i=1;i<=n;i++)
        {cout<<sch[i]<<" ";arr2[y]=sch[i];y++;}
//calculating total computation sequence;
        float time1=0.0 ;float maxv11[16],maxv21[16];
        macha1[1]=time1+a11[arr2[1]];
for(i=2;i<=n;i++)
        {macha1[i]=macha1[i-1]+a11[arr2[i]];}
        machb1[1]=macha1[1]+b11[arr2[1]]+T12[arr2[1]];
for(i=2;i<=n;i++)
        {if((machb1[i-1])>(macha1[i]+T12[arr2[i]]))
        maxv11[i]=machb1[i-1];
else maxv11[i]=macha1[i]+T12[arr2[i]];machb1[i]=maxv11[i]+b1[arr2[i]];}
        machc1[1]=machb1[1]+c11[arr2[1]]+T23[arr2[1]];
for(i=2;i<=n;i++)
        {if((machc1[i-1])>(machb1[i]+T23[arr2[i]]))
        maxv21[i]=machc1[i-1];
else maxv21[i]=machb1[i]+T23[arr2[i]];machc1[i]=maxv21[i]+c1[arr2[i]];}
        float wft,sum1,sum2;sum2=0.0;
for(i=1;i<=n;i++)
        {sum2=sum2+w[i];}
///displaying solution
cout<<"\n\n\n\n\n\t\t\t        #####THE SOLUTION##### ";
cout<<"\n\n\t***************************************************************";
cout<<"\n\n\n\t        Optimal Sequence is : ";
for(i=1;i<=n;i++) {cout<<" "<<arr2[i];}
cout<<endl<<endl<<"In-Out Table is:"<<endl<<endl;
cout<<"Jobs"<<"\t"<<"Machine M1"<<"\t"<<"\t"<<"Machine M2" <<"\t"<<"\t"<<"Machine M3"<<endl;
cout<<arr2[1]<<"\t"<<time1<<"--"<<macha1[1]<<"\t"<<"\t"<<macha1[1]+T12[arr2[1]]<<"--"<<machb1[
1]<<" \t"<<"\t"<<machb1[1]+T23[arr2[1]]<<"--"<<machc1[1]<<endl; sum1=0.0;
for(i=2;i<=n;i++)
        {cout<<arr2[i]<<"\t"<<macha1[i-1]<<"--"<<macha1[i]<<""<<"\t"<<maxv11[i]<<"--"<<machb1[i]<<
" "<<"\t"<<maxv21[i]<<"--"<<machc1[i]<<endl; sum1=sum1+(machc1[i]-macha1[i-1])*w[i];}
cout<<"\n\n\nTotal Elapsed Time (T) = "<<machc1[n]; wft=((machc1[1]*w[1])+sum1)/sum2;
cout<<"\n\n\n The mean weighted flow time is = "<<wft;
cout<<"\n\n\t***************************************************************";
getch();}
```

7. Numerical Illustartion

Consider the following flow shop scheduling problem of 5 jobs and 3 machines problem in which the processing time with their corresponding probabilities, transportation time and weight of jobs is given as below:

Jobs	Machine M_1		$T_{i,1\to 2}$	Machine M_2		$T_{i,2\to 3}$	Machine M_3		Weightage
i	a_{i1}	p_{i1}		a_{i2}	p_{i2}		a_{i3}	p_{i3}	w_i
1	80	0.2	2	60	0.3	1	120	0.1	2
2	120	0.1	3	70	0.2	3	60	0.2	4
3	50	0.2	1	110	0.1	2	70	0.2	3
4	70	0.2	2	50	0.2	4	40	0.3	5
5	40	0.3	4	60	0.2	2	50	0.2	1

Find optimal or near optimal sequence when the break down interval is (a, b) = (30, 35). Also calculate the total elapsed time and mean weighted flow time.

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

As per Step 2; Here Max $\{A_{i1}+T_{i.1\to 2}\}$ =18, Min $\{A_{i2}+T_{i,1\to 2}\}$ =12, Max $\{T_{i,2\to 3}+A_{i3}\}$ =16, Min $\{A_{i2}+T_{i,2\to 3}\}$ =13.Therefore, we have

Max $\{A_{i1}+T_{i.1\to 2}\}\geq$ Min $\{A_{i2}+T_{i,1\to 2}\}$ and Max $\{T_{i,2\to 3}+A_{i3}\}\geq$ Min $\{A_{i2}+T_{i,2\to 3}\}$.

As per Step. 3; The two fictitious machines G and H with processing times G_i and H_i are as in **table 2**.

As per Step 4 &5; the new reduced problem with G_i'' and H_i'' is as in **table 3.**

As per Step 8; The optimal sequence with minimum elapsed time using Johnson's technique is

S =1 – 5 – 2 –3 – 4.

As per Step 9 & 10; The In-Out flow table and checking the effect of break down interval (30, 35) on sequence S, is as in **table 5**

As per Step 11; On considering the effect of the break down interval the original problem reduces to as in **table 6.**

Now, On repeating the procedure to get the optimal sequence for the modified scheduling problem, we get the sequence 2 – 1 – 5 – 3 – 4 which is optimal or near optimal. The In-Out flow table for the modified scheduling problem is: as in **table 7**.

The mean weighted flow time = $\dfrac{49\times 4+(71-17)\times 2+(82-33)\times 1+(97-45)\times 3+(109-55)\times 5}{5+3+2+4+1}=51.666$

Hence the total elapsed time is 109 hrs and the mean weighted flow time is 51.666 hrs.

Conclusion

The new method provides a scheduling optimal sequence with minimum total elapsed time whenever mean weighted production flow time is taken into consideration for 3-machines, n-jobs flow shop scheduling problems. This method is very easy to understand and will help the decision makers in determining a best schedule for a given sets of jobs to control job flow effectively and provide a solution for job sequencing. The study may further be extended by introducing the concept of Setup time, Job block criteria and Rental policy.

References

Baker, K. R. (1974), "Introduction of sequencing and scheduling," *John Wiley and Sons*, New York.

Bellman, R. (1956), "Mathematical aspects of scheduling theory", *J. Soc. Indust. Appl. Math*. 4(3), 168-205.

Belwal & Mittal (2008), "n jobs machine flow shop scheduling problem with break down of machines, transportation time and equivalent job block", *Bulletin of Pure & Applied Sciences-Mathematics,* Jan – June, 2008, source Vol. 27, Source Issue 1.

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

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

Jackson, J. R. (1956), "An extension of Johnson's results on job scheduling", *Nav. Res. Log. Quar*., 3, pp 201-203.

Johnson, S.M. (1954), "Optimal two stage and three stage production schedule with set-up times included", *Nav. Res. Log. Quar*.1(1), pp 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), pp 598-601.

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

Miyazaki, S. & Nishiyama, N. (1980), "Analysis for minimizing weighted mean flow time in flow shop scheduling", *J. O. R. Soc. Of Japan*, 32, pp 118-132.

Narain, L. & Bagga, P.C. (1998), "Two machine flow shop problem with availability constraint on each machine", *JISSOR* XXIV (1-4), pp 17-24.

Nawaz, Ensore & Ham (1983), "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem", *OMEGA*, pp 91-95.

Parker, R.G.. (1995), "Deterministic scheduling theory", *Chapman and Hall*, New York, 1995.

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

Singh,T.P.(1995), "On 2 x n flow-shop problems involving job-block, transportation time, arbitrary time and break down machine time", *PAMS* XXI(1-2), March.

Notes

Note 1. The example discussed here can not be solved using algorithm given in Pandian & Rajendran [2010] as any of the structural conditions are not satisfied.

Table 1. The expected processing times for the machines M_1, M_2 and M_3 are

Jobs	A_{i1}	$T_{i,1\to2}$	A_{i2}	$T_{i,2\to3}$	A_{i3}	w_i
1	16	2	18	1	12	2
2	12	3	14	3	12	4
3	10	1	11	2	14	3
4	14	2	10	4	12	5
5	12	4	12	2	10	1

Table 2 The two fictitious machines G and H with processing times G_i and H_i are

Jobs	G_i	H_i	w_i
1	5	9	2
2	8	8	4
3	4	0	3
4	2	4	5
5	7	8	1

Table 3. The new reduced problem with $G_i^{''}$ and $H_i^{''}$ is

Jobs	$G_i^{''}$	$H_i^{''}$
1	3.5	4.5
2	3	2
3	1.33	1
4	1.4	0.8
5	6	8

Table 5. The In-Out flow table and checking the effect of break down interval (30, 35) on sequence S ,is

Jobs	Machine M_1	$T_{i,1\to 2}$	Machine M_2	$T_{i,2\to 3}$	Machine M_3	w_i
i	In – Out		In – Out		In - Out	
1	0 – 16	2	**18 – 36**	1	37 – 49	**2**
5	16 – 28	4	36 – 48	2	50 – 60	1
2	**28 – 40**	3	48 – 62	3	65 – 77	4
3	40 – 50	1	62 – 73	2	77 – 91	3
4	50 – 64	2	73 – 83	2	91 – 103	5

Table 6. The new reduced problem on considering the effect of the break down interval

Jobs	A_{i1}	$T_{i,1\to 2}$	A_{i2}	$T_{i,2\to 3}$	A_{i3}	w_i
1	16	2	23	1	12	2
2	17	3	14	3	12	4
3	10	1	11	2	14	3
4	14	2	10	4	12	5
5	12	4	12	2	10	1

Table 7. . The In-Out flow table for the modified scheduling problem is

Jobs	Machine M_1	$T_{i,1\to 2}$	Machine M_2	$T_{i,2\to 3}$	Machine M_3	w_i
i	In – Out		In – Out		In - Out	
2	0 – 17	3	20 – 34	3	37 – 49	4
1	17 – 33	2	35 – 58	1	59 – 71	2
5	33 – 45	4	58 – 70	2	72 – 82	1
3	45 – 55	1	70 – 81	2	83 – 97	3
4	55 – 69	2	81 – 91	4	97 – 109	5