Minimizing Rental Cost for n-Jobs, 2-Machines Flow Shop Scheduling, Processing Time Associated With Probabilities Including Transportation Time and Job Block criteria

Sameer Sharma

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

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

Abstract

This paper deals with a heuristic algorithm to minimize the rental cost of the machines for two stage flow shop scheduling problem under specified rental policy in which processing times are associated with their respective probabilities whereas the transportation time from one machine to another machine and an equivalent job block criteria is being considered. The purposed algorithm is easy to understand and provide an important tool for the decision makers. A computer program followed by a numerical illustration is also given to justify the algorithm.

Keywords: Equivalent job, Flow Shop, Rental Policy, Transportation Time, Elapsed Time.

1. Introduction

Scheduling problems concern with the situation in which value of the objective function depends on the order in which tasks have to be performed. A lot of research work has been done in the area of scheduling problems for different situations and different criterions. Johnson [1] gave procedure for finding the optimal schedule for n-jobs, two machine flow-shop problem with minimization of the makespan (i.e. total elapsed time) as the objective. Ignall and Scharge [2] applied Branch and Bound technique for obtaining a sequence which minimizes the total flow time. Chandrasekharan [3] has given a technique based on Branch and Bound method and satisfaction of criterion conditions to obtain a sequence which minimizes total flow-time subject to minimum makespan in a two stage flow shop problem. Bagga P.C.[4], Maggu and Das [5], Szwarch [6], Yoshida & Hitomi [7], Singh T.P.[8], Chandra Sekhran [9], Anup [10], Gupta Deepak [11] etc. derived the optimal algorithm for two/ three or multistage flow shop problems taking into account the various constraints and criteria. Maggu and Das [5] introduced the concept of job-block criteria in the theory of scheduling. This concept is useful and significant in the sense to create a balance between the cost of providing priority in service to the customer and cost of giving services with non-priority customers. The decision maker may decide how much to charge extra to priority customers.

Singh T.P., Gupta Deepak [13] studied n×2 general flowshop problem to minimize rental cost under a predefined rental policy in which the probabilities have been associated with processing time on each machine including job block criteria. In this paper we have extended the study made by Singh T.P., Gupta Deepak [13] by introducing the concept of transportation time. Here we have developed an algorithm to minimize the rental cost of the machines. The problem discussed here is wider and has significant use of theoretical results in process industries.

2. Practical Situations

Various practical situations occur in real life when one has got the assignments but does not have one's own machine or does not have enough money or does not want to take risk of investing huge amount of money to purchase machine. Under such circumstances, the machine has to be taken on rent in order to complete the assignments. As a medical practitioner, in the starting of his career, does not buy expensive machines say X-ray machine, the ultra sound machine etc. but instead take them on rent. Moreover in hospitals/industries 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 become significant. Further, when the machines on which jobs are to be processed are planted at different places, the transportation time which include the loading time, moving time and unloading time etc. has a significant role in production concern and hence significant.

3. Notations

S : Sequence of jobs 1,2,3,….,n

M_j : Machine j, j= 1,2,…….

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 i^{th} job on machine A.

q_i : Probability associated to the processing time B_i of i^{th} job on machine B.

S_i : Sequence obtained from Johnson's procedure to minimize rental cost.

$t_{Ai\to Bi}$: Transportation time from machine A to machine B.

C_j : Rental cost per unit time of machine j.

U_i :Utilization time of B (2^{nd} machine) for each sequence S_i

$t_1(S_i)$:Completion time of last job of sequence S_i on machine A.

$t_2(S_i)$: Completion time of last job of sequence S_i on machine B.

$R(S_i)$: Total rental cost for sequence S_i of all machines.

$CT(S_i)$:Completion time of 1^{st} job of each sequence S_i on machine A.

4. Problem Formulation

Let *n* jobs say i=1,2,3…n be processed on two machines A & B in the order AB. A job i (i=1,2,3…n) has processing time A_i & B_i on each machine respectively, assuming their respective probabilities p_i & q_i such that $0\le p_i \le 1$ & $\Sigma p_i = 1$, $0 \le q_i \le 1$ & $\Sigma q_i=1$ and let $t_{Ai\to Bi}$ be the transportation time from machine A to machine B of each job i. Let an equivalent job β is defined as (k, m) where k, m are any jobs among the given *n* jobs such that k occurs before job m in the order of job block (k , m). The mathematical model of the problem in matrix form can be stated as :

jobs	Machine A		$t_{Ai \to Bi}$	Machine B	
i	A_i	p_i		B_i	q_i
1	A_1	p_1	$t_{A1 \to B1}$	B_1	q_1
2	A_2	p_2	$t_{A2 \to B2}$	B_2	q_2
3	A_3	p_3	$t_{A3 \to B3}$	B_3	q_3
4	A_4	p_4	$t_{A4 \to B4}$	B_4	q_4
---	---	---	---	---	---
---	---	---	---	---	---
n	A_n	p_n	$t_{An \to Bn}$	B_n	q_n

Tableau – 1

Our objective is to find the optimal schedule of all jobs which minimize the total rental cost, when costs per unit time for machines A & B are given while minimizing the makespan.

5. Assumptions

1. We assume **rental policy** that all the machines are taken on rent as and when they are required and are returned as when they are no longer required for processing. Under this policy second machine is taken on rent at time when first job completes its processing on first machine. Therefore idle time of second machine for first job is zero.
2. Jobs are independent to each other.
3. Machine break down is not considered.
4. Pre- emption is not allowed i.e. once a job started on a machine, the process on that machine can't be stopped unless the job is completed.
5. It is given to sequence k jobs $i_1, i_2 \ldots i_k$ as a block or group-job in the order $(i_1, i_2 \ldots i_k)$ showing priority of job i_1 over i_2
6. Jobs may be held in inventory before going to a machine.

6. Algorithm

To obtain optimal schedule, we proceed as

Step 1. Define expected processing time A'_i & B'_i on machine A & B respectively as follows:

$A'_i = A_i * p_i$, $B'_i = B_i * q_i$

Step 2. Define two fictitious machines G & H with processing time G_i & H_i for job i on machines G & H respectively, as:

$G_i = A'_i + t_{Ai \to Bi}$, $H_i = t_{Ai \to Bi} + B'_i$

Step 3. Take equivalent job $\beta = (k, m)$ and define processing time as follows:

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

Step 4. Define a new reduced problem with processing time G_i & H_i where job block (k , m) is replaced by single equivalent job β with processing time G_β & H_β as obtained in step 3.

Step 5. Apply Johnson's [1] technique and obtain an optimal schedule of given jobs, using Johnson's technique. Let the sequence be S_1.

Step 6 : Observe the processing time of 1^{st} job of S_1 on the first machine A . Let it be α.

Step 7 : Obtain all the jobs having processing time on A greater than α. Put these job one by one in the 1^{st} position of the sequence S_1 in the same order. Let these sequences be S_2, S_3, S_4,…S_r

Step 8 : Prepare in-out table for each sequence S_i (i=1,2,…r) and evaluate total completion time of last job of each sequence $t_1(S_i)$ & $t_2(S_i)$ on machine A & B respectively.

Step 9 : Evaluate completion time $CT(S_i)$ of 1^{st} job for each sequence S_i on machine A.

Step 10: Calculate utilization time U_i of 2^{nd} machine for each sequence S_i as:

$U_i = t_2(S_i) - CT(S_i)$ for i=1,2,3,…r.

Step 11: Find Min $\{U_i\}$, i=1,2,…r. let it be corresponding to i=m,then S_m is the optimal sequence for minimum rental cost.

Min rental cost = $t_1(S_m) \times C_1 + U_m \times C_2$

Where C_1 & C_2 are the rental cost per unit time of 1^{st} & 2^{nd} machine respectively.

7. Computer Program

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<process.h>
void display();
void schedule(int,int);
void inout_times(int []);
void update();
void time_for_job_blocks();

float min;
int job_schedule[16],tt[16];int job_schedule_final[16];int n;
float a1[16],b1[16],a11[16],b11[16];float a1_jb,b1_jb;float a1_temp[15],b1_temp[15];
int job_temp[15];int group[2];//variables to store two job blocks
float a1_t[16], b1_t[16];float a1_in[16],a1_out[16];float b1_in[16],b1_out[16];
float ta[16]={32767,32767,32767,32767,32767},tb[16]={32767,32767,32767,32767,32767};
void main()
{
	clrscr();
	int a[16],b[16];float p[16],q[16];int optimal_schedule_temp[16];int optimal_schedule[16];
	float cost_a,cost_b,cost;float min; //Variables to hold the processing times of the job blocks
	cout<<"How many Jobs (<=15) : ";cin>>n;
if(n<1 || n>15)
	{cout<<"Wrong input, No. of jobs should be less than 15..\n Exitting";getch();
```

```
        exit(0);
    }
    cout<<"Enter the processing time and their respective probabilities ";
    for(int i=1;i<=n;i++)
{cout<<"\nEnter the processing time and its probability of "<<i<<" job for machine A : ";cin>>a[i]>>p[i];
cout<<"\nEnter the transportation time of "<<i<<"job from machine A to B : ";cin>>tt[i];
cout<<"\nEnter the processing time and its probability of "<<i<<" job for machine B : ";cin>>b[i]>>q[i];
//Calculate the expected processing times of the jobs for the machines:
    a11[i] = a[i]*p[i];b11[i] = b[i]*q[i];a1[i]=a11[i]+tt[i];b1[i]=b11[i]+tt[i];
    }
    for(int k =1;k<=n;k++)
    {cout<<"\n"<<k<<"\t\t"<< a1[k]<<"\t\t"<< b1[k];}
cout<<"\nEnter the two job blocks (two numbers from 1 to "<<n<<") : ";cin>>group[0]>>group[1];
cout<<"\nEnter the Rental cost of machine A : ";cin>>cost_a;
cout<<"\nEnter the Rental cost of machine B : ";cin>>cost_b;
//Function for expected processing times for two job blocks
    time_for_job_blocks();int t = n-1;
    schedule(t,1);
//Calculating In-Out times
    inout_times(job_schedule_final);
    //REpeat the process for all possible sequences
    for( k=1;k<=n;k++)     //Loop of all possible sequences
    {
        for(int i=1;i<=n;i++)
        {optimal_schedule_temp[i]=job_schedule_final[i];}
        int temp = job_schedule_final[k];optimal_schedule_temp[1]=temp;
        for(i=k;i>1;i--)
        {optimal_schedule_temp[i]=job_schedule_final[i-1];}
//Calling inout_times()
        int flag=0;
        for(i=1;i<n;i++)
        {
            if(optimal_schedule_temp[i]==group[0] && optimal_schedule_temp[i+1]==group[1])
            {flag=1;break;}
        }
        if(flag==1)
        {
            inout_times(optimal_schedule_temp);
            ta[k]=a1_out[n]-a1_in[1];tb[k]=b1_out[n]-b1_in[1];
```

```
                if(tb[k]<tb[k+1])
                {
                    //copy optimal_schedule_temp to optimal_schedule
                    for(int j=1;j<=n;j++)
                    {
                        optimal_schedule[j]=optimal_schedule_temp[j];
                    }}}}

    float smalla = ta[1];float smallb = tb[1];float maxv[16];
    for(int ii=2;ii<=n;ii++)
    {
        if(smalla>ta[ii])
            smalla = ta[ii];
        if(smallb>tb[ii])
            smallb = tb[ii];
    }
    clrscr();
    cout<<"\n\n\n\n\n\n\n\n\t\t\t       #####THE SOLUTION##### ";
    cout<<"\n\n\t******************************************************************";
    cout<<"\n\n\n\t       Optimal Sequence is : ";
    for (ii=1;ii<=n;ii++)
    {cout<<optimal_schedule[ii]<<"      ";}
    cout<<"\n\n\t       The smallest possible time span for machine A is : "<<smalla;
    cout<<"\n\n\t       The smallest possible time span for machine B is : "<<smallb;
    cost = cost_a*smalla+cost_b*smallb;
    cout<<"\n\n\t       Total Minimum Rental cost for both the machines is : "<<cost;
    cout<<"\n\n\n\t******************************************************************";
    getch();
}

void time_for_job_blocks()
{
    //The expected processing times for two job blocks are
    if(b1[group[0]]<a1[group[1]])
    {min = b1[group[0]];}
    else
    {min = a1[group[1]];}
    a1_jb = a1[group[0]]+a1[group[1]] - min; //(b1[k]<a1[m])?b1[k]:a1[m];
    b1_jb = b1[group[0]]+b1[group[1]] - min; //(b1[k]<a1[m])?b1[k]:a1[m];
```

```
    cout<<"/n abeta = "<<a1_jb<<"\n bbeta ="<<b1_jb;
    getch();
}

void inout_times(int schedule[])
{
    for(int i=1;i<=n;i++)
    {
        //Reorder the values of a1[] and b1[] according to sequence
        a1_t[i] = a11[schedule[i]];b1_t[i] = b11[schedule[i]];
    cout<<"\n"<<schedule[i]<<"\t\t"<<a1_t[i]<<"\t\t"<<b1_t[i];
   }
    for(i=1;i<=n;i++)
    {
        if(i==1)
        {a1_in[i]=0.0;  a1_out[i] = a1_in[i]+a1_t[i];b1_in[i] = a1_out[i]+tt[schedule[i]];
            b1_out[i] = b1_in[i]+b1_t[i];}
        else
        {
            a1_in[i]=a1_out[i-1];a1_out[i] = a1_in[i]+a1_t[i];
            if(b1_out[i-1]>=(a1_out[i]+tt[schedule[i]]))
            {b1_in[i] = b1_out[i-1];b1_out[i] = b1_in[i]+b1_t[i];}
            else
            {b1_in[i] = a1_out[i]+tt[schedule[i]];b1_out[i] = b1_in[i]+b1_t[i];
            }}}}
int js1=1,js2=n-1;
void schedule(int t, int tt)
{
    if(t==n-1)
    {js1=1; js2=n-1;}
    if(t>0 && tt==1)
    {
        for(int i=1,j=1;i<=n;i++,j++) //loop from 1 to n-1 as there is one group
        {if(i!=group[0]&&i!=group[1])
            {a1_temp[j] = a1[i];b1_temp[j] = b1[i];job_temp[j] = i;}
            else if(group[0]<group[1] && i==group[0])
            {a1_temp[j] = a1_jb;b1_temp[j] = b1_jb;job_temp[j] = -1;}
            else
            {    j--;}}
```



```
        //Finding smallest in a1
        float min1= 32767;
        int pos_a1;
        for(j=1;j<n;j++)
        {
            if(min1>a1_temp[j])
            {pos_a1 = j;min1 = a1_temp[j];}
        }
        //Finding smallest in b1
        float min2= 32767;int pos_b1;
        for(int k=1;k<n;k++)
        {if(min2>b1_temp[k])
            {pos_b1 = k;min2 = b1_temp[k]      }}
        if(min1<min2)
    {job_schedule[js1] = job_temp[pos_a1];js1++;a1_temp[pos_a1]=32767;b1_temp[pos_a1]=32767;}
        else
    {job_schedule[js2] = job_temp[pos_b1];js2--;a1_temp[pos_b1]=32767;b1_temp[pos_b1]=32767;
        }}
    else if(t>0 && tt!=1)
    {//Finding smallest in a1
        float min1= 32767;int pos_a1;
        for(int i=1;i<n;i++)
        {if(min1>a1_temp[i])
            {pos_a1 = i;min1 = a1_temp[i];
            }}
        //Finding smallest in b1
        float min2= 32767;int pos_b1;
        for(i=1;i<n;i++)
        {if(min2>b1_temp[i])
            {pos_b1 = i;min2 = b1_temp[i];
            }}
        if(min1<min2)
    {job_schedule[js1] = job_temp[pos_a1];js1++;a1_temp[pos_a1]=32767;b1_temp[pos_a1]=32767;}
        else
    {job_schedule[js2] = job_temp[pos_b1];js2--;a1_temp[pos_b1]=32767;b1_temp[pos_b1]=32767;}}
    t--;
    if(t!=0)
    {schedule(t, 2);}
    //final job schedule
```

```
int i=1;
while(job_schedule[i]!=-1)
{job_schedule_final[i]=job_schedule[i];i++;}
job_schedule_final[i]=group[0];i++;
job_schedule_final[i]=group[1];i++;
while(i<=n)
{job_schedule_final[i]=job_schedule[i-1];i++;}}
```

8. Numerical Illustration

Consider 5 jobs and 2 machines problem to minimize the rental cost. The processing times with their respective probabilities and transportation time from one machine to another machine are given as follows:

job i	Machine A		$t_{Ai\rightarrow Bi}$	Machine B	
	A_i	p_i		B_i	q_i
1	12	0.2	6	8	0.1
2	16	0.3	5	12	0.2
3	13	0.3	4	14	0.3
4	18	0.1	3	17	0.2
5	15	0.1	4	18	0.2

Tableau-1

Rental costs per unit time for machines M_1 & M_2 are 15 & 13 units respectively, and jobs 2, 5 are to be processed as an equivalent group job β. Also $\sum p_i=1$, $\sum q_i=1$.

Solution:

As per step 1 expected processing times are as under:

Jobs	A_i'	$t_{Ai\rightarrow Bi}$	B_i'
1	2.4	6	0.8
2	4.8	5	2.4
3	3.9	4	4.2
4	1.8	3	3.4
5	1.5	4	3.6

Tableau-2

As per step 2 :

Calculate $G_i= A'_i + t_{Ai\rightarrow Bi}$ & $H_i= t_{Ai\rightarrow Bi} + B'_i$

Jobs	G_i	H_i
1	8.4	6.8
2	9.8	7.4
3	7.9	8.2
4	4.8	6.4
5	5.5	7.6

Tableau-3

As per step 3 :

the processing times of equivalent job block β = (2,5) are given by

G_β = 9.8+5.5-5.5= 9.8 and H_β = 7.4+7.6 -5.5=9.5

Jobs	G_i	H_i
1	8.4	6.8
β	9.8	9.5
3	7.9	8.2
4	4.8	6.4

Tableau-4

As per step 4 :

Using Johnson's method optimal sequence is

S_1 = 4,3, β,1

i.e. 4-3-2-5-1

Other optimal sequences for minimize rental cost, are

S_2 =1-4-3-2-5

S_3 = 3-4-2-5-1

S_4 = 2-5-4-3-1

In-out table for these sequences are given from tableau-5 to tableau-8:

S_1 = 4-3-2-5-1

Jobs	A	$t_{Ai \to Bi}$	B
	In-Out		In-Out
4	0-1.8	3	4.8-8.2
3	1.8-5.7	4	9.7-13.9
2	5.7-10.5	5	15.5-17.9
5	10.5-12	4	17.9-21.5
1	12-14.4	6	21.5-22.3

Tableau-5

Thus the total elapsed time = 22.3 units and

utilization time for M_2 = 22.3-4.8

=17.5 units.

S_2=1-4-3-2-5

Jobs	A	$t_{Ai \to Bi}$	B
	In-Out		In-Out
1	0-2.4	6	3.0-3.8
4	2.4-4.2	3	7.2-10.6
3	4.2-8.1	4	12.1-16.3
2	8.1-12.9	5	17.9-20.3
5	12.9-14.4	4	20.3-23.9

Tableau-6

Total elapsed time = 23.9 units

Utilization time of B = 23.9- 3.0 =20.9 units

REMARKS:

i. The following algebraic properties can be easily proved with the numerical examples:

a) Equivalent job formation is associative in nature

i.e. the block ((1,3)5) =(1(3.5)).

b) The equivalent job formation rule is non commutative

i.e. the block (1,5) ≠ (5,1).

ii. The study may be extended further for three machines flow shop, also by considering various parameters such as break down interval etc.

References

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.

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

Chandrasekharan Rajendaran [1992], "*Two-Stage Flowshop Scheduling Problem with Bicriteria* " O.R. Soc. Vol. 43, No. 9, 871-84.

P.C.Bagga[1969], *Sequencing in a rental situations*, Journal of Canadian Operation Research Society 7 ,152-153

Maggu P. L and Das G.[1977], *equivalent jobs for job block in job sequencing*, Opsearch, Vol 14, No. 4, 277-281.

Szware, W [1977], *Special cases of the flow shop problems*, Naval Research Log,Quartly 24,403-492.

Yoshida and Hitomi[1979], *optimal two stage production scheduling with set up times separated.* AIIETransactions. Vol. II.pp 261-263.

Singh T.P [1985], *on n×2 shop problem involving job block. Transportation times and Break-down Machine times*, PAMS Vol. XXI No. 1-2