ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

A Mathematical Model for Bicriteria in Constrained Three Stage Flow Shop Scheduling

Deepak Gupta, Sameer Sharma, Seema Sharma, Shefali Aggarwal

Abstract— **This paper presents a bicriteria in n-jobs, three machines flow shop scheduling to minimize the total elapsed time and rental cost of the machines taken on rent under a specified rental policy in which the processing time, independent setup time each associated with probabilities including transportation time and job block criteria. Further the concept of the break down interval for which the machines are not available for the processing is included. A heuristic approach to find optimal or near optimal sequence has been discussed. A computer programme followed by a numerical illustration is given to substantiate the algorithm.**

Index Terms— **Break-down interval, Job block criteria, Transportation time, Rental Cost.**

I. INTRODUCTION

Scheduling is broadly defined as the process of the allocation of resources over time to perform a collection of tasks. Scheduling problems in their simple static and deterministic forms are extremely simple to describe and formulate, but are difficult to solve because they involve complex combinatorial optimization. For example, if n jobs are to be performed on m machines, there are potentially $(n!)^m$ sequences, although many of these may be infeasible due to various constraints. Single criterion is deemed as insufficient for real and practical applications. Thus considering problems with more than one criterion is a practical direction of research for real-life scheduling problems. The bicriteria scheduling problems are motivated by the fact that they are more meaningful from practical point of view. The classical scheduling literature commonly assumes that the machines are never unavailable during the process. But there are feasible sequencing situations where machines while processing the jobs get sudden break-down due to failure of a component of machines 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 stop of flow of electric current to the machines by a government policy due to shortage of electricity production. In each case this may be well observed that working of machines is not continuous and is subject to breakdown for certain interval of time. The majority of scheduling research assumes setup as negligible or part of processing time. While this assumption adversely affects solution quality for many applications which require explicit treatment of setup. Such applications have motivated increasing interest to include setup considerations in scheduling theory. One of the earliest results in flow shop scheduling theory is an algorithm given by Johnson [1] for scheduling jobs in a two machine flowshop to minimize the time at which all jobs are completed. Smith [2] considered minimization of mean flow time and maximum tardiness. Some of the noteworthy heuristic approaches are due to Maggu & Das [3], Yoshida &Hitomi [4], Adiri [5], Singh T.P. [6], Akturk & Gorgulu [7], Brucker and S.Knust [8], Chandramouli [9], Chikhi [10], Belwal and Mittal [11], Khodadadi [12], Pandian and Rajendran [13] by considering various parameters. Gupta, Sharma and Seema [15] studied bicriteria in n x 3 flow shop scheduling under specified rental policy, processing time associated with probabilities including transportation time and job block criteria. We have extended the study made by Gupta and Sharma [15] by introducing the concept of setup time and breakdown interval. This paper considers a more practical scheduling situation in which certain ordering of jobs is prescribed either by technological constraints or by externally imposed policy.

II. PRACTICAL SITUATION

Many applied and experimental situations exist in our day-to-day working in factories and industrial production concerns etc. 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. 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. In his starting career, we find a medical practitioner does not buy expensive machines

Manuscript received Oct 15, 2011.
Deepak Gupta*, Prof. & Head, Department of Mathematics, Maharishi Markandeshwar University, Mullana, Ambala, Haryana, India (e-mail: guptadeepak2003@yahoo.co.in).*
Sameer Sharma*, Department of Mathematics, D. A. V. College, Jalandhar City, Punjab, India (e-mail: samsharma31@yahoo.com).*
Seema Sharma*, Department of Mathematics, D. A. V. College, Jalandhar City, Punjab, India (e-mail: seemasharma7788@yahoo.com).*
Shefali Aggarwal*, Research Scholar, Department of Mathematics, Maharishi Markandeshwar University, Mullana, Ambala, Haryana, India (e-mail: shefaliaggarwalshalu@gmail.com).*

say X-ray machine, the Ultra Sound Machine, Rotating Triple Head Single Positron Emission Computed Tomography Scanner, Patient Monitoring Equipment, and Laboratory Equipment etc., but instead takes on rent. Rental of medical equipment is an affordable and quick solution for hospitals, nursing homes, physicians, which are presently constrained by the availability of limited funds due to the recent global economic recession. Renting enables saving working capital, gives option for having the equipment, and allows upgradation to new technology. Further the priority of one job over the other may be significant due to the relative importance of the jobs. It may be because of urgency or demand of that particular job. Hence, the job block criteria become important. Another event which is mostly considered in the models is the break-down of machines. There may also be delays due to material, changes in release and tail dates, tools unavailability, failure of electric current, the shift pattern of the facility and fluctuations in processing times. All of these events complicate the scheduling problem in most cases. Hence the criterion of break-down interval becomes significant.

III. NOTATIONS

S : Sequence of jobs 1,2,3,….,n
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}
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
L : Length of the break-down interval
A'_{ij} :Expected processing time of i^{th} job after break-down effect on machine M_j
β : Equivalent job for job – block
C_i : Rental cost of i^{th} machine
$L_j(S_k)$: The latest time when machine M_j is taken on rent for sequence S_k
$t_{ij}(S_k)$: Completion time of i^{th} job of sequence S_k on machine M_j
$t'_{ij}(S_k)$: Completion time of i^{th} job of sequence S_k on machine M_j when machine M_j start processing jobs at time $L_j(S_k)$
$T_{i,j\to k}$: Transportation time of i^{th} job from j^{th} machine to k^{th} machine
$I_{ij}(S_k)$: Idle time of machine M_j for job i in the sequence S_k
$U_j(S_k)$:Utilization time for which machine M_j is required, when M_j starts processing jobs at time $L_j(S_k)$
$R(S_k)$: Total rental cost for the sequence S_k of all machine

IV. RENTAL POLICY

The machines will be taken on rent as and when they are required and are returned as and when they are no longer required i.e. the first machine will be taken on rent in the starting of the processing the jobs, 2nd machine will be taken on rent at time when 1st job is completed on 1st machine and transported to 2nd machine, 3rd machine will be taken on rent at time when 1st job is completed on the 2nd machine and transported.

4.1 Definition: Completion time of i^{th} job on machine M_j is denoted by t_{ij} and is defined as

$t_{ij} = max\ (t_{i-1,j} + s_{(i-1),j} \times q_{(i-1),j}\ ,\ t_{i,j-1}) + T_{i,(j-1)\to j} + a_{ij} \times p_{ij}$ for $j \geq 2$.

$= max\ (t_{i-1,j} + S_{i-1,j}\ ,\ t_{i,j-1}) + T_{i,(j-1)\to j} + A_{i,j}$

where $A_{i,j}$= expected processing time of i^{th} job on machine j.

$S_{i,j}$= expected set up time of i^{th} job on machine j.

4.2 Definition:

Completion time of i^{th} job on machine M_j when M_j starts processing jobs at time L_j is denoted by $t'_{i,j}$ and is defined a

$$t'_{i,j} = L_j + \sum_{k=1}^{i} A_{k,j} + \sum_{k=1}^{i-1} S_{k,j} = \sum_{k=1}^{i} I_{k,j} + \sum_{k=1}^{i} A_{k,j} + \sum_{k=1}^{i-1} S_{k,j}$$

Also, $t'_{i,j} = \max(t'_{i,j-1} + S_{i-1,j}, t'_{i-1,j}) + A_{i,j} + T_{i,(j-1)\to j}$.

V. THEOREMS

5.1. Theorem: The processing of jobs on M_3 at time $L_3 = \sum_{i=1}^{n} I_{i,3}$ keeps $t_{n,3}$ unaltered.

Proof: Let $t'_{i,3}$ be the competition time of i^{th} job on machine M_3 when M_3 starts processing of jobs at time L_3. We shall prove the theorem with the help of Mathematical Induction.

Let $P(n): t'_{n,3} = t_{n,3}$

Basic Step: For n = 1

$t'_{1,3} = L_3 + A_{1,3} = I_{1,3} + A_{1,3}$

$= (A_{1,1} + (T_{1,1\to 2} + A_{1,2}) + T_{1,2\to 3}) + A_{1,3} = t_{1,3}$.

Therefore P (1) is true.

Induction Step: Let P (k) be true.

i.e. $t'_{k,3} = t_{k,3}$.

Now, we shall show that P(k+1) is also true.

i.e. $t'_{k+1,3} = t_{k+1,3}$

But $t'_{k+1,3} = \max(t'_{k+1,2}, t'_{k,3} + S_{k,3}) + T_{k,2\to 3} + A_{k+1,3}$ (As per Definition 2)

$\therefore t'_{k+1,3} = \max(t_{k+1,2}, t_{k,3} + S_{k,3}) + T_{k,2\to 3} + A_{k+1,3}$

($\because t'_{k,3} = t_{k,3}$, By Assumption)

$= t_{k+1,3}$ (by Definition)

$\Rightarrow P(k+1)$ is true .

Hence by principle of mathematical induction P(n) is true for all n, .i.e. $t'_{n,3} = t_{n,3}$.

Remarks:

If M_3 starts processing jobs for minimum $L_3(S_r) = t_{n3}(S_r) - \sum_{i=1}^{n} A_{i,3} - \sum_{i=1}^{n-1} S_{i,3}$ then the total elapsed time $t_{n3}(S_r) = L_3(S_r) + \sum_{i=1}^{n} A_{i3} + \sum_{i=1}^{n-1} S_{i3}$ is not altered and M_3 is engaged for minimum time.

Lemma 5.1: If M_3 starts processing jobs at $L_3 = \sum_{i=1}^{n} I_{i,3}$ then

(i). $L_3 > t_{1,2}$

(ii). $t'_{k+1,3} \geq t_{k,2}$, $k > 1$.

5.2. Theorem: The processing of jobs on M_2 at time $L_2 = \min_{1\le k\le n}\{Y_k\}$ keeps total elapsed time unaltered where

$Y_1 = L_3 - A_{1,2} - T_{1,1\to 2}$ and

$$Y_k = t'_{k-1,3} - \sum_{i=1}^{k} A_{i,2} - \sum_{i=1}^{k} T_{i,2\to 3} - \sum_{i=1}^{k-1} S_{i,2};\ k>1.$$

Proof: We have

$$L_2 = \min_{i\le k\le n}\{Y_k\} = Y_r \text{ (say)}$$

In particular for k =1

$$Y_r \le Y_1$$

$$\Rightarrow Y_r + A_{1,2} + T_{1,1\to 2} \le Y_1 + A_{1,2} + T_{1,1\to 2}$$

$$\Rightarrow Y_r + A_{1,2} + T_{1,1\to 2} \le L_3 \qquad \text{---- (1)}$$

$$\left(\because Y_1 = L_3 - A_{1,2} - T_{1,1\to 2}\right)$$

By Lemma 1; we have

$$t_{1,2} \le L_3 \qquad \text{---- (2)}$$

Also, $t'_{1,2} = \max\left(Y_r + A_{1,2} + T_{1,1\to 2}, t_{1,2}\right)$

On combining, we get

$$t'_{1,2} \le L_3$$

For k >1, As $Y_r = \min_{i\le k\le n}\{Y_k\}$

$$\Rightarrow Y_r \le Y_k;\quad k = 2,3\ldots\ldots\ldots,n$$

$$\Rightarrow Y_r + \sum_{i=1}^{k} A_{i,2} + \sum_{i=1}^{k} T_{i,2\to 3} + \sum_{i=1}^{k-1} S_{i,3} \le Y_k + \sum_{i=1}^{k} A_{i,2} + \sum_{i=1}^{k} T_{i,2\to 3} + \sum_{i=1}^{k-1} S_{i,3}$$

$$\Rightarrow Y_r + \sum_{i=1}^{k} A_{i,2} + \sum_{i=1}^{k} T_{i,2\to 3} + \sum_{i=1}^{k-1} S_{i,3} \le t'_{k-1,3} \qquad \text{---- (3)}$$

By Lemma 1; we have

$$t_{k,2} \le t'_{k-1,3} \qquad \text{---- (4)}$$

Also,

$$t'_{k,2} = \max\left(Y_r + \sum_{i=1}^{k} A_{i,2} + \sum_{i=1}^{k} T_{i,2\to 3} + \sum_{i=1}^{k-1} S_{i,3}, t_{k,2}\right)$$

Using (3) and (4) , we get

$$t'_{k,2} \le t'_{k-1,3}$$

Taking $k = n$, we have

$$t'_{n,2} \le t'_{n-1,3} \qquad \text{---- (5)}$$

Total time elapsed = $t_{n,3}$

$$= \max\left(t'_{n,2}, t'_{n-1,3} + S_{n-1,3}\right) + A_{n,3} + T_{n,2\to 3}$$

$$= t'_{n-1,3} + S_{n-1,3} + A_{n,3} + T_{n,2\to 3} \qquad \text{(using 5)}$$

$$= t'_{n,3}.$$

Hence, the total time elapsed remains unaltered if M_2 starts processing jobs at time $L_2 = \min_{i\le k\le n}\{Y_k\}$.

5.3. Theorem: The processing time of jobs on M_2 at time $L_2 > \min_{1\le k\le n}\{Y_k\}$ increase the total time elapsed, where

$Y_1 = L_3 - A_{1,2} - T_{1,1\to 2}$ and

$$Y_k = t'_{k-1,3} - \sum_{i=1}^{k} A_{i,2} - \sum_{i=1}^{k} T_{i,2\to 3} - \sum_{i=1}^{k-1} S_{i,3};\ k>1.$$

The proof of the theorem can be obtained on the same lines as of the previous Theorem 2.

By Theorem 1, if M_3 starts processing jobs at time $L_3(S_r) = t_{n3}(S_r) - \sum_{i=1}^{n} A_{i,3} - \sum_{i=1}^{n-1} S_{i,3}$ then the total elapsed time $t_{n,3}$ is not altered and M_3 is engaged for minimum time equal to utilization time of M_3 Moreover total elapsed time/rental cost of M_1 is always least as utilization time of M_1 is always minimum. Therefore the the objective remains to minimize the elapsed time and hence the rental cost of M_2.

The following algorithm provides the procedure to determine the times at which machines should be taken on rent to minimize the total rental cost without altering the total elapsed time in three machine flow shop problem under rental policy (P).

VI. PROBLEM FORMULATION

Let some job i (i = 1,2,……..,n) are to be processed on three machines M_j (j = 1,2,3) under the specified rental policy P. 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 $T_{i,j\to k}$ be the transportation time of i^{th} job from j^{th} machine to k^{th} machine. Our aim is to find the sequence $\{S_k\}$ of the jobs which minimize the rental cost of all the three machines while minimizing total elapsed time.

The mathematical model of the problem can be stated as:

Minimize $U_j(S_k)$ and

Minimize

$$R(S_k) = t_{n1}(S_k)\times C_1 + U_2(S_k)\times C_2 + U_3(S_k)\times C_3$$

Subject to constraint: Rental Policy (P)

Our objective is to minimize rental cost of machines while minimizing total elapsed time.

VII. ALGORITHM

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

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

Step 2: Check the condition

Either $\text{Min}\{A_{i1} + T_{i,1\to 2} - S_{i2}\} \ge \text{Max}\{A_{i2} + T_{i,1\to 2} - S_{i1}\}$

or $\text{Min}\{A_{i3} + T_{i,2\to 3} - S_{i2}\} \ge \text{Max}\{A_{i2} + T_{i,2\to 3} - S_{i3}\}$

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

$$G_i = A_{i1} + A_{i2} + \max(S_{i1}, S_{i2}) + T_{i,1\to 2}$$

$$H_i = A_{i2} + A_{i3} - S_{i3} + T_{i,2\to 3}$$

Step 4: Find the expected processing time of job block β = (k,m) on fictitious machines G & H using equivalent job block criterion given by Maggu & Das [1977]. Find G_β and H_β using

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

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

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Step 5: Define new reduced problem with processing time G_i & H_i as defined in step 3 and replace job block (k,m) by a single equivalent job β with processing times G_β & H_β as defined in step 4.

Step 6: Using Johnson's procedure, obtain all sequences S_k having minimum elapsed time. Let these be $S_1, S_2, \ldots, S_r$

Step 7: Prepare In – Out tables for the sequences obtained in step 6 and read the effect of break-down interval (a ,b) on different jobs on the lines of *Singh T.P.* [1985].

Step 8: Form a reduced problem with processing times A'_{ij} (j=1,2,3)

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

$$A'_{ij} = A_{ij} + L \quad \forall i, j = 1,2,3$$

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

$$A'_{ij} = A_{ij} \quad \forall i, j = 1,2,3$$

Step 9: Now repeat the procedure to get optimal sequence S'_k

Step 10: Prepare In – Out tables for S'_k and compute total elapsed time $t_{n3}(S'_k)$

Step 11: Compute latest time L_3 for machine M_3 for sequence S'_k as

$$L_3(S'_k) = t_{n3}(S'_k) - \sum_{i=1}^{n} A'_{i3} - \sum_{i=1}^{n-1} S_{i,3}(S'_k)$$

Step 12: For the sequence S'_k ($k = 1,2,\ldots\ldots$,r), compute

I. $t_{n2}(S'_k)$

II. $Y_1(S'_k) = L_3(S'_k) - A'_{1,2}(S'_k) - T_{1,2\to3}$

III. $Y_q(S'_k) = L_3(S'_k) - \sum_{i=1}^{q} A'_{i2}(S'_k) - \sum_{i=1}^{q} T_{i,2\to3} - \sum_{i=1}^{q-1} S_{i,2}(S'_k) + \sum_{i=1}^{q-1} A'_{i,3} + \sum_{i=1}^{q-1} T_{i,1\to2} + \sum_{i=1}^{q-2} S_{i3}(S'_k); q = 2,3,\ldots\ldots,n$

IV. $L_2(S'_k) = \min_{1\le q\le n} \{Y_q(S'_k)\}$

V. $U_2(S'_k) = t_{n2}(S'_k) - L_2(S'_k)$.

Step 13: Find min $\{U_2(S'_k)\}; k = 1,2,\ldots\ldots\ldots\ldots,r$

Let it be for the sequence S'_p and then sequence S'_p will be the optimal sequence.

Step 14: Compute total rental cost of all the three machines for sequence S'_p as:

$$R(S'_p) = t_{n1}(S'_p) \times C_1 + U_2(S'_p) \times C_2 + U_3(S'_p) \times C_3$$

VIII. 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 and jobs 2 and 4 are processed as a group job (2, 4) with breakdown interval (12,14). The rental cost per unit time for machines M_1, M_2 and M_3 are 2 units, 10 units and 8 units respectively, under the specified rental policy P.

(Tableau 2)

	M_1				T	M_2				T	M_3			
i	a_{i1}	p_{i1}	s_{i1}	q_{i1}	$_{1\to2}$	a_{i2}	p_{i2}	s_{i2}	q_{i2}	$_{2\to3}$	a_{i3}	p_{i3}	s_{i3}	q_{i3}
1	27	.2	3	.3	2	7	.3	3	.2	2	19	.2	4	.2
2	30	.2	2	.1	1	20	.2	2	.2	1	18	.3	3	.2
3	41	.1	2	.3	2	20	.2	1	.2	2	14	.2	2	.3
4	23	.2	4	.1	2	23	.1	2	.2	3	23	.1	4	.2
5	20	.3	2	.2	4	10	.2	3	.2	1	25	.2	5	.1

Our objective is to obtain an optimal schedule for above said problem to minimize the total production time / total elapsed time subject to minimization of the total rental cost of the machines.

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 3

Jobs	A_{i1}	S_{i1}	$T_{i,1\to2}$	A_{i2}	S_{i2}	$T_{i,2\to3}$	A_{i3}	S_{i3}
1	5.4	0.9	2	2.1	0.6	2	3.8	0.8
2	6.0	0.2	1	4.0	0.4	1	5.4	0.6
3	4.1	0.6	2	4.0	0.2	2	2.8	0.6
4	4.6	0.4	2	2.3	0.4	3	2.3	0.8
5	6.0	0.4	4	2.0	0.6	1	5.0	0.5

(Tableau 3)

As per step 2 : Here Min $\{A_{i1} + T_{i,1\to2} - S_{i2}\} \geq$ Max$\{A_{i2} + T_{i,1\to2} - S_{i1}\}$

As per step 3: The expected processing time for two fictitious machine G & H is as shown in table 5.

Jobs	1	2	3	4	5
G_i	10.4	11.4	10.7	9.3	12.6
H_i	7.1	9.8	8.2	6.8	7.5

(Tableau 4)

As per Step 4: Here β= (2, 4)

G_β = 11.4 + 9.3 – 9.3 = 11.4, H_β = 9.8 + 6.8 – 9.3 = 7.3

As per Step 5: The reduced problem is

Jobs	1	Β	3	5
G_i	10.4	11.4	10.7	12.6
H_i	7.1	7.3	8.2	7.5

(Tableau 5)

As per Step 6: Using Johnson's method, the optimal sequence is S = 3 – 5 – β – 1 , .i.e. S = 3 – 5 – 2 – 4 – 1

As per Step 7: The In – Out table for the optimal sequence S

J	M_1	T12	M_2	T23	M_3
I	In – Out		In – Out		In - Out
3	0 – 4.1	2	6.1 – 10.1	2	12.1 – 14.9
5	4.7 – 10.7	4	14.7 – 16.7	1	17.7 – 22.7
2	11.1 – 17.1	1	18.1 – 22.1	1	23.2 – 28.6
4	17.3 – 21.9	2	23.9 – 26.2	3	29.2 – 31.5
1	22.3 – 27.7	2	29.7 – 31.8	2	33.8 – 37.6

(Tableau 6)

As per step 8: The new processing times after breakdown effect are as shown in table 7

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Jobs	A'_{i1}	S_{i1}	T12	A'_{i2}	S_{i2}	T23	A'_{i3}	S_{i3}
1	5.4	.9	2	2.1	.6	2	3.8	.8
2	8.0	.2	1	4.0	.4	1	5.4	.6
3	4.1	.6	2	4.0	.2	2	4.8	.6
4	4.6	.4	2	2.3	.4	3	2.3	.8
5	6.0	.4	4	2.0	.6	1	5.0	.5

(Tableau 7)

As per step 9 : Using Johnson's method optimal sequence is $S' : 3-5-2-4-1$

As per step 10: The In-Out table for the sequence S' is as shown in table 8.

J	M_1	T12	M_2	T23	M_3
i	In – Out		In – Out		In - Out
3	0 – 4.1	2	6.1 – 10.1	2	12.1 – 16.9
5	4.7 – 10.7	4	14.7 – 16.7	1	17.7 – 22.7
2	11.1 – 19.1	1	20.1 – 24.1	1	25.1– 30.5
4	19.3 – 23.9	2	25.9 – 28.2	3	31.2 – 33.5
1	24.3 – 29.7	2	31.7 – 33.8	2	35.8 – 39.6

(Tableau 8)

Total elapsed time $t_{n,3}(S')$ = 39.6 units

As per Step 11:

$$L_3(S') = t_{n3}(S') - \sum_{i=1}^{n} A'_{i,3} - \sum_{i=1}^{n-1} S_{i,3}(S')$$

= 39.6 – 21.3 – 2.5 = 15.8 units

As per Step 12: For sequence S', we have

$$t_{n2}(S') = 33.8$$
$$Y_1 = 15.8 - 4.0 - 2 = 9.8$$
$$Y_2 = 15.8 - 9.2 + 6.8 = 13.4$$
$$Y_3 = 15.8 - 14.8 + 16.4 = 17.4$$
$$Y_4 = 15.8 - 20.5 + 23.3 = 18.6$$
$$Y_5 = 15.8 - 25 + 28.2 = 19$$
$$L_2(S') = Min\{Y_k\} = 9.8$$
$$U_2(S') = t_{n2}(S') - L_2(S') = 33.8 - 9.8 = 24$$

The new reduced Bi-objective In – Out table is –

J	M_1	T12	M_2	T23	Machine M_3
i	In – Out		In – Out		In - Out
3	0 – 4.1	2	9.8 – 13.8	2	15.8 – 20.6
5	4.7 – 10.7	4	14.7 – 16.7	1	21.2 – 26.2
2	11.1 – 19.1	1	20.1 – 24.1	1	26.7 – 32.1
4	19.3 – 23.9	2	25.9 – 28.2	3	32.7 – 35.0
1	24.3 – 29.7	2	31.7 – 33.8	2	35.8 – 39.6

(Tableau 9)

The latest possible time at which machine M_2 should be taken on rent = $L_2(S')$ = 9.8 units.

Also, utilization time of machine M_2 = $U_2(S')$ = 24 units.

Total minimum rental cost =

$$R(S') = t_{n1}(S') \times C_1 + U_2(S') \times C_2 + U_3(S') \times C_3$$

=29.7 × 2 + 24 × 10 + 23.8 × 8 = 489.8 units

IX. CONCLUSION

If machine M_3 starts processing the jobs at the latest time $L_3 = t_{n,3} - \sum_{i=1}^{n} A_{i,3}$ then the total elapsed time $t_{n,3}$ is not altered and M_3 is engaged for minimum time equal to sum of processing times of all the jobs on M_3, i.e. reducing the idle time of M_3 to zero. If the machine M_2 is taken on rent when it is required and is returned as soon as it completes the last job, the starting of processing of jobs at the latest time $L_2(S_k) = \min_{1 \le q \le n}\{Y_q(S_k)\}$ on M_2 will, reduce the idle time of all jobs on it. Therefore, the utilization time and hence total rental cost of machine M_2 will be minimum. Also rental cost of M_1 will always be minimum as the idle times of machine M_1 is always zero.

X. APPENDIX

Computer Programme

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<process.h>
int n,j;
float
a1[16],b1[16],c1[16],a11[16],b11[16],c11[16],g[16],h[16],T
12[16],T23[16],s11[16],s22[16],s33[16],g21[16],h21[16],g1
2[16],h12[16];
float macha[16],machb[16],machc[16],macha1[16];
float machb1[16],machc1[16],g11[16],h11[16],g22[16];
float h22[16],g23[16],h23[16];
int group[2];//variables to store two job blocks
int cost_a,cost_b,cost_c,w[16];
float minval,minv,minv1,maxv1[16],maxv2[16],cost;
int bd1,bd2;// Breakdown interval
float gbeta=0.0,hbeta=0.0,gbeta1=0.0,hbeta1=0.0;
void main()
{
        clrscr();
        int a[16],b[16],c[16],j[16],s1[16],s2[16],s3[16];
        float p[16],q[16],r[16],x[16],t1[16],u[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, set up time and the
probabilities of "<<i<<" job for machine A and
Transportation time from Machine A to B : ";
```

```
cin>>a[i]>>p[i]>>s1[i]>>x[i]>>T12[i];
cout<<"\nEnter the processing time, setup time and the
probabilities of "<<i<<" job for machine B and
Transportation time from Machine B to C : ";
            cin>>b[i]>>q[i]>>s2[i]>>t1[i]>>T23[i];
cout<<"\nEnter the processing time and its probability of
"<<i<<"job for machine C: ";
cin>>c[i]>>r[i]>>s3[i]>>u[i];
cout<<"\nEnter the weightage  of "<<i<<" job :"; cin>>w[i];
//Calculate the expected processing & setup 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];
s11[i]=s1[i]*x[i]; s22[i]= s2[i]*t1[i]; s33[i]= s3[i]*u[i];
              }
cout<<endl<<"Expected processing time of machine A, B
and C : \n";
            for(i=1;i<=n;i++)
            {
            cout<<j[i]<<"\t"<<a1[i]<<"\t"<<s11[i]<<"\t"<<T1
2[i]<<"\t"<<b1[i]<<"\t"<<s22[i]<<"\t"<<T23[i]<<"\t"<<c1[i
]<<"\t"<<s33[i]<<"\t"<<w[i];cout<<endl;
            }
cout<<"\nEnter the rental cost of Machine
M1:";cin>>cost_a;
cout<<"\nEnter the rental cost of Machine
M2:";cin>>cost_b;
cout<<"\nEnter the rental cost of Machine
M3:";cin>>cost_c;
cout<<"\nEnter the two breakdown interval:";
cin>>bd1>>bd2;
//Function for two ficticious machine G and H
//Finding Smallest in a1
float mina1;
mina1=a1[1]+T12[1]-s22[1];
for(i=2;i<n;i++)
            {
             if(a1[i]+T12[i]-s22[i]<mina1)
             mina1=a1[i]+T12[i]-s22[i];
            }
//For finding Largest in b1
float maxb1;
maxb1=b1[1]+T12[1]-s11[1];
            for(i=2;i<n;i++)
            {
if(b1[i]+T12[i]-s11[i]>maxb1)
maxb1=b1[i]+T12[i]-s11[i];
            }
float maxb2;
maxb2=b1[1]+T23[1]-s33[i];
for(i=2;i<n;i++)
            {
if(b1[i]+T23[i]-s33[i]>maxb2)
maxb2=b1[i]+T23[i]-s33[i];
            }
//Finding Smallest in c1
float minc1;
minc1=c1[1]+T23[1]-s22[i];
for(i=2;i<n;i++)
            {
if(c1[i]+T23[i]-s22[i]<minc1)
minc1=c1[i]+T23[i]-s22[i];
            }
float maxs;
if(mina1>=maxb1||minc1>=maxb2)
            {
for(i=1;i<=n;i++)
            {
            if(s11[i]>=s22[i])
            {maxs=s11[i];}
            else
            {maxs=s22[i];}
            g11[i]=a1[i]+b1[i]+T12[i]+maxs;
            h11[i]=c1[i]+b1[i]+T23[i]-s33[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"<<g11[i]<<"\t"<<h11[i]<<"\t"<<w[i];
  cout<<endl;}
for(i=1;i<=n;i++)
            {
            if(g11[i]<h11[i])
            {g12[i]=g11[i]+w[i];h12[i]=h11[i];}
else
            {h12[i]=h11[i]+w[i];g12[i]=g11[i];}}
for(i=1;i<=n;i++)
            {g[i]=g12[i]/w[i];h[i]=h12[i]/w[i];}
for(i=1;i<=n;i++)
{cout<<"\n\n"<<j[i]<<"\t"<<g[i]<<"\t"<<h[i]<<"\t";
cout<<endl; cout<<endl;  }
cout<<"\nEnter the two job blocks(two numbers from 1 to
"<<n<<"):";
cin>>group[0]>>group[1];
//calculate G_Beta and H_Beta
if(g[group[1]]<h[group[0]])
{minv=g[group[1]];}
else
{minv=h[group[0]];}
gbeta=g[group[0]]+g[group[1]]-minv;
hbeta=h[group[0]]+h[group[1]]-minv;
cout<<endl<<endl<<"G_Beta="<<gbeta;
cout<<endl<<"H_Beta="<<hbeta;
int f=1;int j1[16];float g1[16],h1[16];
for(i=1;i<=n;i++)
{if(j[i]==group[0]||j[i]==group[1])
{f--;}
else
{j1[f]=j[i];}
f++;}
j1[n-1]=17;
for(i=1;i<=n-2;i++)
{g1[i]=g[j1[i]];h1[i]=h[j1[i]];}
g1[n-1]=gbeta;h1[n-1]=hbeta;
cout<<endl<<endl<<"displaying original scheduling
table"<<endl;
for(i=1;i<=n-1;i++)
{cout<<j1[i]<<"\t"<<g1[i]<<"\t"<<h1[i]<<endl;}
float mingh[16];char ch[16];
for(i=1;i<=n-1;i++)
{
```

```
if(g1[i]<h1[i])
 {mingh[i]=g1[i];ch[i]='g'; }
else
{mingh[i]=h1[i];ch[i]='h';}}
for(i=1;i<=n-1;i++)
{
for(int j=1;j<=n-1;j++)
        if(mingh[i]<mingh[j])
        {float temp=mingh[i]; int temp1=j1[i]; char d=ch[i];
            mingh[i]=mingh[j]; j1[i]=j1[j]; ch[i]=ch[j];
            mingh[j]=temp; j1[j]=temp1; ch[j]=d;    }}
// calculate beta scheduling
float sbeta[16];int t=1,s=0;
for(i=1;i<=n-1;i++)
        {
        if(ch[i]=='h')
        {
         sbeta[(n-s-1)]=j1[i]; s++;
        }
else    if(ch[i]=='g')
        {
        sbeta[t]=j1[i];t++;
        }}
int arr1[16], m=1;
cout<<endl<<endl<<"Job Scheduling:"<<"\t";
for(i=1;i<=n-1;i++)
        {
if(sbeta[i]==17)
{arr1[m]=group[0];arr1[m+1]=group[1];
cout<<group[0]<<" "<<group[1]<<" ";m=m+2;
continue;}
else
        {
        cout<<sbeta[i]<<" ";arr1[m]=sbeta[i];m++;
        }}
//calculating total computation sequence
float time=0.0,macha11[16];
macha[1]=time+a1[arr1[1]];
for(i=2;i<=n;i++)
        {
        macha11[i]=macha[i-1]+s11[arr1[i-1]];
        macha[i]=macha11[i]+a1[arr1[i]];
        }
machb[1]=macha[1]+b1[arr1[1]]+T12[arr1[1]];
for(i=2;i<=n;i++)
        {
if((machb[i-1]+s22[arr1[i-1]])>(macha[i]+T12[arr1[i]]))
        maxv1[i]=machb[i-1]+s22[arr1[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++)
        {
if((machc[i-1]+s33[arr1[i-1]])>(machb[i]+T23[arr1[i]]))
        {
        maxv2[i]=machc[i-1]+s33[arr1[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]<<en
dl;
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"<<macha11[i]<<"--"<<macha[i]<<"
"<<"\t"<<maxv1[i]<<"--"<<machb[i]<<"
"<<"\t"<<maxv2[i]<<"--"<<machc[i]<<"\t"<<w[arr1[i]]<<
endl;
if(macha11[i]<=bd1 && macha[i]<=bd1 || macha11[i]>=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);
            }}
int j2[16];
for(i=1;i<=n;i++)
            {
  j2[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"<<endl;
for(i=1;i<=n;i++)
            {
cout<<endl;
cout<<j2[i]<<"\t"<<"\t"<<a11[i]<<"\t"<<"\t"<<b11[i]<<"\t"
<<"\t"<<c11[i];
cout<<endl;
            }
float mina12,maxb12,maxb22,minc12;
//Finding Smallest in a11
            mina12=a11[1]+T12[1]-s22[1];
            for(i=2;i<n;i++)
            {
if(a11[i]+T12[i]-s22[i-1]<mina12)
              mina12=a11[i]+T12[i]-s22[i-1];}
//For finding Largest in b11
            maxb12=b11[1]+T23[1]-s33[1];
for(i=2;i<n;i++)
            {
            if(b11[i]+T23[i]-s33[i]<maxb12)
              maxb12=b11[i]+T23[i]-s33[i];
            }
            maxb22=b11[1]+T12[1]-s11[i];
for(i=2;i<n;i++)
            {
            if(b11[i]+T12[i]-s11[i]>maxb22)
              maxb22=b11[i]+T12[i]-s11[i];
            }
//Finding Smallest in c12
            minc12=c11[1]+T23[1]-s22[1];
            for(i=2;i<n;i++)
            {
            if(c11[i]+T23[i]-s22[i]<minc12)
              minc12=c11[i]+T23[i]-s22[i];
            }
if(mina12>=maxb22||minc12>=maxb12)
            {
            for(i=1;i<=n;i++)
            {
            g22[i]=a11[i]+b11[i]+T12[i]+maxs;
             h22[i]=c11[i]+b11[i]+T23[i]-s33[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"<<g22[i]<<"\t"<<h22[i]<<"\t"<<w[i];
  cout<<endl;
            }
for(i=1;i<=n;i++)
            {
if(g22[i]<h22[i])
            {
            g23[i]=g22[i]+w[i];h23[i]=h22[i];
            }
else
            {
            h23[i]=h22[i]+w[i];g23[i]=g22[i];
            }}
for(i=1;i<=n;i++)
            {
            g21[i]=g23[i]/w[i];   h21[i]=h23[i]/w[i];
            }
for(i=1;i<=n;i++)
            {
            cout<<"\n\n"<<j[i]<<"\t"<<g21[i]<<"\t"<<h21[i];
            cout<<endl; cout<<endl;
            }
//calculate G_Beta and H_Beta
if(g21[group[1]]<h21[group[0]])
            {
            minv1=g21[group[1]];
            }
else
            {
            minv1=h21[group[0]];
            }
gbeta1=g21[group[0]]+g21[group[1]]-minv1;
hbeta1=h21[group[0]]+h21[group[1]]-minv1;
cout<<endl<<endl<<"G_Beta1="<<gbeta1;
cout<<endl<<"H_Beta1="<<hbeta1;
int f1=1;int j3[16];float g11[16],h11[16];j[i]=i;
for(i=1;i<=n;i++)
            {
if(j[i]==group[0]||j[i]==group[1])
            {
            f1--;
            }
else
            {
            j3[f1]=j[i];}f1++;
            }
j3[n-1]=17;
```

```
for(i=1;i<=n-2;i++)
	{
g11[i]=g21[j3[i]];h11[i]=h21[j3[i]];
	}
g11[n-1]=gbeta1;h11[n-1]=hbeta1;
cout<<endl<<endl<<"displaying original scheduling
table"<<endl;
for(i=1;i<=n-1;i++)
	{
cout<<j3[i]<<"\t"<<g11[i]<<"\t"<<h11[i]<<endl;
	}
float mingh1[16];char ch1[16];
for(i=1;i<=n-1;i++)
	{if(g11[i]<h11[i])
	  {
	mingh1[i]=g11[i];ch1[i]='g';
	}
else
	 {
	mingh1[i]=h11[i];ch1[i]='h';
	}}
	for(i=1;i<=n-1;i++)
	{
	 for( int j=1;j<=n-1;j++)
	       if(mingh1[i]<mingh1[j])
	           {
float temp11=mingh1[i]; int temp12=j3[i]; char d1=ch1[i];
mingh1[i]=mingh1[j]; j3[i]=j3[j]; ch1[i]=ch1[j];
mingh1[j]=temp11; j3[j]=temp12; ch1[j]=d1;
	           }}
// calculate beta scheduling
float sbeta1[16];int d=1,q1=0;
for(i=1;i<=n-1;i++)
	{
	if(ch1[i]=='h')
	{
	sbeta1[(n-q1-1)]=j3[i]; q1++;}
else
	if(ch1[i]=='g')
	{
	sbeta1[d]=j3[i];d++;
	}}
int arr2[16], m1=1;
cout<<endl<<endl<<"Job Scheduling:"<<"\t";
for(i=1;i<=n-1;i++)
{
if(sbeta1[i]==17)
{
arr2[m1]=group[0];arr2[m1+1]=group[1];
cout<<group[0]<<" "<<group[1]<<" ";m1=m1+2;
continue;}
else
{
cout<<sbeta1[i]<<" ";arr2[m1]=sbeta1[i];
m1++;
}
}
//calculating total computation sequence;
float time1=0.0,macha12[16] ;
float maxv11[16],maxv21[16];
macha1[1]=time1+a11[arr2[1]];
for(i=2;i<=n;i++)
	{
	macha12[i]=macha1[i-1]+s11[arr2[i-1]];
	macha1[i]=macha12[i]+a11[arr2[i]];}
	machb1[1]=macha1[1]+b11[arr2[1]]+T12[arr2[1]];
	}
for(i=2;i<=n;i++)
{
if((machb1[i-1]+s22[arr2[i-1]])>(macha1[i]+T12[arr2[i]]))
maxv11[i]=machb1[i-1]+s22[arr2[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]+s33[arr2[i-1]])>(machb1[i]+T23[arr2[i]]))
{
maxv21[i]=machc1[i-1]+s33[arr2[i-1]];
}
else
{
maxv21[i]=machb1[i]+T23[arr2[i]];
machc1[i]=maxv21[i]+c1[arr2[i]];
 }}
///displaying solution
cout<<"\n\n\n\n\n\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;
for(i=2;i<=n;i++)
{cout<<arr2[i]<<"\t"<<macha12[i]<<"--"<<macha1[i]<<""
<<"\t"<<maxv11[i]<<"--"<<machb1[i]<<"
"<<"\t"<<maxv21[i]<<"--"<<machc1[i]<<endl;}
cout<<"\n\n\nTotal Elapsed Time (T) = "<<machc1[n];
float L3,Y[16],min,u2,u3;
float sum1=0.0,sum2=0.0,sum3=0.0;
for(i=1;i<=n;i++)
{sum1=sum1+a11[i];sum2=sum2+b11[i];
sum3=sum3+c11[i]; }
float sum_1;
for(i=1;i<=n;i++)
{sum_1=0.0,s33[0]=0.0;
for(int e=0;e<=n-1;e++)
{sum_1=sum_1+s33[arr2[e]];}}
L3=machc1[n]-sum3-sum_1;
cout<<"\n\nLatest Time When Machine M3 is Taken on
Rent:"<<L3;
cout<<"\n\nTotal Completion Time of Jobs on Machine
M2:"<<machb1[n];
Y[1]=L3-b11[arr2[1]]-T23[arr2[1]];
cout<<"\n\n\tY[1]\t="<<Y[1];
```

```
float sum_2,sum_3;
for(i=2;i<=n;i++)
{
sum_2=0.0,sum_3=0.0;
for(int j=1;j<=i-1;j++)
{
sum_3=sum_3+c11[arr2[j]]+T12[arr2[j]]+s33[arr2[j-1]];
}
for(int k=1;k<=i;k++)
{
sum_2=sum_2+b11[arr2[k]]+T23[arr2[k]]+s22[arr2[k-1]];
}
Y[i]=L3+sum_3-sum_2;
cout<<"\n\n\tY["<<i<<"]\t="<<Y[i];
}
min=Y[1];
for(i=2;i<n;i++)
{
if(Y[i]<min)
min=Y[i];
}
cout<<"\n\nMinimum of Y[i]="<<min;
u2=machb1[n]-min;u3=machc1[n]-L3;
cout<<"\n\n Utiliztaion time of machine M1 =
"<<macha1[n];
cout<<"\n\nUtilization Time of Machine M2="<<u2;
cout<<"\n\n Utiliztion time of machine M3 = "<<u3;
cost=(macha1[n]*cost_a)+(u2*cost_b)+(u3*cost_c);
cout<<"\n\nThe Minimum Possible Rental Cost is="<<cost;
cout<<"\n\n\t************************************
*************************";
getch();
}
```

REFERENCES

[1] S. M. Johnson, "Optimal two and three stage production schedule with set up times included", *Naval Research Logistics Quart*, vol. 1, pp. 61-68, 1954.

[2] W. E. Smith, "Various optimizers for single stage production", *Naval Research Logistics*, vol. 3, pp. 59-66, 1956.

[3] P. L. Maggu and G. Das, "Equivalent jobs for job block in job scheduling" *Opsearch*, vol. 14, pp. 277-281, 1977.

[4] Yoshida and Hitomi, "Optimal two stage production scheduling with set up times separated", *AIIE Transactions*, vol. II, pp. 261-263, 1979.

[5] I. Adiri, J. Bruno, E. Frostig and R. A. H. G. Kan, "Single machine flow time scheduling with a single break-down", *Acta Information*, vol. 26, pp. 679-696, 1979.

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

[7] M. S. Akturk and E. Gorgulu, "Match up scheduling under a machine break- down", *European journal of operational research,* pp. 81-99, 1999.

[8] Brucker and S. Knust , "Complexity results for flow-shop and open-shop scheduling problems with transportation delays" , *Annals of Operational Research*, vol. 129, pp. 81-106, 2004.

[9] A. B. Chandramouli, "Heuristic Approach for n-job,3-machine flow shop scheduling problem involving transportation time, breakdown interval and weights of jobs", *Mathematical and Computational Applications*, vol. 10, pp. 301-305, 2005.

[10] N. Chikhi, "Two machine flow-shop with transportation time". *Thesis of Magister*, Faculty of Mathematics, USTHB University ,Algiers, 2008..

[11] Belwal and Mittal, "n jobs machine flow shop scheduling problem with break down of machines, transportation time and equivalent job block", *Bulletin of Pure & Applied Sciences-Mathematics*, source 27, Source Issue 1.

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

[13] Pandian and Rajendran, "Solving constrained flow-shop scheduling problems with three machines", *Int. J. Contemp. Math. Sciences*, vol. **5**, pp. 921-929, 2010.

[14] D. Gupta, S. Sharma, Seema and Shefali, "Bicriteria in n × 2 flow shop scheduling under specified rental policy ,processing time and setup time each associated with probabilities including job-block". *Industrial Engineering Letters*, vol.1, pp. 1-12, 2011.

[15] D. Gupta, S. Sharma and Seema, "Bicriteria in n x 3 flow shop scheduling under specified rental policy, processing time associated with probabilities including transportation time and job block criteria" *Mathematical Theory and Modeling*, vol. 1, pp. 7-18, 2011.

Deepak Gupta received the Ph.D degree in the field of scheduling from C.C.S. University, Merrut in 2007. He has a credit of 13 books and 71 research papers in various journals of international repute. He is a member of IACSIT, Singapore.

Sameer Sharma received the M.Sc. degree in Mathematics from D. A. V. College, Jalandhar, Punjab in 1999. He has also done M.Phil., P.G.D.St. From Panjab University, Chandigarh and has a credit of 4 books and 35 research papers in various journals of international repute. He is a member of IACSIT, Singapore. He is currently working towards the Ph.D. degree from M.M. University, Mullana, Ambala, India.

Seema Sharma received the M.Sc. degree in Mathematics from D. A. V. College, Jalandhar, Punjab in 1999. She has also done M.Phil. and has a credit of 3 books and 15 research papers in various journals of international repute. She is currently working towards the Ph.D. degree from M.M. University, Mullana, Ambala, India.

Shefali Aggarwal received the M.Sc. degree in Mathematics from D. A. V. College, Chandigarh in 2009. She has a credit of 1 book and 10 research papers in various journals of international repute. She is currently working towards the Ph.D. degree from M.M. University, Mullana, Ambala, India.