International Journal of Scientific and Innovative Mathematical Research (IJSIMR)
Volume 3, Issue 5, May 2015, PP 31-40
ISSN 2347-307X (Print) & ISSN 2347-3142 (Online)
www.arcjournals.org

Bicriteria in Flowshop Problems under Rental Situation

Laxmi Narain

Department of Mathematics
Acharya Narendra Dev College, University of Delhi
Delhi, India
laxminarain@andc.du.ac.in

Abstract: *This paper studies bi-criteria in three-machine flow-shop problems under rental policy P_3. In this paper, Policy P_3 is modified. Here second and third machines will not be taken on rent at times when the first job is completed on first and second machines respectively but these machines will be taken on rent subject to criteria of minimum total elapsed. The objective is: obtain the sequence which minimizes total elapsed time and provides the total rental cost as minimum as possible. We have proved three theorems to obtain a simple and efficient algorithm. The algorithm is illustrated through a numerical example.*

Keywords: *Flow-shop Scheduling, Idle Time, Elapsed Time, Rental Time, Rental Cost.*

1. Introduction

A survey of scheduling literature has revealed the desirability of an optimal schedule being evaluated by more than one performance measures or criteria. Various authors viz., Sen and Dileepan [1], Sen and Gupta [2], Sen and Raiszadeh [3], Smith [4], Van Wassenhove and Gelders [5], Van Wassenhove and Baker [6] have studied the flow-shop problems having more than one optimization measures. Gupta and Dudek [7] strongly recommended the use of combination of criteria total flow-time and total elapsed time. Dileepan and Sen [8] surveyed the bicriteria scheduling research for a single machine. Chandersekhran [9] gave a technique based on Branch-and-Bound method and satisfaction of certain conditions to obtain a sequence which minimizes total flow-time subject to certain conditions which are to be satisfied. Bagga and Ambika [10] provided the procedure for obtaining sequence(s) in n-job, m-machine special flow-shop problems which gives minimum possible makespan while minimizing total flow-time. Narain and Bagga [11] studied n-job, m-machine special flow-shop problems which give minimum possible mean flowtime while minimizing total elapsed time. Narain and Bagga [12] determine the sequence which minimizes the total elapsed time subject to zero total idle time of machines i.e., machines should not remain idle once they start the first job. Narain and Bagga [13] studies n-job, m-machines flowshop problems when processing times of jobs on various machines follow certain conditions and the objective is to obtain a sequence which minimizes total elapsed time under no-idle constant. Narain and Bagga [14] studied n-job, 2-machine flowshop problem and provided an algorithm for obtaining a sequence which gives minimum possible mean flowtime under no-idle constraint.

In flow-shop problem, situation can occur in practice when one has got the assignment but does not have one's own machines or does not have enough money for the purchase of machines, under these circumstances, may take machines on rent to complete the assignment. Minimization of total rental cost of machines will be the criterion in these types of situations.

The following renting policies generally exist:

P_1 : *All the machines are taken on rent at one time and are returned also at one time.*

P_2 : *All the machines are taken on rent at one time and are returned as and when they are no longer required.*

P_3 : *All the machines are taken on rent as and when they are required and are returned as and when they are no longer required for processing.*

Bagga [15] studied three-machine problem under policy P_1 and provide a sequence to minimize the total rental cost of machines. Under P_2; for three-machine flow-shop problem, Bagga and Ambika

[16] provided a Branch-and-Bound algorithm. Bagga and Khurana [17] use Branch-and-Bound technique to solve n-job, 2-machine flow-shop problems under two policies: (i) when both machines are hired simultaneously and (ii) when the 2nd machine is hired only when the first job is completed on 1st machine. Narain [18] studied n x m flowshop problem under policy P_3 and provided an algorithm to find the minimum times at which machines should be taken on rent so that total rental cost is minimum.

This paper studies bi-criteria in three-machine flow-shop problems under rental policy P_3. In this paper, Policy P_3 is modified. Here second and third machines will not be taken on rent at times when the first job is completed on first and second machines respectively but these machines will be taken on rent subject to criteria of minimum total elapsed. The objective is: Obtain the sequence which minimizes total elapsed time and provides the total rental cost as minimum as possible. For any sequence S,

Total rental cost of machines = $\sum_{j=1}^{3}\sum_{i=1}^{n}[p_{i,j}(S)+I_{i,j}(S)]\times C_j$

Where $p_{i,j}(S)$ is the processing time of i^th^ job of sequence S on machine M_j, $I_{i,j}(S)$ is the idle time of machine M_j for i^th^ job of sequence S and C_j is rental cost per unit time of machine M_j. Here, the processing times $p_{i\,j}(S)$ and rental cost $C_j(S)$ are constant. Therefore, we can only reduce idle times $I_{i,j}(S)$. To reduce idle times on machines, we delay the times of renting of machines to process jobs. We have obtained a simple and efficient algorithm to provide the sequence which minimizes total elapsed time and provides the total rental cost as minimum as possible. Numerical example is given to illustrate the algorithm.

2. NOTATIONS

S	:	Sequence of jobs 1, 2, …, n.
M_j	:	Machine j; j=1, 2, 3.
$p_{i,j}(S)$	:	Processing time of i^th^ job of sequence S on machine M_j.
$I_{i,j}(S)$	:	Idle time of machine M_j for i^th^ job of sequence S.
C_j	:	Rental cost per unit time of machine M_j.
$H_j(S)$	:	The time when M_j is taken on rent for sequence S.
$Z_{i,j}(S)$	:	Completion time of i^th^ job of sequence S on machine M_j.
$Z'_{i,j}(S)$	:	Completion time of i^th^ job of sequence S on machine M_j when M_j starts processing jobs at time $H_j(S)$.
$T_2(S)$	:	Total time for which M_2 is required when M_2 starts processing jobs at time $H_2(S)$.
		i = 1, 2, …, n and j= 1, 2, 3.

3. MATHEMATICAL FORMULATION

Let n jobs require processing over three machines M_1, M_2 and M_3 in the order $M_1 \rightarrow M_2 \rightarrow M_3$.

Theorem 3.1: *If we start processing jobs on M_3 at time $H_3 = \sum_{i=1}^{k} I_{i,3}$, then $Z_{k,3}$ will remain unaltered.*

Proof: Let $Z'_{i,3}$ be the completion time of i^th^ job on machine M_3 when M_3 starts processing jobs at time H_3. The proof of the theorem is based on the method of *mathematical induction.*

For k = 1;

$Z'_{1,3} = H_3 + p_{1,3}$

$= \sum_{i=1}^{1} I_{i,3} + p_{1,3}$

$= p_{1,1} + p_{1,2} + p_{1,3}$

$= Z_{1,3}$

Therefore, the result holds for k = 1.

Let the result holds for k = m

For k = m+1;

$$Z'_{m+1,3} = \max(Z_{m+1,2}, Z'_{m,3}) + p_{m+1,3}$$

$$= \max\left(Z_{m+1,2}, H_3 + \sum_{i=1}^{m} p_{i,3}\right) + p_{m+1,3}$$

$$= \max\left(Z_{m+1,2}, \sum_{i=1}^{m+1} I_{i,3} + \sum_{i=1}^{m} p_{i,3}\right) + p_{m+1,3}$$

$$= \max\left(Z_{m+1,2}, \sum_{i=1}^{m} I_{i,3} + \sum_{i=1}^{m} p_{i,3} + I_{m+1,3}\right) + p_{m+1,3}$$

$$= \max(Z_{m+1,2}, Z_{m,3} + \max(Z_{m+1,2} - Z_{m,3}, 0)) + p_{m+1,3}$$

$$= \max(Z_{m+1,2}, \max(Z_{m+1,2}, Z_{m,3})) + p_{m+1,3}$$

$$= \max(Z_{m+1,2}, Z_{m,3}) + p_{m+1,3}$$

$$= Z_{m+1,3}$$

Therefore, the result holds for k = m+1 also.

Hence, by mathematical induction this theorem holds for all k, where k = 1, 2, ..., n.

If M_3 starts processing jobs at time H_3, where $H_3 = Z_{n,3} - \sum_{i=1}^{n} p_{i,3}$, then total elapsed time $Z_{n,3}$ is not altered and M_3 is engaged for minimum time equal to sum of the processing times of all the jobs on M_3. Moreover, it can be easily shown that if M_3 starts processing jobs at time H_3, then

$$Z'_{k,3} = H_3 + \sum_{i=1}^{k} p_{i,3}$$

Lemma 3.1: *If M_3 starts processing jobs at time $H_3 = \sum_{i=1}^{n} I_{i,3}$,*

then $H_3 \geq Z_{1,2}$ and $Z'_{k,3} \geq Z_{k,2}$ for k > 1.

Proof: $H_3 = \sum_{i=1}^{n} I_{i,3}$

$$= I_{1,3} + \sum_{i=2}^{n} I_{i,3}$$

$$= Z_{1,2} + \sum_{i=2}^{n} I_{i,3}$$

Since, $\sum_{i=2}^{n} I_{i,3} \geq 0$, therefore, $H_3 \geq Z_{1,2}$

Now, $I_{k,3} = \max(Z_{k,2} - Z_{k-1,3}, 0)$

Therefore, $I_{k,3} \geq Z_{k,2} - Z_{k-1,3}$

i.e., $Z_{k-1,3} + I_{k,3} \geq Z_{k,2}$

i.e., $\sum_{i=1}^{k-1} I_{i,3} + \sum_{i=1}^{k-1} p_{i,3} + I_{k,3} \geq Z_{k,2}$

i.e., $\sum_{i=1}^{k} I_{i,3} + \sum_{i=1}^{k-1} p_{i,3} \geq Z_{k,2}$

Since, $\sum_{i=k+1}^{n} I_{i,3} \geq 0$,

Therefore, $\sum_{i=1}^{k} I_{i,3} + \sum_{i=k+1}^{n} I_{i,3} + \sum_{i=1}^{k-1} p_{i,3} \geq Z_{k,2}$

i.e., $\sum_{i=1}^{n} I_{i,3} + \sum_{i=1}^{k-1} p_{i,3} \geq Z_{k,2}$

i.e., $H_3 + \sum_{i=1}^{k-1} p_{i,3} \geq Z_{k,2}$

i.e., $Z'_{k-1,3} \geq Z_{k,}$

Hence, this lemma is proved.

Theorem 3.2: *Total elapsed time will not be altered, if M_2 starts processing jobs at time $H_2 = min \{Y_k\}$, where*

$$Y_1 = H_3 - p_{1,2}$$

and

$$Y_k = Z'_{k-1,3} - \sum_{i=1}^{k} p_{i,2}\ ; \quad k=2, 3, \ldots, n$$

Proof: $H_2 = Y_r = \min \{Y_k\}$; k=1, 2, …, n

For k = 1;

$Y_r = \min \{Y_k\}$; k=1, 2, …, n

Therefore, $Y_r \leq Y_1$

i.e., $Y_r + p_{1,2} \leq Y_1 + p_{1,2}$

i.e., $Y_r + p_{1,2} \leq H_3$ (1)

From Lemma 3.1;

$Z_{1,2} \leq H_3$ (2)

Now,

$Z'_{1,2} = \max (Y_r + p_{1,2}, Z_{1,2})$

From equations (1) and (2);

$Z'_{1,2} \leq H_3$ (3)

For k > 1;

$Y_r = \min \{Y_k\}$;   k=2, 3, …, n

Therefore, $Y_r \leq Y_k$;   k=2, 3, …, n

i.e., $Y_r + \sum_{i=1}^{k} p_{i,2} \leq Y_k + \sum_{i=1}^{k} p_{i,2}$

i.e., $Y_r + \sum_{i=1}^{k} p_{i,2} \leq Z'_{k-1,3}$ (4)

From Lemma 3.1;

$Z_{k,2} \leq Z'_{k-1,3}$ (5)

Now,

$Z'_{k,2} = \max (Y_r + \sum_{i=1}^{k} p_{i,2} , Z_{k,2})$

From equation (4) and (5);

$Z'_{k,2} \leq Z'_{k-1,3}$;   k=2, 3, …, n (6)

Taking k = n in equation (6);

$Z'_{n,2} \leq Z'_{n-1,3}$ (7)

Total elapsed time $= \max (Z'_{n,2} , Z'_{n-1,3}) + p_{n,3}$

$= Z'_{n-1,3} + p_{n,3}$

$= Z'_{n,3}$

$= Z_{n,3}$

Hence, total elapsed time will not be altered if M_2 starts processing jobs at time $H_2 = \min \{Y_k\}$;   k=1, 2, …, n. □

Theorem 3.3: *Total elapsed time will increase, if M_2 starts processing jobs at time $H_2 > \min \{Y_k\}$, where*

$Y_1 = H_3 - p_{1,2}$

and

$Y_k = Z'_{k-1,3} - \sum_{i=1}^{k} p_{i,2}$;   *k=2, 3, …, n*

Proof: There arise two cases:

Case 1: $H_2 > Y_1$, then $H_2 + p_{1,2} > Y_1 + p_{1,2} = H_3$

i.e., $H_2 + p_{1,2} > H_3$

Therefore, total elapsed time $\geq H_2 + p_{1,2} + \sum_{i=1}^{n} p_{i,3}$

$\geq H_3 + \sum_{i=1}^{n} p_{i,3} = Z'_{n,3} = Z_{n,3}$

Hence, total elapsed time will increase if M_2 starts processing jobs at time $H_2 > Y_1$.

Case 2: Let $Y_r = \min \{Y_k\}$;   k=1, 2, …, n

Let $H_2 = Y_k$, then

$$Y_k + \sum_{i=1}^{r} p_{i,2} > Y_r + \sum_{i=1}^{r} p_{i,2} \tag{8}$$

Now,

$$Z'_{r,2} = \max\left(Y_k + \sum_{i=1}^{r} p_{i,2}, Z_{r,2}\right)$$

Therefore, $Z'_{r,2} \geq Y_k + \sum_{i=1}^{r} p_{i,2}$

From equation (8);

$$Z'_{r,2} > Y_r + \sum_{i=1}^{r} p_{i,2} = Z'_{r-1,3}$$

i.e., $Z'_{r,2} > Z'_{r-1,3}$ (9)

M_3 will start processing job r at time $= \max(Z'_{r,2}, Z'_{r-1,3})$

$$= Z'_{r,2} \tag{10}$$

Therefore, total elapsed time $\geq Z'_{r,2} + \sum_{i=1}^{n} p_{i,3}$

$$\geq Z'_{r-1,3} + \sum_{i=1}^{n} p_{i,3} = Z'_{n,3} = Z_{n,3}$$

Hence, total elapsed time will increase if M_2 starts processing jobs at time $H_2 > \min\{Y_k\}$;

k=1, 2, …, n.

By Theorem 3.1; the starting of processing jobs at time H_3 on M_3 will reduce the idle time of M_3 to zero and M_3 will be required only for time equivalent to the sum of the processing times of all the jobs on it. Therefore, total rental cost of M_3 will be minimum (least). Total rental cost of M_1 will always be minimum (least), since idle time of M_1 is always zero. Therefore, the objective is to minimize the rental cost of machine M_2.

In Branch-and-Bound technique provided by Lominiciki [19], if instead of terminating the procedure when the total elapsed time of a sequence is less than / equal to the value attached with all the unbranched vertices, the termination is done when the total elapsed time of a sequence is strictly less than the value, then all the other optimal sequences (if any) can be obtained.

The following algorithm provides the procedure to obtain the sequence which gives minimum possible rental cost while minimizing total elapsed time in three-machine flow-shop problem under Policy P_3.

4. Algorithm

Algorithm 4.1:

Step 1: Obtain all the sequences having minimum total elapsed time by Branch-and-Bound technique. Let these sequences be $S_1, S_2, \ldots, S_r$.

Step 2: Compute total elapsed time $Z_{n,3}(S_1)$.

Step 3: Compute rental time H_3 of M_3 for sequence S_1

$$H_3 = Z_{n,3}(S_1) - \sum_{i=1}^{n} p_{i,3}(S_1)$$

Step 4: For sequences $S_1, S_2, \ldots, S_r$, compute

(i) $Z_{n,2}(S_j)$

(ii) $Y_1(S_j) = H_3 - p_{1,2}(S_j)$

$$Y_k(S_j) = H_3 + \sum_{i=1}^{k-1} p_{i,3}(S_j) - \sum_{i=1}^{k} p_{i,2}(S_j)\text{; } k=2, 3, \ldots, n$$

(iii) $H_2(S_j) = \min\{Y_k(S_j)\}$; k=2, 3, …, n

(iv) $T_2(S_j) = Z_{n,2}(S_j) - H_2(S_j)$

j = 1, 2, …, r.

Step 5: Find $\min\{T_2(S_j)\}$; j=1, 2, …, r.

Let it be for sequence S_p, then S_p is the optimal sequence.

Step 6: Compute rental time H_2 of M_2 for sequence S_p

$H_2 = H_2(S_p)$.

Step 7: Compute total rental cost for sequence S_p

$$R(S_p) = \sum_{i=1}^{n} p_{i,1} \times C_1 + T_2(S_p) \times C_2 + \sum_{i=1}^{n} p_{i,3} \times C_3$$

5. EXAMPLE

Example 5.1: Consider 5-job, 3-machine flow-shop problem with processing times in hours as given in Table 1. The rental costs per hour for machines M_1, M_2 and M_3 are 5 units, 10 units and 8 units respectively.

Table 1. *Processing Times of Jobs on Machines*

Jobs	Machines		
	M_1	**M_2**	**M_3**
1	2	5	6
2	6	7	5
3	9	7	6
4	10	5	7
5	8	4	1

Applying Algorithm 4.1;

Step 1: Provides four sequences S_1 = 1-2-3-4-5; S_2 = 2-1-3-4-5; S_3 = 1-3-2-4-5 and S_4 = 1-3-4-2-5

Step 2: To obtain minimum total elapsed time, the completion time In-Out of sequence S_1 is given as in table 2.

Table 2. *Completion time In-Out for sequence S_1*

Jobs	Machines		
	M_1 In-Out	**M_2** In-Out	**M_3** In-Out
1	0-2	2-7	7-13
2	2-8	8-15	15-20
3	8-17	17-24	24-30
4	17-27	27-32	32-39
5	27-35	35-39	39-40

Therefore, minimum total elapsed time = 40 hours.

Step 3: Latest time at which machine M_3 should be taken on rent = $Z_{n,3}(S_1) - \sum_{i=1}^{n} p_{i,3}$

= 40 – 25 = 15 hours.

Step 4: To obtain the completion times of last job of sequences S_1 on machine M_2, its completion time In-Out table is given as in Table 3

S_1 = 1-2-3-4-5

Table 3. *Completion time In-Out*

Jobs	**Machines**	
	M_1 In-Out	**M_2 In-Out**
1	0-2	2-7
2	2-8	8-15
3	8-17	17-24
4	17-27	27-32
5	27-35	35-39

Therefore, $Z_{n,2}(S_1) = 39$

$Y_1(S_1) = H_3 - p_{1,2}(S_1) = 15 – 5 = 10$

$$Y_2(S_1) = H_3 + \sum_{i=1}^{1} p_{i,3}(S_1) - \sum_{i=1}^{2} p_{i,2}(S_1)$$

$= 15 + 6 – (5+7)$

$= 21 – 12 = 9$

$$Y_3(S_1) = H_3 + \sum_{i=1}^{2} p_{i,3}(S_1) - \sum_{i=1}^{3} p_{i,2}(S_1)$$

$= 15 + (6+5) – (5+7+7)$

$= 15 + 11 – 19 = 7$

$$Y_4(S_1) = H_3 + \sum_{i=1}^{3} p_{i,3}(S_1) - \sum_{i=1}^{4} p_{i,2}(S_1)$$

$= 15 + (6+5+6) – (5+7+7+5)$

$= 15 + 17 – 24 = 8$

$$Y_5(S_1) = H_3 + \sum_{i=1}^{4} p_{i,3}(S_1) - \sum_{i=1}^{5} p_{i,2}(S_1)$$

$= 15 + (6+5+6+7) – (5+7+7+5+4)$

$= 15 + 24 – 28 = 11$

$H_2(S_1) = \min \{Y_k(S_1)\}; \quad k=1, 2, \ldots, 5$

$= \min \{10, 9, 7, 8, 11\}$

$=7$

$T_2(S_1) = Z_{n,2}(S_1) - H_2(S_1)$

$= 39 – 7 = 32$

Similarly, applying Step 4 for sequences S_2, S_3 and S_4, we get

$T_2(S_2) = 32$; $T_2(S_3) = 31$ and $T_2(S_4) = 30$

Step 5: Minimum of $T_2(S_1)$; $T_2(S_2)$; $T_2(S_3)$ and $T_2(S_4) = \min\{32, 32, 31, 30\} = 30$ hours.

This minimum is corresponding to the sequence S_4 = 1-3-4-2-5. Therefore, 1-3-4-2-5 is the optimal sequence.

Step 6: Latest time at which machine M_2 should be taken on rent = 9 hours.

Step 7: Total rental cost for sequence 1-3-4-2-5 is

$$R(S_4) = \sum_{i=1}^{n} p_{i,1} \times C_1 + T_2(S_4) \times C_2 + \sum_{i=1}^{n} p_{i,3} \times C_3$$

$$= 35 \times 5 + 30 \times 10 + 25 \times 8$$

$$= 175 + 300 + 200 = 675 \text{ units}$$

Hence, 1-3-4-2-5 is the optimal sequence having total rental cost as 675 units, when M_1 is taken on rent (starts processing jobs) in the starting, M_2 is taken on rent (starts processing jobs) after 9 hours and M_3 is taken on rent (starts processing jobs) after 15 hours.

6. CONCLUSION

In this paper a Bi-criteria problem in 3-machine flow-shop problem is considered. The objective is to obtain a sequence which minimizes total elapsed time and gives total rental cost as minimum as possible. We have proved three theorems to find out the times at which machines should be taken on rent so that total elapsed do not change when we delay the processing of jobs on machines.

REFERENCES

[1]. Sen, T. and Dileepan, P., A bicriterion scheduling problem involving total flowtime and total tardiness", Journal of Information and Optimization Sciences, 20(2), 155-170, (1999).

[2]. Sen, T. and Gupta, S. K., A branch and bound procedure to solve a bicriterion scheduling problem, AIIE Transactions, 15, 84-88 (1983).

[3]. Sen, T. and Raiszadeh, F., An algorithm to minimize total flowtime and maximum job lateness in the two-machine flowshop system, Journal of Information and Optimization Sciences, 18(1), 9-16 (1997).

[4]. Smith, W. E., Various optimizers for single stage production, Naval Research Logistics, 3, 59-66 (1956).

[5]. Van Wassenhove, L. N. and Gelders, L. F., Solving a bicriterion scheduling problem European Journal of Operational Research, 4, 42-48 (1980).

[6]. Van Wassenhove, L. N. and Baker, K. R., A bicriterion approach to time/cost trade-off in sequence, European Journal of Operational Research, 11, 48-54 (1982).

[7]. Gupta, J. N. D. and Dudek, R. A., Optimality criteria for flowshop schedule, AIIE Transactions, 3(3), 199-205 (1971).

[8]. Dileepan, P. and Sen, T., Bicriterion state scheduling research for a single machine, Omega, 16, 53-59 (1988).

[9]. Chandersekharan Rajendran, Two-stage production scheduling problem with bicriteria", Operational Research Society, U. K., 43(9), 871-884 (1992).

[10]. Bagga, P. C. and Ambika Bhambani, Bicriteria in special flowshop problems, Indian Journal of Pure and Applied Mathematics, 33(4), 435-441 (2002).

[11]. Narain, L. and Bagga, P. C., Bi-criteria in $n \times m$ flow-shop problems, IAPQR Transactions, 31, 57-71 (2006).

[12]. Narain, L. and Bagga, P. C., Minimizing total elapsed time subject to zero idle time of machines in n×3 flow-shop problem, Indian Journal of Pure & Applied Mathematics, 34, 219-228 (2003).

[13]. Narain, L. and Bagga, P. C., Flow-shop/no-idle scheduling to minimize total elapsed time, Journal of Global Optimization, 33, 349-367 (2004).

[14]. Narain, L. and Bagga, P. C., Flow-shop/no-idle scheduling to minimize mean flow-time, ANZIAM Journal, 47, 265-275 (2005).

[15]. Bagga, P. C., Sequencing in a rental situation, Jr. of Canadian Operations Research Society, 7, 152-153 (1969).

[16]. Bagga, P. C. and Ambika Bhambani, Minimizing rental costs in three-machine sequencing problem, Jr. of Indian Association for Productivity, Quality and Reliability, 21, 73-77 (1996).

[17]. Bagga, P. C. and Khurana, K., Minimizing waiting cost of jobs in renting situations, Journal of Indian Association for Production, Quality and Reliability, 8(1), (1983).

[18]. Narain, L., Minimizing total hiring cost of machines in $n \times m$ flow-shop problem, Journal of Decision and Mathematical Sciences, 7, 23-32 (2002).

[19]. Lomnicki, Z. A., A branch-and-bound algorithm for the exact solution of the three- machine scheduling problem, Operational Research Quarterly, 16, 89-100 (1965).

AUTHOR'S BIOGRAPHY

Dr laxmi Narain is currently an Associate Professor in Department of Mathematics, Acharya Narendra Dev College, University of Delhi, Delhi, India. He has specialization in Optimization Technique. He has done Ph.D. in Flow-shop Sequencing Problem from Department of Mathematics, University of Delhi under the Guidance of Dr. P. C. Bagga. He has published research papers in National and International Journals.