Optimize Renting Times of Machines in Flow-Shop Scheduling

Laxmi Narain

Abstract— **This paper studies three-machine scheduling problems in the situation when one has got the assignment but does not have one's own machines and has to take machines on rent to complete the assignment. Minimization of total rental cost of machines may be the criterion in this type of situation. Here, we have considered a rental policy in which 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 some criterion. The objective is: for a given sequence obtain the latest times at which the machines should be taken on rent so that total rental cost is minimum without altering the total elapsed time. We have obtained a simple and efficient algorithm, without using Branch-and-Bound technique. Numerical example is given to illustrate the algorithm.**

Index Terms*— *Flow-shop, Scheduling, Idle Time, Completion Time, Elapsed Time, Rental Time, Rental Cost.

I. Introduction

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:

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

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

Policy III: *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 [1] studied three-machine problem under policy P_1 and provide the sequence to minimize the total rental cost of machines. Under P_2; for three-machine flow-shop problem, Bagga and Ambika [2] provided a Branch-and-Bound algorithm.

In this paper we are considering rental policy in bi-criteria scheduling problems 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 [3-16] 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 [6] surveyed the bicriteria scheduling research for a singe machine. Chandersekhran [5] 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 [4] 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 [8] 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 [10] 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 [9] 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.

This paper studies bi-criteria in three-machine flow-shop problems under rental ***Policy III***. In this paper, ***Policy III*** 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 minimum total elapsed. The objective is: Obtain the time at which machines should be taken on rent so that total rental cost as minimum as possible without altering the total elapsed time. 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 times at which machines should be taken on rent so that total rental cost as minimum as possible without altering the total elapsed time.

Numerical example is given to illustrate the algorithm.

II. MATHEMATICAL FORMULATION

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.

LAXMI NARAIN, Associate Professor, Department of Mathematics, Acharya Narendra Dev College, University of Delhi, Delhi, India.

$I_{i,j}(S)$: 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 $H_2(S)$.
$i = 1, 2, \ldots, n$ and $j= 1, 2, 3$.

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

Theorem 2.1: *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;

$$\begin{aligned}
Z'_{m+1,3} &= \max(Z_{m+1,2}, Z'_{m,3}) + p_{m+1,3} \\
&= \max\left(Z_{m+1,2}, H_3 + \sum_{i=1}^{m} p_{i,3}\right) + p_{m+1,3} \\
&= \max\left(Z_{m+1,2}, \sum_{i=1}^{m+1} I_{i,3} + \sum_{i=1}^{m} p_{i,3}\right) + p_{m+1,3} \\
&= \max\left(Z_{m+1,2}, \sum_{i=1}^{m} I_{i,3} + \sum_{i=1}^{m} p_{i,3} + I_{m+1,3}\right) + p_{m+1,3} \\
&= \max(Z_{m+1,2}, Z_{m,3} + \max(Z_{m+1,2} - Z_{m,3}, 0)) + p_{m+1,3} \\
&= \max(Z_{m+1,2}, \max(Z_{m+1,2}, Z_{m,3})) + p_{m+1,3} \\
&= \max(Z_{m+1,2}, Z_{m,3}) + p_{m+1,3} \\
&= Z_{m+1,3}
\end{aligned}$$

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 2.1: *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 2.3: *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: $H_2 = Y_r = \min\{Y_k\}$; $k=1, 2, \ldots, n$
For k = 1;
$Y_r = \min\{Y_k\}$; $k=1, 2, \ldots, n$
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;
$Y_r = \min\{Y_k\}$; $k=2, 3, \ldots, n$

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 2.1;

$Z_{k,2} \leq Z'_{k-1,3}$ … (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 2.4: *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

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

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

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

Now,

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

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 $= \max (Z'_{r,2}, Z'_{r-1,3})$
$= Z'_{r,2}$ … (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 2.1; 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.

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.

III. ALGORITHM

Algorithm 3.1:

Step 1: Let S be the given sequence.

Step 2: Compute $Z_{n,2}(S)$ and $Z_{n,3}(S)$.

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

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

Step 4: For sequence S, compute

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

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

k=2, 3, …, n

Step 5: Compute rental time H_2 of M_2 for sequence S

$H_2 = \min \{Y_k\}$; k=1, 2, …, n.

Step 6: Compute total rental cost for sequence S

$$R(S) = \sum_{i=1}^{n} p_{i,1} \times C_1 + (Z_{n,2}(S) - H_2) \times C_2 + (Z_{n,3}(S) - H_3) \times C_3$$

IV. EXAMPLE

Example 4.1: Consider the 10-Job, 3-Machine flow-shop problem with processing times in hours as given in Table 1. The rental costs per unit time for machines M_1, M_2 and M_3 are Rs. 50, Rs. 100 and Rs. 75 per hour respectively. Jobs are processed in the sequence 1-2-3-4-5-6-7-8-9-10.

Table 1: Processing Times of Jobs on Machines

Jobs	Machines		
	M_1	M_2	M_3

1	2	2	3
2	4	5	7
3	9	3	4
4	5	7	12
5	5	11	5
6	15	5	6
7	10	2	5
8	4	5	2
9	6	3	4
10	7	1	1

Applying Algorithm 3.1;
The given sequence S = 1-2-3-4-5-6-7-8-9-10 (Step 1). For determining the completion time of last job on machines M_2 and M_3, this sequence is enumerated and its completion time In-Out is given in Table 2.

Table 2: Completion Times In-Out for Sequence S

Jobs	**Machines**		
	M_1 In-Out	M_2 In-Out	M_3 In-Out
1	0-2	2-4	4-7
2	2-6	6-11	11-18
3	6-15	15-18	18-22
4	15-20	20-27	27-39
5	20-25	27-38	39-44
6	25-40	40-45	45-51
7	40-50	50-52	52-57
8	50-54	54-59	59-61
9	54-60	60-63	63-67
10	60-67	67-68	68-69

Thus, the completion time $Z_{10,2}$ = 68 hours and $Z_{10,3}$ = 69 hours (Step 2).
The rental time H_3 of machine M_3 for sequence S is

$$H_3 = Z_{10,3} - \sum_{i=1}^{10} p_{i,3}$$
$$= 69 - 49 = 20$$

Therefore, machine M_3 should be taken on rent after 20 hours of starting the processing of the first job on machine M_1 (Step 3).
For sequence S,

$$Y_1 = H_3 - p_{1,2}$$
$$= 20 - 2 = 18$$

$$Y_2 = H_3 + \sum_{i=1}^{1} p_{i,3} - \sum_{i=1}^{2} p_{i,2}$$
$$= 20 + 3 - 7 = 16$$

$$Y_3 = H_3 + \sum_{i=1}^{2} p_{i,3} - \sum_{i=1}^{3} p_{i,2}$$
$$= 20 + (3+7) - (2+5+3)$$
$$= 20 + 10 - 10 = 20$$

$$Y_4 = H_3 + \sum_{i=1}^{3} p_{i,3} - \sum_{i=1}^{4} p_{i,2}$$
$$= 20 + (3+7+4) - (2+5+3+7)$$
$$= 20 + 14 - 17 = 17$$

Continuing in this way,
$Y_5 = 18$; $Y_6 = 18$; $Y_7 = 22$; $Y_8 = 22$; $Y_9 = 21$ and $Y_{10} = 24$ (Step 4).
Rental time H_2 of machine M_2 for sequence S is

$$H_2 = \min \{Y_k\}$$
$$= \min \{18, 16, 20, 17, 18, 18, 22, 22, 21, 24\}$$
$$= 16$$

Therefore, machine M_2 should be taken on rent after 16 hours of starting the processing of the first job on machine M_1 (Step 5).
Total rental cost of machines for sequence S is

$$R(S) = \sum_{i=1}^{n} p_{i,1} \times C_1 + (Z_{n,2}(S) - H_2) \times C_2 + (Z_{n,3}(S) - H_3) \times C_3$$
$$= 67 \times 50 + (68-16) \times 100 + (69-20) \times 75$$
$$= 3350 + 52 \times 100 + 49 \times 75$$
$$= 3350 + 5200 + 3675$$
$$= 12225 \quad \text{(Step 6)}$$

Hence, for sequence S = 1-2-3-4-5-6-7-8-9-10 the minimum total rental cost is Rs. 12,225 without altering the total elapsed time (69 hours) when machine M_1 is taken on rent in the starting of processing the jobs, M_2 after 16 hours of starting the processing of first job on machine M_1 and M_3 after 20 hours of starting the processing of first job on machine

V. CONCLUSION

In this paper, 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. A simple and efficient algorithm is developed by using these theorems which provide the times at which machines should be taken on rent so that total rental cost is as minimum as possible without altering the total elapsed time.

REFERENCES

[1] Bagga, P.C., "Sequencing in a rental situation", *Jr. of Canadian Operations Research Society, vol.* 7, 1969, pp. 152-153.
[2] Bagga, P.C. and Ambika Bhambani, " Minimizing rental costs in three-machine sequencing problem", *Jr. of Indian Association for Productivity, Quality and Reliability*, vol. 21, 1996, pp. 73-77.
[3] Bagga, P.C. and Ambika Bhambani, " Bicriteria in flowshop scheduling problems", *Journal of Combinatorics, Information and System Sciences*, vol. 21, no. 3-4, 1996.
[4] Bagga, P.C. and Ambika Bhambani, "Bicriteria in special flowshop problems", *Indian Journal Pure and Applied Mathematics*, vol. 33, no. 4, 2002, pp. 435- 441.
[5] Chandersekharan Rajendran, "Two-stage production scheduling problem with bicriteria", *Operational Research Society, U. K*, vol. 43, no. 9, 1992, pp. 871-884.
[6] Dileepan, P. and Sen, T, "Bicriterion state scheduling research for a single machine", *Omega*, vol. 16, 1988, pp. 53-59.
[7] Gupta, J.N.D. and Dudek, R.A, "Optimality criteria for flowshop schedule", *AIIE Transactions*, vol. 3, no. 3, 1971, pp. 199-205.
[8] 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*, vol. 34, 2003, pp. 219-228.
[9] Narain, L. and Bagga, P. C., "Flow-shop/no-idle scheduling to minimize total elapsed time", *Journal of Global Optimization*, vol. 33, 2004, pp. 349-367.

[10] Narain, L. and Bagga, P. C., "Flow-shop/no-idle scheduling to minimize mean flow-time", *ANZIAM. Journal*, vol. 47, 2005, pp. 265- 275.

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

[12] Sen, T. and Gupta, S.K., "A branch and bound procedure to solve a bi-criterion scheduling problem", *AIIE Transactions*, vol. 15, 1983, pp. 84- 88.

[13] Sen, T. and Dileepan, P., "A bicriterion scheduling problem involving total flowtime and total tardiness", *Journal of Information and Optimization Sciences*, vol. 20, no. 2, 1999, pp. 155-170

[14] 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*, vol. 18, no. 1, 1997, pp. 9-16.

[15] Sen, T., Raiszadeh, F. and Dileepan, P., "A branch-and-bound approach to the bicriterion scheduling problem involving total flowtime and range of lateness", *Management Science*, vol. 34, no. 2, 1988, pp. 254-260.

[16] Van Wassenhove, L. N. and Gelders, L. F., "Solving a bicriterion scheduling problem", *European Journal of Operational Research*, vol. 4, 1980, pp. 42- 48.

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

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.