Advances in Theoretical and Applied Mathematics
ISSN 0973-4554 Volume 11, Number 4 (2016), pp. 485-494

http://www.ripublication.com

Scheduling in Rental Situation with Unavailability Constraint

Laxmi Narain

(Associate Professor, Department of Mathematics, Acharya Narendra Dev College, University of Delhi, Delhi, India, e-mail: laxminarain@andc.du.ac.in)

Abstract

This paper studies n-job, 3-machine flow-shop problems under the circumstances when one has got the assignment but does not have one's own machines and has to take machines on rent. Minimization of total rental cost of machines will be the criterion in such situation. The objective is to obtain a sequence which minimize total rental cost with unavailability constraint on each machine. In this paper we consider the rental policy under which all the machines are taken on rent simultaneously in the starting of processing the jobs and are returned as and when a machine is no longer required for processing the jobs. Under unavailability constraint, it is assumed that unavailable time is known in advance (deterministic) and if a job cannot finish before the unavailable period of a machine then job can continue after the machine is available again (resumable). An algorithm is developed using Branch-and-Bound technique to solve the problem optimally. The Algorithm is illustrated through a numerical example.

Keywords: Flow-shop, Scheduling, Deterministic, Resumable, Rental Cost

1. INTRODUCTION

In real life, situation may arise, when one has got the assignment but does not have one's own machines and does not have money or does not want to take risk of investing 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 such situation.

Generally, the following renting policies exist:

- **I** : All the machines are taken on rent at one time in the starting and are returned also at one time when last job completes processing on last machine.
- II : All the machines are taken on rent at one time in the starting and are returned as and when they are no longer required for processing.
- III : All the machines are taken on rent as and when they are required for processing and are returned as and when they are no longer required for processing.

In literature it is generally considered that machines are always available for processing but in real industry settings, machines may not be available for processing during the scheduling period due to breakdown (stochastic) or due to preventive maintenance (deterministic). Adiri et al. [1] studied a single-machine problem with machine breakdown. They studied both the stochastic case and the deterministic case. For the deterministic case, the objective function was minimization of total completion time under the assumption that a job must be restarted if it did not finish before the breakdown. Lee and Liman [6] studied the two parallel-machine scheduling problem with an availability constraint on one-machine and with the objective function of total completion time. They used dynamic programming to solve it.

Lee [5] studied the machine scheduling problem with an availability constraint under different performance measures (total elapsed time, total weighted completion time, tardiness, and number of tardy jobs) for other machine situations, including single machine and parallel machines. Lee [4] studied the two-machine flow-shop problem in deterministic environment under two cases; availability constraint on machine M_1 only, and on machine M_2 only. He provided pseudo-polynomial dynamic programming algorithm to solve the problem optimally. Narain [12] studied 2-machine flow-shop problem with an unavailability constraint on each machine problem. He provided an algorithm using Branch-and bound technique and Johnson's Algorithm to solve the problem optimally.

Authors [2, 3, 8 – 11, 13 – 15] studied the above rental policies under the assumption that machines are always available. In this paper, we consider the case where machines may not be available due to preventive maintenance. Here, the objective is to obtain a sequence which minimize total rental cost of machines under rental policy

II with an unavailability constraint on each machine. Under rental policy II, all the machines are taken on rent simultaneously in the starting of processing the jobs and are returned as and when a machine is no longer required for processing the jobs. Under unavailability constraint, it is assumed that unavailable time is known in advance (deterministic) and if a job cannot finish before the unavailable period of a machine then job can continue after the machine is available again (resumable). An algorithm is developed using Branch-and-Bound technique to solve the problem optimally. The Algorithm is illustrated through a numerical example.

2. NOTATIONS

S	: Sequence of jobs 1, 2, …, n.
M_j	: Machine j, j = 1, 2, 3.
s_j, t_j	: Machine M_j is unavailable from s_j to t_j, where $0 \leq s_j \leq t_j$.
p_{ij}	: Processing time for job i on machine M_j.
I_{ij}	: Idle time of machine M_j for job i.
C_j	: Rental cost of machine M_j.
J_r	: Partial schedule of r scheduled jobs.
J_r'	: Set of remaining (n-r) free jobs.
$t(J_r, j)$	: The time when the last job of the assigned schedule J_r is completed on machine M_j, where j = 1, 2, 3.
t(S, j)	: The time when the last job of the sequence S is completed on machine M_j, where j = 1, 2, 3.
LB[Jr]	: Lower bound corresponding to partial schedule J_r, irrespective of any schedule of J_r'.

3. MATHEMATICAL FORMULATION

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

Without any loss of generality, we can assume that the jobs are processed according to sequence S, where S = 1, 2, …, n.

For sequence S, total rental cost of machines is

$$R(S) = \sum_{j=1}^{3}\sum_{i=1}^{n}\left(p_{ij} + I_{ij}\right)x\ C_j$$

$$= \sum_{i=1}^{n}[(p_{i1} + I_{i1})x\ C_1 + (p_{i2} + I_{i2})x\ C_2 + (p_{i3} + I_{i3})x\ 3]$$

$$= \sum_{i=1}^{n} p_{i1}\ x\ C_1 + \sum_{i=1}^{n} I_{i1}\ x\ C_1 + \sum_{i=1}^{n} p_{i2}\ x\ C_2 + \sum_{i=1}^{n} I_{i2}\ x\ C_2 + \sum_{i=1}^{n} p_{i3}\ x\ C_3 + \sum_{i=1}^{n} I_{i3}\ x\ C_3$$

Under rental policy II, $\sum_{i=1}^{n} I_{i1} = t_1 - s_1$, which is constant.

Therefore, total rental cost of machines

$$R(S) = \sum_{i=1}^{n} p_{i1}\ x\ C_1 + (t_1 - s_1)\ x\ C_1 + \sum_{i=1}^{n} p_{i2}\ x\ C_2 + \sum_{i=1}^{n} I_{i2}\ x\ C_2 + \sum_{i=1}^{n} p_{i3}\ x\ C_3 + \sum_{i=1}^{n} I_{i3}\ x\ C_3$$

Here processing times p_{ij} and rental costs C_j are constants $\forall$ i and j.

Therefore, total rental cost of machines is minimum when $\sum_{i=1}^{n} I_{i2}\ x\ C_2 + \sum_{i=1}^{n} I_{i3}\ x\ C_3$ is minimum.

Here $\sum_{i=1}^{n} I_{i2}$ includes $t_2 - s_2$ and $\sum_{i=1}^{n} I_{i3}$ includes $t_3 - s_3$ also.

Evaluation of Lower Bound

The lower bound for any partial schedule J_r is the **Value** such that whatever be the order of the remaining jobs to follow J_r the total rental cost of machines should never be the less than the **Value.** The lower bound for any partial schedule J_r is obtained under the assumption that jobs of J_r' does not wait for processing on particular machine and jobs after completing the processing on this machine are not waiting for processing on the remaining machine as if the machines are always available for processing. This reduces the problem of a single machine processing and the criterion for obtaining the optimal order of single machine can be obtained.

4. ALGORITHM

The Branch-and-Bound technique is applied to minimize the total rental cost of machines. The lower bound LB[J_r] for any partial schedule J_r is evaluated through the following steps:

Step 1: Compute

(i) $$g_1 = \begin{cases} t(J_r, 1) + \sum_{i \in J'_r} p_{i1} + \min_{i \in J'_r} p_{i2} + (t_1 - s_1); & \text{if } t(J_r, 1) \leq s_1 \\ t(J_r, 1) + \sum_{i \in J'_r} p_{i1} + \min_{i \in J'_r} p_{i2}; & \text{otherwise} \end{cases}$$

(ii) $$g_2 = \begin{cases} t(J_r, 2) + \sum_{i \in J'_r} p_{i2} + (t_2 - s_2); & \text{if } t(J_r, 2) \leq s_2 \\ t(J_r, 2) + \sum_{i \in J'_r} p_{i2}; & \text{otherwise} \end{cases}$$

(iii) $g = \max[g_1, g_2]$

Step 2: Compute

(i) $$G_1 = \begin{cases} t(J_r, 1) + \sum_{i \in J'_r} p_{i1} + \min_{i \in J'_r} [p_{i2} + p_{i3}] + (t_1 - s_1); & \text{if } t(J_r, 1) \leq s_1 \\ t(J_r, 1) + \sum_{i \in J'_r} p_{i1} + \min_{i \in J'_r} [p_{i2} + p_{i3}]; & \text{otherwise} \end{cases}$$

(ii) $$G_2 = \begin{cases} t(J_r, 2) + \sum_{i \in J'_r} p_{i2} + \min_{i \in J'_r} p_{i3} + (t_2 - s_2); & \text{if } t(J_r, 2) \leq s_2 \\ t(J_r, 2) + \sum_{i \in J'_r} p_{i2} + \min_{i \in J'_r} p_{i3}; & \text{otherwise} \end{cases}$$

(iii) $$G_3 = \begin{cases} t(J_r, 3) + \sum_{i \in J'_r} p_{i3} + (t_3 - s_3); & \text{if } t(J_r, 3) \leq s_3 \\ t(J_r, 3) + \sum_{i \in J'_r} p_{i3}; & \text{otherwise} \end{cases}$$

(iv) $G = \max[G_1, G_2, G_3]$

Step 3: Compute

$$\text{LB}[J_r] = g \times C_2 + G \times C_3$$

5. EXAMPLE

Consider 4-job, 3-machine scheduling problem with processing times as given in Table: T1. It is also given that M_1 will not be available for 9 units of time from 7 to 16, M_2 will not be available for 4 units of time from 15 to 19 and M_3 will not be available for 2 units of time from 22 to 24 i.e., $s_1 = 7$, $t_1 = 16$, $s_2 = 15$, $t_2 = 19$ and $s_3 = 22$, $t_3 = 24$. The rental cost per unit time for M_1, M_2 and M_3 is 10 units, 20 units and 15 units respectively. i.e., $C_1 = 10$, $C_2 = 20$ and $C_3 = 15$.

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

Table: T1

Applying algorithm;

For J_r = (1),

Step 1: $t(J_r, 1) = 5$, $t(J_r, 2) = 5 + 6 + 4 = 15$, $t(J_r, 3) = 15 + 3 = 18$

$$\text{(i) } g_1 = \begin{cases} t(J_r, 1) + \sum_{i\in J_r'} p_{i1} + \min_{i\in J_r'} p_{i2} + (t_1 - s_1); & \text{if } t(J_r, 1) \le s_1 \\ t(J_r, 1) + \sum_{i\in J_r'} p_{i1} + \min_{i\in J_r'} p_{i2}; & \text{otherwise} \end{cases}$$

$= 5 + (6+4+10) + \min\{8, 8, 5\} + (16 - 7)$

$= 25 + 5 + 9 = 39$

$$\text{(ii) } g_2 = \begin{cases} t(J_r, 2) + \sum_{i\in J_r'} p_{i2} + (t_2 - s_2); & \text{if } t(J_r, 2) \le s_2 \\ t(J_r, 2) + \sum_{i\in J_r'} p_{i2}; & \text{otherwise} \end{cases}$$

$= 15 + (8 + 8 + 5)$

$= 15 + 21 = 36$

$$\text{(iii) } g = \max[g_1, g_2]$$

$= \max\{39, 36\} = 39$

Step 2:

(i) $$G_1 = \begin{cases} t(J_r, 1) + \sum_{i \in J_r'} p_{i1} + \min\limits_{i \in J_r'} [p_{i2} + p_{i3}] + (t_1 - s_1); & \text{if } t(J_r, 1) \leq s_1 \\ t(J_r, 1) + \sum_{i \in J_r'} p_{i1} + \min\limits_{i \in J_r'} [p_{i2} + p_{i3}]; & \text{otherwise} \end{cases}$$

= 5 + (6+4+10) + min [17, 14, 12] + (16 – 7)
= 5 + 20 + 12 + 9 = 46

(ii) $$G_2 = \begin{cases} t(J_r, 2) + \sum_{i \in J_r'} p_{i2} + \min\limits_{i \in J_r'} p_{i3} + (t_2 - s_2); & \text{if } t(J_r, 2) \leq s_2 \\ t(J_r, 2) + \sum_{i \in J_r'} p_{i2} + \min\limits_{i \in J_r'} p_{i3}; & \text{otherwise} \end{cases}$$

= 15 + (8 + 8 + 5) + min [9, 6, 7]
= 15 + 21 +6 = 42

(iii) $$G_3 = \begin{cases} t(J_r, 3) + \sum_{i \in J_r'} p_{i3} + (t_3 - s_3); & \text{if } t(J_r, 3) \leq s_3 \\ t(J_r, 3) + \sum_{i \in J_r'} p_{i3}; & \text{otherwise} \end{cases}$$

= 18 + (9 + 6 + 7) + (24 – 22)
= 18 + 22 + 2 = 42

(iv) $G = \max [G_1, G_2, G_3]$
= max {46, 42, 42}
= 46

Step 3:

$LB[J_r] = g \times C_2 + G \times C_3$
= 39 x 20 + 46 x 15
= 780 + 690 = 1470

Similarly, the lower bounds for partial schedule J_r = (2), (3) and (4) are 1440, 1425 and 1580 units respectively. Minimum value of lower bound is 1425 for J_r = (3). Therefore,

J_r = (3) is the branching node.

Continuing in this way, the Branch-and-Bound algorithm is applied for evaluations of relevant lower bounds as given in Table: T2.

J_r	g	G	LB[J_r]
1	39	46	1470
2	39	44	1440
3	39	43	1425
4	43	48	1580
31	39	46	1470
32	39	46	1470
34	42	47	1545
21	39	46	1470
23	39	47	1485
24	44	47	1585
2134	—	—	1485
2143	—	—	1595
3124	—	—	1500
3142	—	—	1560
3214	—	—	1470
3241	—	—	1490

Table: T2

The complete scheduling tree is shown in Figure: F1.

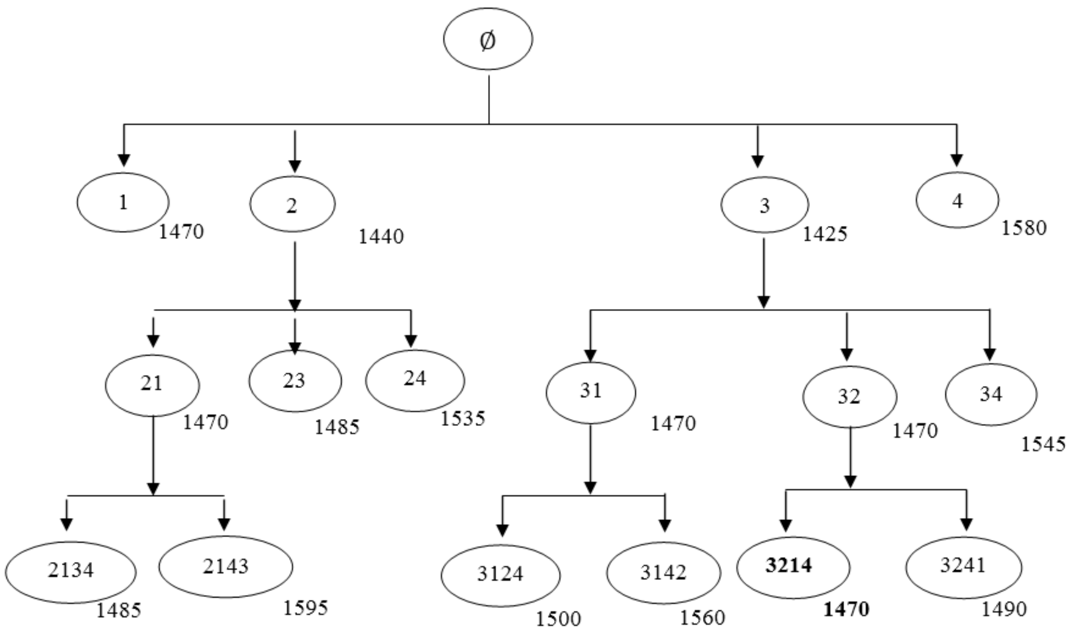

Figure: F1 Scheduling Tree

Total rental cost of machines = 1470 + 34 x 10
= 1470 + 340 = 1810

Hence, 3-2-1-4 is the optimal sequence with minimum total rental cost of machines as 1810 units.

6. FUTURE SCOPE

This paper studies 3-machine flow-shop problem with one availability constrain on each machine. This can be extended with more than one availability constraint on each machine. This can also be extended to general m-machine flow-shop problems.

REFERENCES

[1] Adiri, I., Bruno, J., Frostig, E. and RinnooyKan, A.H.G, 1989, "Single Machine Flow-time Scheduling with a Single Breakdown", Acta Inform., 26, pp. 679–696.

[2] Bagga, P.C., 1969, "Sequencing in a Rental Situation", Jr. of Canadian Operations Research Society, 7, pp. 152-153.

[3] Bagga, P.C. and Ambika Bhambani, 1996, "Minimizing Rental Costs in Three-machine Sequencing Problem", Jr. of Indian Association for Productivity, Quality and Reliability, 21, pp. 73-77.

[4] Lee, C.Y., 1997, "Minimizing the Make-span in the Two-machine Flow-shop Scheduling Problem with an Availability Constraint", Operations Research Letters, 20, pp. 129-139.

[5] Lee, C.Y., 1996, "Machine Scheduling with an Availability Constraint", Journal of Global Optimization, 9, pp. 395-416.

[6] Lee, C.Y. and Liman, S.D., 1992, "Single Machine Flow-time Scheduling with Scheduled Maintenance", Acta Inform., 29, pp. 375-382.

[7] Lomnicki, Z.A., 1965, "A Branch-and-Bound Algorithm for the Exact Solution of the Three-machine Scheduling Problem", Operational Research Quarterly, 16, pp. 89-100.

[8] Narain, L., 2015, "Minimizing Rental Cost in 2-Machine Flow-shop Sequencing Problem", International Journal of Management and Social Sciences Research (IJMSSR), 4(6), pp. 16-20.

[9] Narain, L., 2015, "Minimizing Rental Cost in Three-Machine Flow-shop Problems", International Journal of Innovative Research in Computer Science & Technology (IJIRCST), 3(3), 139-142.

[10] Narain, L., 2015 "Minimizing Rental Cost in 3-Machine Specially Structured Flow-shop Problems", Int. J. Engg. Res. & Sci. & Tech., 4(3), pp. 1-6.

[11] Narain, L., 2015, "Optimize Renting Times of Machines in Flow-Shop Scheduling", International Journal of Engineering and Applied Sciences (IJEAS), 2(5), 84-88.

[12] Narain, L., 2003, “Two- machine Flow-shop Problems with Availability Constraint on Each Machine”, Journal of Indian Society of Statistics and Operations Research, 24, pp. 17-24.

[13] Narain, L. and Bagga, P. C., 2005, “Flow-shop/No-idle Scheduling to Minimize Total Elapsed Time”, Journal of Global Optimization, 33, pp. 349-367.

[14] Narain, L. and Bagga, P. C., 2005, “Flow-shop/No-idle Scheduling to Minimize Mean Flowtime”, ANZIAM (Journal of Australian Mathematical Society), 47, pp. 265-275.

[15] Narain, L. and Bagga, P.C., 2003, “Minimizing Total Elapsed Time subject to Zero Idle Time of Machines in n × 3 Flow-shop Problem", Indian Journal of Pure & Applied Mathematics, 34, pp. 219-228.