Indian J. pure appl. Math., 34(2) : 219-228, February 2003

MINIMIZING TOTAL ELAPSED TIME SUBJECT TO ZERO TOTAL IDLE TIME OF MACHINES IN $n \times 3$ FLOWSHOP PROBLEM

LAXMI NARAIN

Department of Mathematics, University of Delhi

AND

P. C. BAGGA

Moti Lal Nehru College (E), University of Delhi

(*Received* 25 *January* 2001; *after revision* 18 *April* 2002; *accepted* 5 *July* 2002)

This paper studies $n \times 3$ flowshop problem when machines are taken on rent. The objective is to 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.

Key Words : Sequencing Theory; Completion Time; Elapsed Time; Optimization; Hiring Cost; Hiring Time

1. INTRODUCTION

Many research papers exist in literature which deal with $n \times m$ flowshop problems with minimization of total elapsed time as the criterion. The situation can occur in practice when one undertakes the project of processing the jobs but does not have one's own machines. For example, in a realistic situation, if a programmer gets the work of computerising examination results of a board/university, he may need three machines, viz., data entry machine, computer and printer, to be taken on rent. In this case, the objective can be: to determine the job sequence which minimizes the total elapsed time subject to zero total idle time of machines. Here the machines will be taken on rent for minimum possible time.

$$\text{We know that total hiring cost} = \sum_{j=1}^{3} \sum_{i=1}^{n} [p\,(i,j) + I\,(i,j)] \times C_j,$$

where $p(i, j)$ is the processing time of job i on machine j, $I\ (i, j)$ is the idle time of machine j for job i and C_j is hiring cost per unit time of machine j.

The processing times $p(i, j)$ and hiring costs C_j are constants so we can only reduce idle times $I\ (i, j)$ for $i = 1, 2, ..., n$; $j = 1, 2, 3$. Therefore, the total hiring cost of the machines will be minimum when the idle times of all the machines will be minimum.

The following hiring policies generally exist :

Policy 1 : All the machines are taken on rent at one time and are returned also at one time.

Policy 2 : All the machines are taken on rent at one time and are returned as and when they are no longer required for processing.

Policy 3 : All the machines are taken on rent as and when they are required and are returned as soon as they complete the last job.

Under Policy 1; obviously, the sequence which minimizes the total elapsed time will also minimize the total hiring cost. For two-machine problem, Johnson[3] and for three-machine problem, Lomnicki[4] or Bagga[1] minimize the total hiring cost.

Under Policy 2; for two-machine problem, Johnson's sequence will minimie the total elapsed time and will also minimize the total hiring cost. For three-machine problem, Bagga and Ambika[2] study the problem when the machines are taken on rent for processing the jobs when the rental costs of the machines are different and the objective being that of minimization of total rental cost of the machines. The problem is solved by applying the branch-and-bound technique.

Under Policy 1 and Policy 2; all the machines are taken on rent at one time (i.e., in the starting) so there will be idle times on machine 2 and machine 3. Therefore, it is not possible to have zero total idle time of machines under Policy 1 and Policy 2.

Under Policy 3; we can have zero total idle time of machines. For two-machine problem, the result is quite trivial. Take the Johnson's sequence which minimize the total elapsed time. The 2nd machine can be hired at time H_2, where H_2 = Total elapsed time — Sum of the processing times of all the jobs on 2nd machine.

The starting of the job at time H_2 on the 2nd machine will reduce the idle time on the 2nd machine to zero and this will be required only for the time equal to the sum of the processing times of all the jobs on it. Since the idle time on 1st machine is always zero so 1st machine is required only for the time equal to the sum of the processing times of all the jobs on it. Therefore, 1st machine can be hired at the time when work starts and can be returned as soon as the last job is completed on it. Hence for two-machine problem, the Johnson's sequence will minimize the total elapsed time subject to zero total idle time of machines.

For three-machine problem, it is not always true that the sequence which has the minimum total elapsed time will also have the minimum total elapsed time subject to zero total idle time of machines. For example, consider a 4-job, 3-machine problem whose processing times (in hours) are given as in Table I.

TABLE I :

Jobs	Machines		
	1	2	3
1	5	4	5
2	6	3	6
3	7	2	5
4	4	3	1

In this problem, for sequence J_4 = (1234) the minimum total elapsed time is 26 hours but the minimum total elapsed time subject to zero total idle time of machines is 34 hours for sequences J_4 = (2143), (2413), (3124), (3142) and (3214) the minimum total elapsed time subject to zero total idle time of machines is 32 hours. Hence it is not always true that the sequence which has minimum total elapsed time will also have the minimum total elapsed time subject to zero total idle time of machines.

This paper provides an algorithm to determine the sequence which minimizes the total elapsed time subject to zero total idle time of machines under policy 3 for a general $n \times 3$ flowshop problem. The algorithm is illustrated through a numerical example.

2. Mathematical Formulation

Notations

$p(i, j)$ = The processing time of job i on machine j.

$Z(i, j)$ = The completion time of job i on machine j when all the machines are taken at the same time.

$I(i, j)$ = The idle time of machine j for job i.

H_j = The time when machine j is hired.

$Z'(i,j)$ = The completion time of job i on machine j when machine j starts processing jobs at time H_j.

I_r = max $[Z'(1, 2)-H_3,\ Z'(2, 2) - Z'(1, 3),\ \ldots,\ Z'(r,2)-Z'(r-1,3), 0]$ where $r = 1, 2, 3, \ldots, n$.

$Z''(i, 3)$ = The completion time of job i on machine 3 when it starts processing jobs at time H_3+I_r, where $i = 1, 2, 3, \ldots, r$.

$I''(i,3)$ = The idle time of machine 3 for job i when machine 3 starts processing jobs at time H_3+I_r.

J_r = The partial schedule of r scheduled jobs, where $r = 1, 2, 3, \ldots, n$.

J'_r = The 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 $j = 1, 2, 3$.

$$g_1 = t(J_r, 1) + \sum_{i \in J'_r} p(i, 1) + \min_{i \in J'_r} [p(i, 2) + p(i, 3)]$$

$$g_2 = t(J_r, 2) + \sum_{i \in J'_r} p(i, 2) + \min_{i \in J'_r} p(i, 3)$$

$$g_3 = t(J_r, 3) + \sum_{i \in J'_r} p(i, 3)$$

Without any loss of generality we can consider the job sequence $S = (123 \ldots n)$ in the proof of the following theorem[8]

Theorem 2.1 — *If we start working on machine 2 at time* $H_2 = \sum_{i=1}^{k} I(i, 2)$ *then* $Z(k, 2)$ *will remain unaltered.*

PROOF : Let $Z'(i, 2)$ be the completion time of job i on machine 2 when it starts processing jobs at time H_2. The proof of the theorem is based on the method of mathematical induction.

For $k = 1$, we have

$$Z'(1, 2) = H_2 + p(1, 2)$$

$$= I(1, 2) + p(1, 2)$$

$$= p(1, 1) + p(1, 2)$$

$$= Z(1, 2).$$

Therefore, the result holds for $k = 1$.

Let the result hold for $k = m$.

For $k = m + 1$, we have

$$Z'(m+1, 2) = \max[Z(m+1, 1), Z'(m, 2)] + p(m+1, 2)$$

$$= \max[Z(m+1, 1), Z(m, 2) + I(m+1, 2)] + p(m+1, 2)$$

$$= \max[Z(m+1, 1), Z(m, 2) + \max$$

$$[Z(m+1, 1) - Z(m, 2), 0]] + p(m+1, 2)$$

$$= \max[Z(m+1, 1) + \max[Z(m+1, 1), Z(m, 2)]] + p(m+1, 2)$$

$$= \max[Z(m+1, 1), Z(m, 2)] + p(m+1, 2)$$

$$= Z(m+1, 2).$$

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

Hence by mathematical induction the theorem holds for all k, where $k = 1, 2, 3, ..., n$.

Thus if machine 2 is hired at time H_2, where

$$H_2 = Z(n, 2) - \sum_{i=1}^{n} p(i, 2)$$

$$= \sum_{i=1}^{n} I(i, 2)$$

then $Z(n, 2)$ is not altered and machine 2 is engaged for minimum time equal to the sum of the processing times of all the jobs on machine 2. Hence, when machine 2 starts working at time H_2, then there is no idle time on machine 2.

***Theorem* 2.2** — *If we start working on machine 2 at time* $H_2 = I(1, 2)$ *and on machine 3 at time* $H_3 = \sum_{i=1}^{k} I(i, 3)$ *then Z (k, 3) will remain unaltered and also the idle time on machine 3 will be zero.*

PROOF : Proof is on the same line as in Theorem 2.1.

***Theorem* 2.3** — *If we start working on machine 2 at time* $H_2 = \sum_{i=1}^{k} I(i, 2)$ and on machine 3 at time $(H_3 + I_k)$, where $H_3 = \sum_{i=1}^{k} I(i, 3)$ and

$$I_k = \max [Z'(1, 3) - H_3, Z'(2, 2) - Z'(1, 3), ..., Z'(k, 2) - Z'(k-1, 3), 0]$$

then $Z''(k, 3) = Z(k, 3) + I_k$ and idle time on machines is zero.

PROOF : Let $Z'(i, 2)$ be the completion time of job i on machine 2 when it starts processing jobs at time H_2. Let $Z'(i, 3)$ be the completion time of job i on machine 3 when it starts processing jobs at time H_3. Let $Z''(i, 3)$ be the completion time of job i on machine 3 when it starts processing jobs at time $H_3 + I_k$.

The proof of the theorem is based on the method of mathematical induction.

For $k = 1$, we have

$$H_3 + I_1 = H_3 + \max [Z'(1, 2) - H_3, 0]$$

$$\geq H_3 + Z'(1, 2) - H_3$$

$$\geq Z'(1, 2)$$

$$\therefore \; I''(1, 3) = \max [Z'(1, 2) - (H_3 + I_1), 0] = 0$$

and

$$Z''(1, 3) = \max [H_3 + I_1, Z'(1, 2)] + p(1, 3)$$

$$= H_3 + I_1 + p(1, 3)$$

$$= H_3 + p(1, 3) + I_1$$

$$= Z(1, 3) + I_1$$

Therefore, the result holds for $k = 1$.

Let the result hold for $k = m$, then

$$I''(m, 3) = 0$$

i.e., $$Z''(m-1, 3) \geq Z'(m, 2)$$

and $$Z''(m, 3) = Z(m, 3) + I_m$$

For $k = m + 1$, we have

$$Z''(m, 3) = \max [Z''(m-1, 3), Z'(m, 2)] + p(m, 3)$$

$$= \max [Z''(m-1, 3) + p(m, 3), Z'(m, 2) + p(m, 3)]$$

$$= \max \left[H_3 + I_{m+1} + \sum_{i=1}^{m-1} p(i, 3) + p(m, 3), Z'(m, 2) + p(m, 3) \right]$$

$$= \max [Z'(m, 3) + I_{m+1}, Z'(m, 2) + p(m, 3)]$$

$$= \max [Z'(m, 3) + \max [Z'(m+1, 2) - Z'(m, 3), Z(m, 2) - Z'(m-1, 3),$$
$$\ldots, Z'(1, 2) - H_3, 0], Z'(m, 2) + p(m, 3)]$$

$$\geq \max [Z'(m, 3) + Z'(m+1, 2) - Z' *m, 3), Z'(m, 2) + p(m, 3)]$$

$$= \max [Z'(m+1, 2), Z'(m, 2) + p(m, 3)]$$

$$\geq Z'(m+1, 2)$$

$$\therefore \quad I''(m+1, 3) = \max [Z'(m+1, 2) - Z''(m, 3), 0] = 0$$

and $$Z''(m+1, 3) = \max [Z''(m, 3), Z'(m+1, 2)] + p(m+1, 3)$$

$$= Z''(m, 3) + p(m+1, 3)$$

$$= H_3 + I_{m+1} + \sum_{i=1}^{m} p(i, 3) + p(m+1), 3\}$$

$$= Z(m+1, 3) + I_{m+1}$$

Therefore, the result holds for $k = m + 1$ also,
Hence, by mathematical induction the theorem holds for all k,. where $k = 1, 2, 3, \ldots, n$.

3. Branch-and-Bound Algorithm

The branch-and-bound technique is explained by Lomnicki[4]. The lower bounds for any partial schedule J_r are evaluated through the following steps :

Step 1 — Calculate

$$(i)\ g_1 = t(J_r, 1) + \sum_{i \in J_r'} p(i, 1) + \min_{i \in J_r'} [p(i, 2) + p(i, 3)].$$

$$(ii)\ g_2 = t(J_r, 2) + \sum_{i \in J_r'} p(i, 2) + \min_{i \in J_r'} p(i, 3)$$

$$(iii)\ g_3 = t(J_r, 3) + \sum_{i \in J_r'} p(i, 3)$$

where g_1, g_2 and g_3 are the same as described by Lomnicki[4].

Step 2 — Define

$$g = \max [g_1, g_2, g_3]$$

Step 3 — Define

$$I_r = \max [Z'(1, 2) - H_3, Z'(2\ 2) - Z'(1, 3), .., Z'(r, 2) - Z'(r-1, 3), 0]$$

where I_r is the idle time on machine 3 for schedule J_r when machine 2 is hired at time H_2 and machine 3 at time H_3.

If $g > g_3$ then there is at least $g - g_3$ idle time on machine 3. By Theorem 2.2, machine 3 can be hired at time $H_3 + (g - g_3)$ without changing g. Therefore, when machine 3 is hired at time $H_3 + (g - g_3)$ then idle time on machine 3 is $\max [I_r - (g - g_3), 0]$.

Step 4 — Define

$$G = g + \max [I_r - (g - g_3), 0]$$

where G is the lower bound for any partial schedule J_r of r jobs.

4. Example

Example 4.1 — Consider a 5-job, 3-machine sequencing problem whose processing times (in hours) are given as in Table II.

TABLE II :

	Machines		
Jobs	1	2	3
1	2	5	13
2	7	10	9
3	9	11	5
4	8	8	10
5	9	2	1

Step 1 — When $J_1 = (1)$, we have

$$t(J_1, 1) = 2\ t(J_1, 2) = 2 + 5 = 7, t(J_1, 3) = 2 + 5 + 13 = 20$$

$$g_1 = t(J_1, 1) + \sum_{i \in J_1'} p(i, 1) + \min_{i \in J_1'} [p(i, 2) + p(i, 3)]$$

$$= 2 + (7 + 9 + 8 + 9) + \min [10 + 9, 11 + 5, 8 + 10, 2 + 1]$$

$$= 2 + 33 + \min [19, 16, 18, 3]$$

$$= 2 + 33 + 3$$

$$= 38$$

$$g_2 = t(J_1, 2) + \sum_{i \in J_1'} p(i, 2) + \min_{i \in J_1'} p(i, 3)$$

$$= 7 + (10 + 11 + 8 + 2) + \min [9, 5, 10, 1]$$

$$= 7 + 31 + 1 = 39$$

$$g_3 = t(J_1, 3) + \sum_{i \in J_1'} p(i, 3)$$

$$= 20 + (9 + 5 + 10 + 1)$$

$$= 20 + 25 = 45$$

Step 2 —

$$g = \max [g_1, g_2, g_3]$$

$$= \max [38, 39, 45] = 45$$

Step 3 —

$$I_1 = \max [t(J_1, 2) - t(J_1, 3) + p(1, 3), 0]$$

$$= \max [7 - 20 + 13, 0]$$

$$= \max [0, 0] = 0.$$

Step 4 —

$$G = g + \max [I_1 - g + g_3, 0]$$

$= 45 + \max [0 - 45 + 45, 0]$

$= 45 + 0 = 45.$

Applying the above steps for J_1 = (2), (3), (4) and (5), evaluations of relevant lower bounds are given as in Table III.

TABLE III:

J_1	$(J_1, 2)$	[illegible]	[illegible]	g_1	g_2	g_3	g	I_1	G
1	2	7	20	38	39	45	45	0	45
2	7	17	26	38	44	55	55	0	55
3	9	20	25	38	46	58	58	0	58
4	8	16	26	38	45	54	54	0	54
5	9	11	12	51	50	49	51	0	51

Step 5 — $G = \min [45, 55, 58, 54, 51] = 45$

$\therefore J_1 = 1$ is the branching node.

Now take the partial schedule $J_r = J_2$ consisting of two jobs as J_2 = (12), (13), (14) and (15) the relevant lower bounds evaluations are given as in Table IV.

TABLE IV:

J_2	$(J_2, 2)$	[illegible]	[illegible]	g_1	g_2	g_3	g	I_2	G
12	9	19	29	38	41	45	45	2	47
13	11	22	27	38	43	47	47	2	49
14	10	18	30	38	42	45	45	3	48
15	11	13	21	51	47	45	51	4	51

Minimum value of lower bound G is 47 for J_2 = 12.

$\therefore J_2 = 12$ is the branching node.

Continuing in this way, we have the scheduling tree as in Figure 1.

Hence, J_5 = (12435) is the optimal sequence and the minimum total elapsed time subject to zero total idle time of machines is 47 hours. Total idle time of machines is zero if machine 1 is hired in the starting, machine after 4 hours from the starting and machine 3 after 9 hours from the starting.

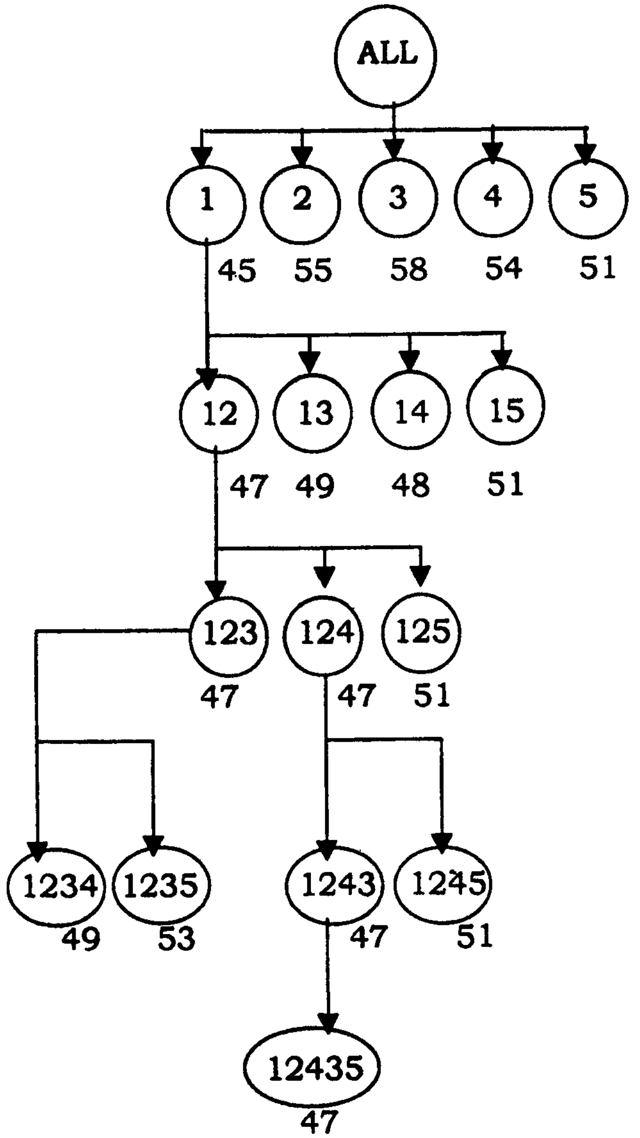

FIG. 1

ACKNOWLEDGEMENT

The authors are extremely grateful to the referee for his valuable comments and suggestions.

REFERENCES

1. P. C. Bagga, *Canad. OR Soc.* **7** (1969) 152-53.
2. P. C. Bagga and Ambika Bhambani, *Indian Assoc. Productivity, Quality and Reliability*, **21** (1996) 73-77.
3. S. M. Johnson, *Nav. Res. Log. Q.*, **1** (1954) 61-68.
4. Z. A. Lomnicki, *Opnl. Res. Q.*, **16** (1965) 89-100.