Journal of the Operations Research
Society of Japan
Vol.21, No.2, June, 1978

AN ADJACENT PAIRWISE APPROACH TO THE MEAN FLOW-TIME SCHEDULING PROBLEM.

Shigeji Miyazaki, Noriyuki Nishiyama and Fumio Hashimoto
University of Osaka Prefecture

(Received December 10, 1977; Revised February 22, 1978)

Abstract In this paper, sufficient conditions to decide the precedence relation between neighboring two jobs are presented by means of an adjacent pairwise interchage method for minimizing mean flow-time in flow-shop scheduling. On the bases of the sufficient conditions, a computational algorithm is proposed for an optimal or near optimal solution. The mean flow-time by this algorithm puts 90% of the optimal value as an average of over one hundred problems. The algorithm can be executed even by manual calculations within the time proportional to nxm^2, where n and m are the number of jobs and machines respectively.

1. Introduction

Ever so much research [1∿9] has been devoted to flow-shop scheduling, yet relatively few results exist for performance measures other than maximal flow-time. For instance, Nabeshima [5] presented an algorithm based on the sufficient conditions to minimize maximal flow-time in flow-shop scheduling where no passing is allowed. The same approach, however, has not been applied to the mean flow-time problem, which is as significant a performance measure as the maximal flow-time.

In this paper, the sufficient conditions are given to decide the precedence relation between neighboring two jobs to minimize the mean flow-time in flow-shop scheduling problem. An algorithm based on the sufficient conditions is also presented for an optimal or near optimal solution.

The computational experience shows that the approximation ratio between obtained solutions and the optimal ones indicates 90 % as an average of over one hundred problems. Moreover, it shows that the algorithm can be executed even by manual calculations within the time proportional to (the number of jobs) × (the number of machines)^2.

2. Definition of the Model

The discussed model can be defined as follows:

1) Let n be the number of jobs to be processed, and ith job in an arbitrary sequence S is denoted by J_i where $i=1,2,\cdots,n$. All these jobs are available for processing at time zero.

2) The manufacturing system consists of m different machines which are numbered according to the order of production stage. Let M_j be the jth machine in the system where $j=1, 2,\cdots,m$. Every machine is continuously available.

3) Every job is completed through the same production stage that is $M_1 \rightarrow M_2 \rightarrow, \cdots, \rightarrow M_m$.

4) Let $p_{i,j}$ denote the processing time of J_i on M_j. Setup times for operations are sequence-independent and are included in processing times. Handling times are assumed to be so limited that they can be neglected.

5) The same job sequence occurs on each machine; in other words, no passing is allowed in the shop.

6) The other conditions on usual flow-shop problem are also assumed.

3. Formulation of Mean Flow-Time

Let $T_j(i)$ denote the partial flow-time of J_i counted from the completion

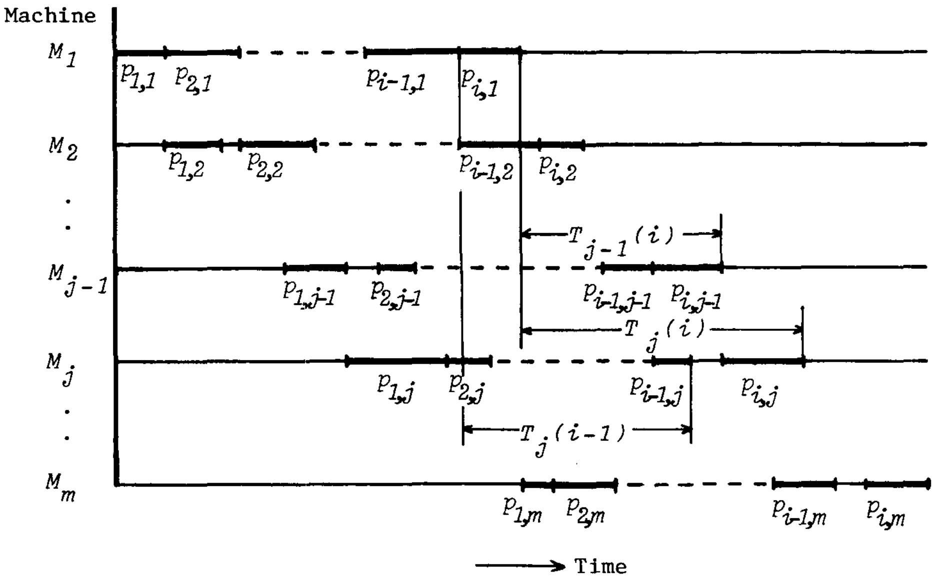

Fig. 1 Definition of $T_j(i)$.

time of J_i on M_1 to that of J_i on M_j, referring to Fig. 1. There exist recurrence relations (3.1) on $T_j(i)$, because the starting time of J_i on M_j should not be earlier than both the completion time of J_i on M_{j-1} and the completion time of J_{i-1} on M_j.

$$(3.1)\qquad T_j(i)=p_{i,j}+max\{T_j(i-1)-p_{i,1},\ T_{j-1}(i)\},\ i=1,2,\cdots,n\ ;\ j=2,3,\cdots m,$$

where $T_j(0)\equiv 0$, $T_1(i)\equiv 0$.

Let F_i be the flow-time of J_i, that is, the whole elapsed time of J_i counted from the starting time of J_1 on M_1, to the completion time of J_i on M_m. Then F_i is given by:

$$(3.2)\qquad F_i=\sum_{t=1}^{i}p_{t,1}+T_m(i),\quad i=1,2,\cdots,n.$$

Mean flow-time $\bar{F}$ for n different jobs processed on m machines is:

$$(3.3)\qquad \bar{F}=\frac{1}{n}\sum_{i=1}^{n}F_i=\frac{1}{n}\sum_{i=1}^{n}\{\sum_{t=1}^{i}p_{t,1}+T_m(i)\}.$$

Therefore we have, from (3.3),

$$(3.4)\qquad n\bar{F}=\sum_{i=1}^{n}F_i=\sum_{i=1}^{n}\{\sum_{t=1}^{i}p_{t,1}+T_m(i)\},$$

where $n\bar{F}$ expresses the total flow-time of n jobs, $n\bar{F}$ shall be used in place of $\bar{F}$ in the further analysis, since n is a constant independent of a sequence.

In the sequence S, let s be a subsequence consisting of the first $q-1$ jobs, that is, $J_1,J_2,\cdots,J_{q-1}$, and in succession to s, J_q and J_{q+1} (these two jobs are called adjacent two jobs hereafter) are assumed to be processed in the order $J_q\ J_{q+1}$. Now consider the sequence S' in which J_q and J_{q+1} are pairwise interchanged and are processed in the order $J_{q+1}\ J_q$. The sequence is the same for the first $q-1$ jobs and the last $(n-q-1)$ jobs under either S or S' as illustrated in Fig. 2.

In order to distinguish the notation of partial flow-time under S from S', let $T_m(q)$, $T_m(q,q+1)$, and $T_m(i)_S$ $(i=q+2,q+3,\cdots,n)$ denote the partial flow-time of J_q, J_{q+1}, and J_i $(i=q+2,q+3,\cdots,n)$ under S in turn, and let $T_m(q+1)$, $T_m(q+1,q)$, and $T_m(i)_{S'}$ $(i=q+2,q+3,\cdots,n)$ denote the partial flow-time of J_{q+1}, J_q, and J_i $(i=q+2,q+3,\cdots,n)$ under S' in turn, moreover let $\bar{F}'$ be the mean flow-time under S'. Then the total flow-time under S and S' are formulized by (3.5) and (3.6) respectively, of which terms are divided into four parts that

$$S = J_1,\ J_2,\ \cdots\cdots,\ J_{q-1},\ J_q,\ J_{q+1},\ \cdots\cdots,\ J_n$$

$$S' = \underbrace{J_1,\ J_2,\ \cdots\cdots,\ J_{q-1},}_{\text{Partial Sequence } s}\ J_{q+1},\ J_q,\ \cdots\cdots,\ J_n$$

Fig. 2 Sequence S and S'.

are the summation of the flow-times over all jobs belonging to s, the flow-time of J_q, the flow-time of J_{q+1}, and the summation of the flow-times over all jobs following J_{q+1}.

$$(3.5)\qquad n\bar{F} = \sum_{i=1}^{q-1}\{\sum_{t=1}^{i} p_{t,1} + T_m(i)\} + \sum_{t=1}^{q-1} p_{t,1} + p_{q,1} + T_m(q) + \sum_{t=1}^{q+1} p_{t,1} + T_m(q,q+1) + \sum_{i=q+2}^{n}\{\sum_{t=1}^{i} p_{t,1} + T_m(i)_S\}.$$

$$(3.6)\qquad n\bar{F}' = \sum_{i=1}^{q-1}\{\sum_{t=1}^{i} p_{t,1} + T_m(i)\} + \sum_{t=1}^{q-1} p_{t,1} + p_{q+1,1} + T_m(q+1) + \sum_{t=1}^{q+1} p_{t,1} + T_m(q+1,q) + \sum_{i=q+2}^{n}\{\sum_{t=1}^{i} p_{t,1} + T_m(i)_{S'}\}.$$

Eliminating the common terms between (3.5) and (3.6) from the each equation, and denoting the remaining, $\langle n\bar{F}\rangle$ and $\langle n\bar{F}'\rangle$, respectively, we have:

$$(3.7)\qquad \langle nF\rangle = p_{q,1} + T_m(q) + T_m(q,q+1) + \sum_{i=q+2}^{n} T_m(i)_S\ ,$$

and

$$(3.8)\qquad \langle n\bar{F}'\rangle = p_{q+1,1} + T_m(q+1) + T_m(q+1,q) + \sum_{i=q+2}^{n} T_m(i)_{S'}\ .$$

4. The Order of Adjacent Two Jobs

If,

$$(4.1)\qquad \langle n\bar{F}\rangle \leqq \langle n\bar{F}'\rangle$$

that is:

$$(4.2)\qquad n\bar{F} \leqq n\bar{F}'$$

holds, J_{q+1} cannot directly precede J_q in the optimal sequence, because (3.7) and (3.8) exist independent of adjacent two jobs position in a sequence. Either of J_q and J_{q+1} can optimally precede the other, provided there exists equality in (4.1). Therefore, we shall investigate the sufficient conditions to satisfy (4.1) in the following:

Comparing each term of (3.7) with the corresponding term of (3.8), we have:

$$p_{q,1} + T_m(q) \leqq p_{q+1,1} + T_m(q+1) , \tag{4.3}$$

$$T_m(q,q+1) \leqq T_m(q+1,q) , \tag{4.4}$$

and

$$\sum_{i=q+2}^{n} T_m(i)_S \leqq \sum_{i=q+2}^{n} T_m(i)_{S'} , \tag{4.5}$$

which are to be sufficient conditions of (4.1).

$T_m(q)$ and $T_m(q+1)$ in (4.3) are, from (3.1), given by:

$$T_m(q) = p_{q,m} + max\{ T_m(q-1) - p_{q,1}, T_{m-1}(q)\} , \tag{4.6}$$

and

$$T_m(q+1) = p_{q+1,m} + max\{ T_m(q-1) - p_{q+1,1}, T_{m-1}(q+1)\} . \tag{4.7}$$

Working out recurrence relations (4.6) and (4.7), we have:

$$T_m(q) = \max_{r=1\sim m}\{ T_{m-r+1}(q-1) + \sum_{t=1}^{r} p_{q,m-t+1}\} - p_{q,1} , \tag{4.8}$$

and

$$T_m(q+1) = \max_{r=1\sim m}\{ T_{m-r+1}(q-1) + \sum_{t=1}^{r} p_{q+1,m-t+1}\} - p_{q+1,1} . \tag{4.9}$$

Substituted into (4.3), (4.8) and (4.9) give

$$\max_{r=1\sim m}\{ T_{m-r+1}(q-1) + \sum_{t=1}^{r} p_{q,m-t+1}\} \leqq \max_{r=1\sim m}\{ T_{m-r+1}(q-1) + \sum_{t=1}^{r} p_{q+1,m-t+1}\}. \tag{4.10}$$

Since $T_{m-r+1}(q-1)$ is common between both sides of (4.10), the comparison between each relative term of (4.10) gives the next m different inequalities:

$$\sum_{t=1}^{r} p_{q,m-t+1} \leqq \sum_{t=1}^{r} p_{q+1,m-t+1}, \quad (r=1,2,\cdots,m) , \tag{4.11}$$

which are to be sufficient conditions of (4.10) that is equivalent to (4.3), as completely proved at the following theorem.

Now, we shall investigate the sufficient conditions of (4.4) and (4.5).

The partal flow-time of J_{q+2} under S is given as similar to (4.8),

$$(4.12)\qquad T_j(q+2)_S = \max_{r=1\sim j}\{T_{j-r+1}(q,q+1) + \sum_{t=1}^{r} p_{q+2,j-t+1}\} - p_{q+2,1},$$

which is the nondecreasing function of each j terms $T_k(q,q+1)$ $(k=1,2,\cdots,j)$.

The partial flow-time of J_{q+2} under S' is also given as similar to (4.9),

$$(4.13)\qquad T_j(q+2)_{S'} = \max_{r=1\sim j}\{T_{j-r+1}(q+1,q) + \sum_{t=1}^{r} p_{q+2,j-t+1}\} - p_{q+2,1},$$

which is the nondecreasing function of each j terms $T_k(q+1,q)$ $(k=1,2,\cdots,j)$.

So that, if:

$$(4.14)\qquad T_j(q,q+1) \leqq T_j(q+1,q), \quad (j=1,2,\cdots,m),$$

then

$$(4.15)\qquad T_j(q+2)_S \leqq T_j(q+2)_{S'}, \quad (j=1,2,\cdots,m).$$

Moreover, under (4.15), $T_j(q+3)_S \leqq T_j(q+3)_{S'}$ holds for every j $(j=1,2,\cdots,m)$ from (3.1), and similar inequalities as (4.15) hold for $J_{q+4},\cdots,J_n$ in turn.

Therefore, (4.14) should be sufficient conditions of (4.5), by the reason that (4.5) is concerned with the summations of the partial flow-time of J_{q+2}, $J_{q+3},\cdots,J_n$ on M_m. Inequality (4.4) is contained in (4.14), since (4.14) in the case of $j=m$ corresponds to (4.4).

Nabeshima [5] proved that the next $m(m-1)/2$ inequalities:

$$(4.16)\qquad \min(p_{q,j}, p_{q+1,j+1}) \leqq \min(p_{q+1,j}, p_{q,j+1}), \quad (j=1,2,\cdots,m-1),$$

and

$$(4.17)\quad \min\left(\sum_{j=u}^{v} p_{q,j}, \sum_{j=u+1}^{v+1} p_{q+1,j}\right) \leqq \min\left(\sum_{j=u}^{v} p_{q+1,j}, \sum_{j=u+1}^{v+1} p_{q,j}\right), \quad (1 \leqq u<v \leqq m-1)$$

are the sufficient conditions of (4.14).

The discussion above should lead to the following theorem:

Theorem. If $m(m+1)/2$ inequalities:

$$(4.11)\qquad \sum_{t=1}^{r} p_{q,m-t+1} \leqq \sum_{t=1}^{r} p_{q+1,m-t+1}, \quad (r=1,2,\cdots,m),$$

(4.16) $min(p_{q,j}, p_{q+1,j+1}) \leqq min(p_{q+1,j}, p_{q,j+1})$, ($j=1,2,\cdots,m-1$) ,

and

(4.17) $min(\sum_{j=u}^{v} p_{q,j}, \sum_{j=u+1}^{v+1} p_{q+1,j}) \leqq min(\sum_{j=u}^{v} p_{q+1,j}, \sum_{j=u+1}^{v+1} p_{q,j})$, ($1 \leqq u < v \leqq m-1$),

hold, then J_{q+1} cannot directly precede J_q to minimize the mean-flow time in flow-shop scheduling where no passing is allowed. If equality signs hold in all of (4.11), (4.16), and (4.17), each of J_q and J_{q+1} may directly precede the other.

Proof: The demonstration here may be restricted to the statement that (4.11) is sufficient condition of (4.3) for any number of machines m over two, since Nabeshima [5] has proved that (4.16) and (4.17) are the sufficient conditions of (4.4) and (4.5). The theorem can be demonstrated by mathematical induction as follows:

Inequality (4.3) for $m=2$ is

$$p_{q,1} + T_2(q) \leqq p_{q+1,1} + T_2(q+1),$$

which is, from(3.1), rewritten by :

(4.18)
$$max\{T_2(q-1) + p_{q,2}, T_1(q-1) + p_{q,2} + p_{q,1}\}$$
$$\leqq max\{T_2(q-1) + p_{q+1,2}, T_1(q-1) + p_{q+1,2} + p_{q+1,1}\}.$$

Sufficient conditions of (4.18) are simply given by:

$$p_{q,2} \leqq p_{q+1,2},$$

and

$$p_{q,2} + p_{q,1} \leqq p_{q+1,2} + p_{q+1,1}.$$

These inequalities coincide with (4.11) for $m=2$, consequently (4.11) should be the sufficient conditions of (4.3) for $m=2$.

Now suppose that (4.11) is the sufficient condition of (4.3) for $m=K$, where K is an arbitrary integer greater or equal than 2. This assumption may be rewritten by the following statement that

(4.19)
$$\sum_{t=1}^{r} p_{q,K-t+1} \leqq \sum_{t=1}^{r} p_{q+1,K-t+1}, \quad (r=1,2,\cdots,K)$$

are the sufficient conditions of

(4.20)
$$p_{q,1} + T_K(q) \leqq p_{q+1,1} + T_K(q+1).$$

For $m=K+1$, the partial flow-time under S and S', $T_{K+1}(q)$ and $T_{K+1}(q+1)$, are given as established in (3.1):

$$T_{K+1}(q) = p_{q,K+1} + max\{T_{K+1}(q-1) - p_{q,1}, T_K(q)\}, \tag{4.21}$$

and

$$T_{K+1}(q+1) = p_{q+1,K+1} + max\{T_{K+1}(q-1) - p_{q+1,1}, T_K(q+1)\}. \tag{4.22}$$

Therefore, each side of (4.3) for $m=K+1$ is respectively:

$$p_{q,1} + T_{K+1}(q) = max\{p_{q,K+1} + T_{K+1}(q-1), p_{q,1} + p_{q,K+1} + T_K(q)\}, \tag{4.23}$$

and

$$p_{q+1,1} + T_{K+1}(q+1) = max\{p_{q+1,K+1} + T_{K+1}(q-1), p_{q+1,1} + p_{q+1,K+1} + T_K(q+1)\}. \tag{4.24}$$

From (4.23) and (4.24), the sufficient conditions of

$$p_{q,1} + T_{K+1}(q) \leqq p_{q+1,1} + T_{K+1}(q+1)$$

should be given by:

$$p_{q,K+1} \leqq p_{q+1,K+1}, \tag{4.25}$$

and

$$p_{q,1} + p_{q,K+1} + T_K(q) \leqq p_{q+1,1} + p_{q+1,K+1} + T_K(q+1). \tag{4.26}$$

Since, from the preliminary assumption for $m=K$, the sufficient conditions of

$$p_{q,1} + T_K(q) \leqq p_{q+1,1} + T_K(q+1) \tag{4.20}$$

are (4.19), and each side of (4.20) is the first and third term of corresponding side of (4.26), so that:

$$\sum_{t=1}^{r} p_{q,K-t+1} + p_{q,K+1} \leqq \sum_{t=1}^{r} p_{q+1,K-t+1} + p_{q+1,K+1}, \quad (r=1,2,\cdots,K). \tag{4.27}$$

should be the sufficient conditions of (4.26).

Rearranging (4.25) and (4.27), we have

$$\sum_{t=1}^{r} p_{q,K-t+2} \leqq \sum_{t=1}^{r} p_{q+1,K-t+2}, \quad (r=1,2,\cdots,K+1). \tag{4.28}$$

These inequalities (4.28) coincide with (4.11) for $m=K+1$. Therefore, (4.11) should be sufficient condition of (4.3) for $m=K+1$, under the assumption that (4.11) is the sufficient condition of (4.3) for $m=K$.

These statements above prove that (4.11) is the sufficient condition of (4.3) for any integer m greater or equal than 2.

5. Algorithm

We now propose an algorithm to find an optimal or near optimal solution that minimizes the mean flow-time using above theorem, since the theorem holds for any positions of the adjacent two jobs in the sequence. The algorithm will be explained by solving an example problem composed of four jobs and four machines. The processing times of each operation in the example are listed in Table 1.

Step 1. Decide m kinds of temporary sequences which can lead from (4.11) as follows: compile a table of which the first, second and last mth row value correspond to $p_{i,m}$, $p_{i,m}+p_{i,m-1}$, and $p_{i,m}+\cdots+p_{i,1}$ in turn, as tabulated in Table 2. Make m kinds of temporary sequences consisting of n jobs in accordance with the nondecreasing order of each row value in Table 2. Assign an integer between 1 and n to each

Table 1. Processing Times.

j	$p_{i,j}$	J_1	J_2	J_3	J_4
1	$p_{i,1}$	5	5	3	6
2	$p_{i,2}$	7	6	5	5
3	$p_{i,3}$	3	5	5	7
4	$p_{i,4}$	4	3	6	5

Table 2. Sum of Processing Times.

r	$\sum p_{i,j}$	J_1	J_2	J_3	J_4
1	$p_{i,4}$	4	3	6	5
2	$p_{i,4}+p_{i,3}$	7	8	11	12
3	$p_{i,4}+\cdots+p_{i,2}$	14	14	16	17
4	$p_{i,4}+\cdots+p_{i,1}$	19	19	19	23

Table 3. Ordinal Numbers by Step 1.

r	J_1	J_2	J_3	J_4
1	2	1	4	3
2	1	2	3	4
3	1	1	3	4
4	1	1	1	4

Table 4. Ordinal Numbers by Step 2.

j	J_1	J_2	J_3	J_4
1	2	2	1	4
2	4	3	1	1
3	1	4	2	3

Table 5. Ordinal Numbers by Step 3.

u	v	J_1	J_2	J_3	J_4
1	2	4	2	1	2
	3	4	3	1	2
2	3	3	3	1	2

job according to its ordinal number in the temporary sequence as shown in Table 3. In case more than two jobs have the same value in a row, assign the same integers to them. An example of this can be seen in the third and fourth rows of Table 3.

Step 2. Make $m-1$ kinds of temporary sequences, in terms of applying Johnson's Rule [4] to (4.16) for each value of $j=1,2,\cdots,m-1$. Assign an integer between 1 and n to each job according to its ordinal number in this temporary sequence as shown in Table 4. Assign the same integers to the jobs which can occupy the same position in the temporary sequence by this step.

Step 3. Apply Johnson's Rule to (4.17), regarding the serial machines from M_u to M_v as the first machine and the serial machines from M_{u+1} to M_{v+1} as the second machine. Total number of temporary sequences made by this step becomes $(m-1)(m-2)/2$ which coincides with the total number of combinations between u and v under the restrictions on their range shown in (4.17). Assign an integer between 1 and n to each job according to its ordinal number in this temporary sequence as shown in Table 5.

Step 4. Calculate the sum of integers assigned to each job in the Step 1, 2, and 3 as shown in Table 6. Arrange each job in the nondecreasing order of the total integers. Break a tie by placing jobs with lower original numbers first.

Table 6. Sum of Ordinal Numbers.

Job	J_1	J_2	J_3	J_4
Sum of Ordinal Numbers	23	22	18	29

The solution for this example becomes $J_3 - J_2 - J_1 - J_4$, which is the optimum. In case all of the temporary sequences are equal, the solution inevitably becomes the optimum.

6. Efficiency of the Algorithm

In order to verify the efficiency of the algorithm, 16 examples tabulated in Table 7 were solved. Obtained solutions were appraised by the approximation ratio defined by (6.1), and the results are summarized in Table 8.

$$\eta = 100 \times (w-a)/(w-o) \quad (\%) \tag{6.1}$$

Where η is the approximation ratio, w, a, and o are the maximal (worst), the obtained, and the optimal value of performance measure respectively. The

Table 7. Solved Flow-Shop Problems.

m \ n	3	4	5	6	
3	P_1	P_2	P_3	P_4	P : Standard deviation of processing times (Uniform random integers with mean 5)
4	P_2	P_1	P_4	P_3	$P_1=0.8$, $P_2=1.4$,
5	P_3	P_4	P_2	P_1	$P_3=2.0$, $P_4=2.6$.
6	P_4	P_3	P_1	P_2	

Table 8. Approximation Ratio.

m \ n	3	4	5	6
3	83.3	92.9	81.8	84.8
4	100.0	85.7	96.6	98.4
5	100.0	98.5	79.3	89.7
6	100.0	88.2	73.3	97.9

(%)

Table 9. The Results of Analysis of Variance, $F(3, 6 : 0.05) = 4.76$.

Factor	S	Φ	V	F_0
n	375.4	3	125.1	3.58
m	188.5	3	62.8	1.80
P	331.4	3	110.5	3.17
E	209.5	6	34.9	
Sum	1104.8	15		

Table 10. The Required Times for Solutions by Hand.

m \ n	3	4	5	6
3	1.5	3.5	4	5
4	6	7	7.5	11.5
5	9	9.3	12.5	18
6	9	15	19.5	24

(minutes)

approxomation ratio defined by (6.1) contains the maximal and the optimal value of performance measure so as to reach 0% in case the obtained solution coincides with the maximal one, and 100% in case the obtained solution coincides. with the optimal one.

The average of ratios in Table 8 is 90.7% and the analysis of variance shown in Table 9 indicates that none of the three factors, the number of jobs, the number of machines, and the standard deviation of processing times affect the approximation ratio.

Another 90 different examples which have 2 to 6 jobs and 2 to 6 machines were solved, having shown 90.2% for the average approximation ratio. The algorithm may be executed even by hand calculations which need the time proportional to $n \times m^2$, since the time is subject to the number of jobs and the number of inequalities. Table 10 is an example of the executed time by manual calculations.

The step 4 of the proposed algorithm can be replaced by the similar procedure as the step 2, 3, and 4 in Nabeshima's. Although the replaced algorithm may generate as high an approximation ratio as the proposed one, it takes two or three times as long to ececute the replaced one than to execute the original one.

7. Conclusions

The summary of the results is as follows:

1) The sufficient conditions were presented to minimize mean flow-time in flow-shop scheduling where no passing is allowed. It was proved by mathematical induction that the conditions should exist for any number of machines.

2) The algorithm based on the sufficient conditions is proposed for an optimal or near optimal solution.

3) More than one hundred examples were solved by the algorithm, having shown 90% for the approximation ratio on an average. None of the three factors, the number of jobs, the number of machines, and the standard deviation of processing times affected the value of approximation ratio.

4) The algorithm may be executed even by hand calculations which need the time proportional to $n \times m^2$.

Acknowledgement

The authors wish to thank Mr. Hirao Fujikawa for his help in deriving the equations and in achieving the numerical examples throughout this study.

References

[1] Gupta, J. N. D.: Optimal Flowshop Scheduling with Due Dates and Penalty Costs. *Journal of the Operations Research Society of Japan*, Vol. 14, No. 3(1971), 35-46.

[2] Gupta, J. N. D.: Heuristic Algorithms for Multistage Flowshop Scheduling Problem. *AIIE Transactions*, Vol. 4, No. 1(1972), 11-18.

[3] Ignall, E. and Schrage, L.: Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems. *Operations Research*, Vol. 13, No. 3(1965), 400-412.

[4] Johnson, S. M.: Optimal Two- and Three-Stage Production Schedules with Setup Times Included. *Naval Research Logistics Quarterly*, Vol. 1, No. 1 (1954), 61-68.

[5] Nabeshima, I.: The Order of n Items Processed on m Machines [III]. *Journal of the Operations Research Society of Japan*, Vol. 16, No. 3(1973), 163-185.

[6] Panwalkar, S. S. and Khan, A. W.: An Ordered Flow-Shop Sequencing Problem with Mean Completion Time Criterion. *International Journal of Production Research*, Vol. 14, No.5(1976), 631-635.

[7] Smith, M. L., Panwalkar, S. S. and Dudek, R. A.: Flow-Shop Sequencing Problem with Ordered Processing Time Matrices: A General Case. *Naval Research Logistics Quarterly,* Vol. 23, No. 3(1976), 481-486.

[8] Szwarc. W.: Optimal Elimination Methods in the $m \times n$ Flow-Shop Scheduling Problem. *Operations Research,* Vol. 21, No. 6(1973), 1250-1259.

[9] Yueh Ming-I : On the n Job, m Machine Sequencing Problem of Flow-Shop. *OR'75 : Proceedings of the Seventh International Conference on Operational Research (ed. K. B. Haley).* North-Holland Publishing Company, Amsterdam, 1976, 179-200.

Shigeji MIYAZAKI: Department of Industrial Engineering, College of Engineering, University of Osaka Prefecture, Mozu-Ume-Machi, Sakai, Osaka, 591, Japan.