THE ORDER OF \( n \) ITEMS PROCESSED ON \\
\( m \) MACHINES [III]

ICHIRO NABESHIMA

University of Electro-Communications

(Received September 18, 1972)

Abstract

Sufficient conditions are presented for determining the definite order of adjacent two jobs (items) for min-makespan problem in flow shop where no passing is allowed, by using dynamic programming formulation. In relation to them, their applicabilities to the algorithms for obtaining the optimal sequence are mentioned.

Finally algorithms for obtaining the approximate sequence are presented together with numerical example. It is obtained satisfying approximate sequence.

1. Introduction

Recently many branch-and-bound type algorithms [1]-[5] for obtaining the optimal schedule for min-makespan problem in job shop have been presented. On the other hand, there exist special types of algorithms [6]-[12] for this problem in flow shop for each case according to whether passing is allowed or not, based on the structure of the flow shop. Especially the problem for three machines case in flow shop can be assumed that no passing is allowed which
makes the analysis easier.

This paper presents certain analytic devices which may be efficient to use together with those algorithms in such flow shop.

That is the set of sufficient conditions for determining the definite order of adjacent two jobs regardless of their position in optimal sequence, which is derived by dynamic programming formulation. First sufficient conditions for three machines case are presented and its generalizations to \( m \) machines case follow. Finally efficient algorithms for obtaining the approximate sequence for the problem are presented together with numerical example.

2. Elapsed Times \( T_k(i_q) \)

Let \( \omega = i_1i_2 \ldots i_{n-1}i_n \) be any sequence of \( n \) jobs and \( T_k(i_q) \) be the elapsed time of job \( i_q \) counted from the completion time of this job on first machine \( M_1 \) to that of this job on \( k \)th machine \( M_k \) (\( q = 1 \sim n, \ k = 2 \sim m \)), that is, \( T_k(i_q) \) is the time committed ahead on \( M_k \) to \( M_1 \) by the processing of job \( i_q \). Then total elapsed time (makespan) \( TE(\omega) \) is expressed by

\[
    TE(\omega) = \sum_{q=1}^{n} p_{i_q,1} + T_m(i_n)
\]

where \( p_{i,j} \) means the processing time of job \( i \) on machine \( M_j \).

Since the first term of (2.1) is constant for any sequence \( \omega \), hence, in order to obtain the optimal sequence, it is sufficient to seek for the sequence which has minimum \( T_m(i_n) \) among all sequences.

On the other hand, next recurrence relations hold for \( T_k(i_q) \):

\[
    T_k(i_q) = p_{i_q,k} + \max[T_k(i_{q-1}) - p_{i_q,1}, \ T_{k-1}(i_q)]
\]

\( (q = 1 \sim n, \ k = 2 \sim m) \)

where \( T_k(i_0) = 0, \ T_1(i_0) = 0 \), by the reason that the starting time of the processing of job \( i_q \) on \( M_k \) is not sooner than both completion time of job \( i_{q-1} \) on \( M_k \) and completion time of job \( i_q \) on \( M_{k-1} \) under taking the origin of time scale as the completion time of job \( i_q \) on \( M_1 \).
From these recurrence relations, we see that each $T_k(i_q)$ is an increasing function of $T_k(i_{q-1})$ and $T_{k-1}(i_q)$ ($q=1 \sim n$, $k=2 \sim m$).

Also, by using these relations successively, we can compute total elapsed time (makespan) of any sequence.

3. Order of Adjacent Two Jobs in Optimal Sequence

When optimal scheduling procedure is employed and after the processing of certain definite subsequence $S$ with last job $l$, let two jobs $i, j$ be processed after $S$.

If two jobs $i, j$ are processed in this order after $S$, then next recurrence relations on $T_k(i)$ and $T_k(ij) = T_k(j)$ hold from (2.2) where notation $T_k(ij)$ will be used to show the order of $i, j$ and in general $T_k(S)$ means $T_k(l)$ of last job $l$ of the sequence $S$:

\begin{align*}
(3.1) \quad T_k(i) &= p_{i,k} + \max \{ T_k(l) - p_{i,1}, T_{k-1}(i) \} \\
(3.2) \quad T_k(ij) &= p_{i,k} + \max \{ T_k(i) - p_{i,1}, T_{k-1}(ij) \} \\
& \quad (k=2 \sim m)
\end{align*}

where $T_i(i) = 0$, $T_i(ij) = 0$ and $T_k(l) = 0$ when $i$ is a first job.

If we interchange the order of $i$ and $j$, then recurrence relations which can be easily obtained by exchanging $i$ and $j$ in (3.1) and (3.2) hold.

Then, since $T_k(i)$, $T_k(ij)$ say is an increasing function of $T_k(l)$ and $T_{k-1}(i)$, $T_k(i)$ and $T_{k-1}(ij)$ respectively ($k=2 \sim m$) and optimal sequence has minimum $T_m(i_n)$ among all sequences with last job $i_n$, if $(m-1)$ inequalities $T_k(ij) \leq T_k(ji)$ ($k=2 \sim m$) hold, next theorem obviously holds: (cf. Theorem 1 in [17])

Theorem 1. When optimal scheduling procedure is employed and after the processing of definite subsequence $S$ with last job $l$, let two jobs $i, j$ be processed after $S$. Then if $(m-1)$ inequalities

\begin{align*}
(3.3) \quad T_k(ij) &\leq T_k(ji) \quad (k=2 \sim m)
\end{align*}

hold, we can regard that job $i$ precedes job $j$ after $S$ in order to construct the optimal sequence.
4. Constant Order of Definite Adjacent Two Jobs for Three Machines Case

In the following sections 4 and 5, they will be shown sufficient conditions followed from theorem 1 to decide the constant order of any two definite adjacent jobs regardless of their position in order to construct the optimal sequence for $m$ machines ($m \geq 2$) where they coincide with Johnson's criterion for two machines case ($m = 2$) [15].

In order to make clear the content and process of the demonstration of these sufficient conditions, first we shall treat three machines case ($m = 3$) in this section.

4.1 Proposed Sufficient Conditions

By using the recurrence relations (3.1), (3.2) for $k = 2, 3$ and putting $T_3(i)$ into $T_3(ij)$ and $T_3(i)$, we have

\begin{align*}
T_3(ij) &= p_{i,3} + p_{i,2} - p_{i,1} - p_{j,1} + \max \left[ T_3(l), p_{i,1}, p_{i,1} + p_{j,1} - p_{i,3} \right] \\
&= p_{i,3} - p_{i,1} - p_{j,1} + \max \left[ T_3(l) + p_{i,2} - p_{i,3}, p_{i,1} + p_{j,1} - p_{i,3} \right] \\
&= p_{i,2} + p_{j,2} - p_{i,3} + p_{i,1} + p_{j,1} - p_{i,3}
\end{align*}

(4.1)

\begin{align*}
T_3(i) &= p_{i,3} - p_{i,1} + \max \left[ T_3(l), T_2(l) + p_{i,2}, T_2(l) + p_{j,1} + p_{j,2} - p_{i,3}, T_2(l) + p_{j,1} + p_{j,2} - p_{i,3}, p_{i,1} + p_{j,1} + p_{j,2} - p_{i,3} \right] \\
&= p_{i,2} + p_{j,2} - p_{i,3} + p_{i,1} + p_{j,1} + p_{j,2} - p_{i,3}
\end{align*}

(4.2)

Then, by putting (4.2) and (4.3) into $T_3(ij)$, we have

\begin{align*}
T_3(ij) &= p_{i,3} - p_{i,1} - p_{j,1} + \max \left[ T_3(l), T_3(l) + p_{i,3}, T_3(l) + p_{i,3} - p_{j,1}, T_3(l) + p_{i,3}, p_{i,3} - p_{j,1}, T_3(l) + p_{j,2} - p_{i,3}, p_{j,2} - p_{i,3}, p_{i,3} + p_{j,3} - p_{i,3} \right] \\
&= p_{i,2} + p_{j,2} - p_{i,3} + p_{i,1} + p_{j,1} + p_{j,2} - p_{i,3}
\end{align*}

(4.4)

By the same way we have

\begin{align*}
T_3(ji) &= p_{i,3} + p_{j,3} - p_{j,1} - p_{i,1} + \max \left[ T_3(l), p_{j,1}, p_{j,1} + p_{i,1} - p_{j,3} \right] \\
&= p_{i,3} + p_{j,3} - p_{j,1} - p_{i,1} + \max \left[ T_3(l), T_2(l) + p_{j,2}, T_2(l) + p_{j,2} - p_{j,3}, p_{j,2} + p_{i,1}, p_{j,1} + p_{j,2} - p_{j,3}, p_{j,1} + p_{i,1} + p_{i,2} - p_{j,3} \right] \\
&= p_{i,3} + p_{j,3} - p_{j,1} - p_{i,1}
\end{align*}

(4.5)

Hence, if inequality

\begin{align*}
T_3(ij) &= p_{i,3} - p_{i,1} - p_{j,1} + \max \left[ T_3(l), T_3(l) + p_{i,3}, T_3(l) + p_{i,3} - p_{j,1}, T_3(l) + p_{i,3}, p_{i,3} - p_{j,1}, T_3(l) + p_{j,2} - p_{i,3}, p_{j,2} - p_{i,3}, p_{i,3} + p_{j,3} - p_{i,3} \right] \\
&= p_{i,2} + p_{j,2} - p_{i,3} + p_{i,1} + p_{j,1} + p_{j,2} - p_{i,3}
\end{align*}

(4.6)
The Order of n Items Processed on m Machines [III]

(4.7) \( \max [p_{i,1}, p_{i,1}+p_{j,1}-p_{i,3}] \leq \max [p_{j,1}, p_{j,1}+p_{i,1}-p_{j,3}] \)

holds, inequality \( T_3(ij) \leq T_3(ji) \) follows.

Inequality (4.7) is equivalent to

(4.8) \( \min [p_{i,1}, p_{j,3}] \leq \min [p_{j,1}, p_{i,3}] . \)

Next, it follows from \( T_3(ij) \leq T_3(ji) \)

(4.9) \[
\max \{ T_3(l), \max \{ T_3(l)+p_{i,3}, T_3(l)+p_{i,3}+p_{j,3}-p_{i,3}\}, \\
\max \{ p_{i,3}+p_{i,1}, p_{i,1}+p_{i,3}+p_{j,3}-p_{i,3}, p_{i,1}+p_{j,1}+p_{j,3}-p_{i,3}\}\}
\]

\( \leq \max \{ T_3(l), \max \{ T_3(l)+p_{j,3}, T_3(l)+p_{j,3}+p_{i,3}-p_{j,3}\}, \\
\max \{ p_{j,3}+p_{j,1}, p_{j,1}+p_{j,3}+p_{i,3}-p_{j,3}, p_{j,1}+p_{i,1}+p_{i,3}-p_{j,3}\}\} . \)

So if both inequalities

(4.10) \[
\max \{ T_3(l)+p_{i,3}, T_3(l)+p_{i,3}+p_{j,3}-p_{i,3}\}
\]

\( \leq \max \{ T_3(l)+p_{j,3}, T_3(l)+p_{j,3}+p_{i,3}-p_{j,3}\} \)

and

(4.11) \[
\max \{ p_{i,3}+p_{i,1}, p_{i,1}+p_{i,3}+p_{j,3}-p_{i,3}, p_{i,1}+p_{j,1}+p_{j,3}-p_{i,3}\}
\]

\( \leq \max \{ p_{j,3}+p_{j,1}, p_{j,1}+p_{j,3}+p_{i,3}-p_{j,3}, p_{j,1}+p_{i,1}+p_{i,3}-p_{j,3}\} . \)

hold, then (4.9), that is, \( T_3(ij) \leq T_3(ji) \) follows.

Here, (4.10) follows from inequality

(4.12) \[
\max [p_{i,3}, p_{i,3}+p_{j,3}-p_{i,3}] \leq \max [p_{j,3}, p_{j,3}+p_{i,3}-p_{j,3}] \]

which is equivalent to

(4.13) \[
\min [p_{i,1}+p_{i,3}, p_{i,1}+p_{j,3}, p_{j,3}+p_{j,3}] \leq \min [p_{j,1}+p_{j,3}, p_{j,1}+p_{i,3}, p_{i,3}+p_{i,3}] . \]

And, by subtracting the quantity \( p_{i,1}+p_{j,1}+p_{i,3}+p_{j,3} \) from both sides of (4.11), we have ultimately

(4.13) \[
\min [p_{i,1}+p_{i,3}, p_{i,1}+p_{j,3}, p_{j,3}+p_{j,3}] \leq \min [p_{j,1}+p_{j,3}, p_{j,1}+p_{i,3}, p_{i,3}+p_{i,3}] . \]

Hence next theorem holds from theorem 1:

**Theorem 2.** If three inequalities

(4.8) \[
\min [p_{i,1}, p_{j,3}] \leq \min [p_{j,1}, p_{i,3}] ,
\]

(4.12) \[
\min [p_{i,1}, p_{j,3}] \leq \min [p_{j,1}, p_{i,3}] ,
\]

(4.13) \[
\min [p_{i,1}+p_{i,3}, p_{i,1}+p_{j,3}, p_{j,3}+p_{j,3}] \leq \min [p_{j,1}+p_{j,3}, p_{j,1}+p_{i,3}, p_{i,3}+p_{i,3}] .
\]

hold, then always job \( i \) must precede job \( j \), for adjacent two jobs \( i, j \).
regardless of their position in order to construct the optimal sequence. In these three inequalities, transitive property does not hold for (4.13). The theorem with three inequalities as sufficient conditions, each of them has transitive property, is obtained by using next lemma which will be also applied to similar theorem for general \( m \) machines case in section 5.1.

**Lemma.** If three inequalities

\[
\begin{align*}
\text{(4.14)} & \quad \min [A_1, A_2] \leq \min [A_3, A_4], \\
\text{(4.15)} & \quad \min [B_1, B_2] \leq \min [B_3, B_4], \\
\text{(4.16)} & \quad \min [A_1 + B_1, A_2 + B_2] \leq \min [A_3 + B_3, A_4 + B_4]
\end{align*}
\]

hold, then inequality

\[
\begin{align*}
\text{(4.17)} & \quad \min [A_1 + B_1, A_1 + B_2, A_2 + B_2] \\
& \leq \min [A_3 + B_3, A_3 + B_4, A_4 + B_4]
\end{align*}
\]

follows except the case \((P)\) where four inequalities

\[
A_2 < A_1, A_3 < \min [A_1, A_4], B_1 < B_2, B_4 < \min [B_2, B_3]
\]

hold.

**Proof.** Let \( \underline{A} = \min [A_1, A_2], \quad \overline{A} = \min [A_3, A_4], \quad B = \min [B_1, B_2], \quad \overline{B} = \min [B_3, B_4] \).

In case where \( \underline{A} = A_1, \overline{B} = B_1 \); or \( A = A_1, \overline{B} = B_2 \); or \( A = A_2, B = B_3, A_1 + B_1 \) or \( A_1 + B_2 \) or \( A_2 + B_3 \) is a smallest term among six terms in (4.17) and then (4.17) holds. In each case, inequality (4.16) can be omitted from conditions.

Then we devide the remained case where \( \underline{A} = A_2 < A_1, \overline{B} = B_1 < B_2 \) into subsequent cases. If \( A_3 \leq A_1 \) or \( B_4 \geq B_2 \), then it holds \( A_3 + B_4 \geq A_1 + B_1 \) or \( A_3 + B_1 \geq A_1 + B_3 \) respectively which shows that (4.16) coincides with (4.17). In case where \( \underline{A} = A_3, \overline{B} = B_3 \), they hold \( A_3 + B_2 < A_1 + B_3, A_3 + B_3 \leq A_3 + B_4 \) and then (4.16) coincides with (4.17). In case where \( \underline{A} = A_4, \overline{B} = B_4 \), they hold \( A_2 + B_3 < A_1 + B_4, A_4 + B_1 \leq A_3 + B_1 \) and then (4.16) coincides with (4.17). In case where \( \underline{A} = A_1, \overline{B} = B_1 \), also they hold \( A_2 + B_2 < A_1 + B_2, A_1 + B_1 \leq A_3 + B_1 \) and then
(4.16) coincides with (4.17).

So that, remaining case is the case (P) which does not always derive (4.17). Q.E.D.

Example of case (P) not leading to (4.17)

\[
\begin{align*}
\min [6, 2] &< \min [3, 5] \\
\min [2, 6] &< \min [7, 4] \\
\min [6+2, 2+6] &< \min [3+7, 5+4] < \min [3+7, 3+4, 5+4].
\end{align*}
\]

By this lemma, we obtain next theorem from theorem 2:

**Theorem 3.** If three inequalities

\[
\begin{align*}
(4.8) & \quad \min [p_{i,1}, p_{j,1}] \leq \min [p_{j,1}, p_{i,1}], \\
(4.12) & \quad \min [p_{i,2}, p_{j,2}] \leq \min [p_{j,2}, p_{i,2}], \\
(4.18) & \quad \min [p_{i,1}+p_{i,2}, p_{j,1}+p_{j,2}] \leq \min [p_{j,1}+p_{j,2}, p_{i,1}+p_{i,2}]
\end{align*}
\]

hold, then the same conclusion as that in theorem 2 follows.

When there is equality in each of these inequalities, either ordering is optimal. This working rule is called as general working rule.

(cf. 4.2)

**Proof.** In lemma they hold \( A_3 = B_3 \) and \( A_4 = B_1 \). If the case (P) holds, then they hold \( A_2 \leq A_3 < A_4 \) from (4.14) and \( B_1 \leq B_1 < B_3 \) from (4.15). The latter inequality becomes \( A_1 \leq B_1 < A_1 \) by \( A_2 = B_3, A_4 = B_1 \) which contradicts with the former. Then the case (P) does not occur. Q.E.D.

4.2 Remarks on Johnson Working Rule

(1) Johnson has stated, in his first paper about optimal sequencing on two and three machines [15], that a sequence which is simultaneously optimal both for first two machines \( M_1, M_2 \) and for last two machines \( M_3, M_3 \) in three machines case becomes optimal for three machines \( M_1, M_2, M_3 \). This fact is true under his working rule [15] for deciding only one definite optimal sequence for two machines in case tie. That is, next theorem 4 holds which can be proved easily by showing that (4.13) holds for each case where \( A \) and \( B \)
take possible values. (cf. proof of theorem 5 in next section).

**Theorem 4.** If job $i$ simultaneously precedes jobs $j$ both on $M_1$, $M_2$ and on $M_2$, $M_3$ under Johnson working rule for deciding one definite optimal sequence for two machines by using

\[
\begin{align*}
(4.8) & \quad \min [p_{i,1}, p_{i,3}] \leq \min [p_{j,1}, p_{i,3}], \\
(4.12) & \quad \min [p_{i,2}, p_{j,3}] \leq \min [p_{j,2}, p_{i,3}]
\end{align*}
\]

respectively, then job $i$ must precede job $j$ under transitive property in order to construct the optimal sequence for three machines $M_1$, $M_2$, $M_3$.

(2) There exist the cases where only one of the above two theorems 3, 4 can derive the optimal sequence.

Then general theorem with detailed sufficient conditions, which includes both theorem 3 and theorem 4 as special cases will be shown in next section.

### 4.3 General Theorem

**Theorem 5.** I. In case where $p_{j,2} \leq p_{i,1}$ or $p_{i,3} \geq p_{j,3}$ holds, if two inequalities

\[
\begin{align*}
(4.8) & \quad \min [p_{i,1}, p_{i,3}] \leq \min [p_{j,1}, p_{i,3}], \\
(4.12) & \quad \min [p_{i,2}, p_{j,3}] \leq \min [p_{j,2}, p_{i,3}]
\end{align*}
\]

hold, then job $i$ must precede job $j$ under transitive property in order to construct the optimal sequence for three machines. When all equalities hold, either ordering is optimal.

II. In case where both $p_{j,2} < p_{i,1}$ and $p_{i,3} < p_{j,3}$ hold, if three inequalities (4.8), (4.12) (they yield $p_{i,2} = p_{j,2}$) and

\[
(4.18) \quad \min [p_{i,1} + p_{i,3}, p_{j,3} + p_{j,3}] \leq \min [p_{j,1} + p_{j,2}, p_{i,2} + p_{i,3}]
\]

hold, then the same conclusions as that in case I hold.

**Proof.** By theorem 3, it is necessary only to prove the case I. In case I, they hold $A = p_{i,1}$ and $B = p_{i,3}$, or $A = p_{i,1}$ and $B = p_{j,3}$, or $A = p_{j,2}$ and $B = p_{j,3}$. Hence (4.13) in theorem 2 follows. Q.E.D.
Conditions in case II is schematically shown in Fig. 1. Under the conditions in theorem 4 where Johnson working rule is applied, the case II in theorem 5 does not occur because equalities hold in (4.8) and (4.12) and job \( j \) precedes job \( i \) for \( M_1, M_2 \) when \( i \) is smaller than \( j \) and job \( j \) precedes job \( i \) for \( M_3, M_4 \) when \( j \) is smaller than \( i \). Hence theorem 5 includes theorem 4. And obviously theorem 5 includes theorem 3. Theorem 5 is used as a criterion in algorithm for determining the optimal sequence for three machines.

5. Constant Order of Definite Adjacent Two Jobs for General \( m \) Machines Case

In this section, generalization of each theorem at former section 4 to \( m \) machines case \((m \geq 2)\) will be presented.

5.1. Generalization of Theorem 3

By using mathematical induction and lemma at section 4.1, we can generalize theorem 3 which shows sufficient conditions with transitive property for deciding constant order of adjacent two jobs \( i, j \) for three machines, to \( m \) machines case \((m \geq 2)\) in order to construct the optimal sequence. The generalized theorem is stated as below:

**Theorem 6.** If \( m(m-1)/2 \) inequalities

\[
\min [p_{i,k}, p_{j,k+1}] \leq \min [p_{i,k}, p_{i,k+1}],
\]

\((k = 1 \sim m-1)\)
(5.2) \[ \min \left[ \sum_{k=u}^{v} p_{i,k}, \sum_{k=u+1}^{v+1} p_{j,k} \right] \leq \min \left[ \sum_{k=u}^{v} p_{j,k}, \sum_{k=u+1}^{v+1} p_{i,k} \right] \]

\((1 \leq u < v \leq m-1)\)

hold, then \(T_k(ij) \leq T_k(ji)\) \((k=2 \sim m)\) hold regardless of the value of \(T_k(l)\) \((k=2 \sim m)\) and so always job \(i\) must precede job \(j\) for adjacent two jobs \(i, j\) regardless of their position in order to construct the optimal sequence. When there exist equalities in all inequalities, either ordering is optimal.

**Proof.**

I. **Sufficient conditions to lead** \(T_k(ij) \leq T_k(ji), (k=2 \sim m).\)

When optimal scheduling procedure is employed and after the processing of certain definite subsequence \(S\) with last job \(l\), let two jobs \(i\) and \(j\) be processed after \(S\).

In case where these two jobs are processed by the order \(ij\), they hold recurrence relations (3.1), (3.2). By successively putting \(T_{q-1}(i)\) into \(T_q(i)\) and putting \(T_q(i), T_{q-1}(ij)\) into \(T_q(ij)\) \((q=2 \sim k)\), finally it holds

\[
T_k(ij) = \max \left[ \max_{r=0 \sim k-2} \left( T_k(l), \sum_{q=k-1-r}^{k-1} p_{i,q}, T_k(l) \right) \right]
\]

\[
+ p_{i,k-1} + p_{j,k-1} - p_{i,k} + \sum_{q=k-1-r}^{k-2} p_{i,q}, T_k(l)
\]

\[
+ p_{i,k-2} + \sum_{q=k-2}^{k-1} p_{j,q} - p_{i,k} + \sum_{q=k-1-r}^{k-3} p_{i,q}, T_k(l)
\]

\[
+ p_{i,k-3} + \sum_{q=k-3}^{k-2} p_{j,q} - p_{i,k} + \sum_{q=k-1-r}^{k-4} p_{i,q}, \ldots, T_k(l)
\]

\[
+ p_{i,k-r} + \sum_{q=k-r}^{k-1} p_{j,q} - p_{i,k} + p_{i,k-1-r}, T_k(l)
\]

\[
+ p_{i,k-1-r} + \sum_{q=k-1-r}^{k-1} p_{j,q} - p_{i,k}
\]

\[
= \max \left[ T_k(l), \sum_{r=0 \sim k-2} \left( T_k(l) \right) + p_{i,k-1-r} \right]
\]
The Order of \( n \) Items Processed on \( m \) Machines [III]

\[
+ \sum_{q=k-l-1}^{k-1} p_{j,q} - p_{i,k} + \sum_{q=k-r-1}^{k-t-2} p_{i,q}\]
\[
= \max \left[ T_k(l), \max_{r=0-k-2} A(ij, t, r, k) \right]
\]

\((k=2 \sim m)\)

where \( T_1(l)=0, \sum_{q=u}^{v} = 0 \) \((u>v)\).

By the same way it holds for the order \( ji \)

\((5.4)\)

\[ T_k(ji) = \max \left[ T_k(l), \max_{r=0-k-2} \max_{t=1-k-1} \left( T_{k-t-1}(l) + p_{j,k-t-1} + \sum_{q=k-t-1}^{k-1} p_{j,q} - p_{i,k} + \sum_{q=k-r-1}^{k-t-2} p_{j,q} \right) \right] \]
\[
= \max \left[ T_k(l), \max_{r=0-k-2} A(ji, t, r, k) \right].
\]

\((k=2 \sim m)\)

Hence if inequality

\((5.5)\)

\[ \max_{r=0-k-2} A(ij, t, r, k) \leq \max_{r=0-k-2} A(ji, t, r, k) \]

holds, then it becomes

\((5.6)\)

\[ T_k(ij) \leq T_k(ji), \quad (k=2 \sim m). \]

Also in order to show the existence of \((5.5)\), it is sufficient to show that next \((k-1)\) inequalities hold for all \( r \) \((r=0 \sim k-2)\):

\((5.7)\)

\[ A(ij, t, r, k) \leq A(ji, t, r, k). \]

Since

\[ A(ij, t, r, k) = T_{k-r-1}(l) + \max_{t=1-k-1} \left( p_{i,k-t-1} + \sum_{q=k-t-1}^{k-1} p_{j,q} - p_{i,k} + \sum_{q=k-r-1}^{k-t-2} p_{j,q} \right) \]
\[ \equiv T_{k-r-1}(l) + B(ij, t, r, k) \]

say, if they hold

\[ B(ij, t, r, k) \leq B(ji, t, r, k); \]

that is, if they hold

\[ \max_{t\in\mathcal{T}} \sum_{s\in\mathcal{S}} \max_{s'\in\mathcal{S}} \mathcal{R}(s, s', t) \]
\[
(5.8) \quad \max \left[ \sum_{q=k-r-1}^{k-1} p_{t,q}, \max_{t=0-r-1} \left( p_{t,k-r-1} + \sum_{q=k-r-1}^{k-1} p_{j,q} - p_{t,k} \right) \right]
\]

for all \( r (r=0\sim k-2) \), then (5.7) hold.

By subtracting the quantity \( \sum_{q=k-r-1}^{k-1} p_{t,q} + \sum_{q=k-r-1}^{k-1} p_{j,q} \) from both sides of (5.8), we have next inequalities by using the relation

\[
\max[-A, -B, -C] = -\min[A, B, C]:
\]

\[
(5.9) \quad \min \left[ \sum_{q=k-r-1}^{k-1} p_{i,q}, \min_{t=0-r-1} \left( \sum_{q=k-r-1}^{k-1} p_{t,q} + \sum_{q=k-t-1}^{k-1} p_{i,q} \right), \sum_{q=k-r-1}^{k-1} p_{j,q} \right]
\]

So that, if (5.9) hold for all \( r (r=0\sim k-2) \) and \( k (k=2\sim m) \), then (5.6) hold and job \( i \) precedes job \( j \).

II. Mathematical induction to prove the existence of (5.9) for all \( m (m\geq2) \).

For \( m=2 \), it becomes \( r=0 \) and both conditions (5.1), (5.2) and (5.9) reduce to the same inequality:

\[
\min[p_{t,i}, p_{j,s}] \leq \min[p_{j,i}, p_{i,s}]
\]

which shows that (5.9) holds for \( m=2 \). Also it holds for \( m=3 \) by theorem 3.

Next we assume that (5.9) hold for all \( k (k=2\sim K) \) under conditions (5.1), (5.2) for \( m=K \):
The Order of n Items Processed on m Machines [III]

\[ \min \{ \rho_{i,k}, \rho_{j,k+1} \} \leq \min \{ \rho_{j,k}, \rho_{i,k+1} \}, \]
\( (k=1 \sim K-1) \)

\[ \min \left[ \sum_{k=1}^{v} \rho_{i,k}, \sum_{k=1}^{v+1} \rho_{j,k} \right] \leq \min \left[ \sum_{k=1}^{v} \rho_{j,k}, \sum_{k=1}^{v+1} \rho_{i,k} \right]. \]
\( (1 \leq u < v \leq K-1) \)

Then we must show that (5.9) hold for \( m=K+1 \) under conditions (5.1), (5.2) for \( m=K+1 \); that is, under conditions (5.10), (5.11) and \( K \) additional conditions

\[ \min \left[ \sum_{k=K-r}^{K} \rho_{i,k}, \sum_{k=K-r+1}^{K+1} \rho_{j,k} \right] \leq \min \left[ \sum_{k=K-r}^{K} \rho_{j,k}, \sum_{k=K-r+1}^{K+1} \rho_{i,k} \right]. \]
\( (r=0 \sim K-1) \)

Also they hold, by assumption, next inequalities for \( k=K \) under conditions (5.10), (5.11):

\[ \min \left[ \sum_{q=K-r-1}^{K-1} \rho_{i,q}, \min \left( \sum_{t=0}^{K-t-2} \rho_{i,q} + \sum_{q=K-t}^{K} \rho_{j,q} \right), \sum_{q=K-t}^{K} \rho_{j,q} \right]. \]
\( \leq \min \left[ \sum_{q=K-r-1}^{K-1} \rho_{j,q}, \min \left( \sum_{t=0}^{K-t-2} \rho_{j,q} + \sum_{q=K-t}^{K} \rho_{i,q} \right), \sum_{q=K-t}^{K} \rho_{i,q} \right]. \)
\( (r=0 \sim K-2) \)

On the other hand, inequalities (5.9) for \( k=K+1 \) are

\[ \min \left[ \sum_{q=K-r}^{K} \rho_{i,q}, \min \left( \sum_{t=0}^{K-t-1} \rho_{i,q} + \sum_{q=K-t+1}^{K+1} \rho_{j,q} \right), \sum_{q=K-t+1}^{K+1} \rho_{j,q} \right]. \]
\( \leq \min \left[ \sum_{q=K-r}^{K} \rho_{j,q}, \min \left( \sum_{t=0}^{K-t-1} \rho_{j,q} + \sum_{q=K-t+1}^{K+1} \rho_{i,q} \right), \sum_{q=K-t+1}^{K+1} \rho_{i,q} \right]. \)
\( (r=0 \sim K-1) \)

These hold for all \( r \) \( (r=0 \sim K-2) \) since (5.14) for all \( r \) \( (r=0 \sim K-2) \) can be derived by taking \( K+1 \) for \( K \) in (5.10), (5.11) and (5.13) for \( r \) \( (r=0 \sim K-2) \) under conditions (5.12).

Hence it is remained to prove the existence of (5.14) for \( r=K-1 \); that is, next inequality:
which is expressed as next form:

\[
\min \left[ \sum_{q=1}^{K} p_{i,q}, \min_{r=0-K-2} \left( \sum_{q=1}^{K-1} p_{i,q} + \frac{K+1}{q=K-t+1} \sum_{q=1}^{K+1} p_{i,q} \right) \right] \\
\leq \min \left[ \sum_{q=1}^{K} p_{j,q}, \min_{r=0-K-3} \left( \sum_{q=1}^{K-1} p_{j,q} + \frac{K+1}{q=K-t+1} \sum_{q=1}^{K+1} p_{i,q} \right) \right]
\]

By putting each term, \( \min \{ \} \), in left and right side of (5.15) as \( A \), \( B \) respectively, (5.15) is expressed as

\[
(5.16) \quad \min \left[ \sum_{q=3}^{K} p_{i,q}, \min_{r=0-K-3} \left( \sum_{q=3}^{K-1} p_{i,q} \right) \right] \\
\leq \min \left[ \sum_{q=3}^{K} p_{j,q}, \min_{r=0-K-3} \left( \sum_{q=3}^{K-1} p_{j,q} \right) \right]
\]

So, in order to prove (5.15), first it is shown that inequality

\[
(5.17) \quad \min \left[ \sum_{q=3}^{K+1} p_{i,q}, \sum_{q=3}^{K+1} p_{j,q} \right] \leq \min \left[ \sum_{q=3}^{K+1} p_{i,q}, \sum_{q=3}^{K+1} p_{j,q} \right]
\]

holds under conditions (5.10), (5.11) and (5.12) because existing inequality for \( r=K-2 \) in (5.14) is

\[
(5.18) \quad \min \left[ \sum_{q=3}^{K+1} p_{i,q}, \sum_{q=3}^{K+1} p_{i,q} \right] \leq \min \left[ \sum_{q=3}^{K+1} p_{i,q}, \sum_{q=3}^{K+1} p_{i,q} \right]
\]

and then we can have (5.17) under conditions (5.10), (5.11), (5.12) by substituting \( \sum_{q=1}^{2} p_{i,q}, \sum_{q=2}^{3} p_{j,q}; \sum_{q=1}^{2} p_{j,q}, \sum_{q=2}^{3} p_{i,q} \) for \( p_{i,s}, p_{i,s}; p_{j,s}, p_{i,s} \) in (5.18) respectively.

Next, inequality (5.16) is proved by using lemma at section 4.1 under conditions (5.17), (5.18) and
(5.19) \[ \min [p_{i,1}, p_{j,2}] \leq \min [p_{j,1}, p_{i,1}] \]
as below:

The case \( P \) where all next inequalities:

(5.20) \[ p_{j,3} < p_{i,1} \]

(5.21) \[ p_{j,1} < p_{i,2} , \]
\[ p_{j,1} < p_{i,1} , \quad A < \sum_{q=3}^{K+1} p_{j,q} , \quad \sum_{q=3}^{K+1} p_{i,q} < B , \]

(5.22) \[ \sum_{q=3}^{K+1} p_{i,q} < \sum_{q=3}^{K+1} p_{j,q} , \]
hold does not occur because if all these inequalities hold, they become \( p_{j,3} < p_{i,1} \) from (5.19), (5.20), (5.21) and then \( p_{j,1} \leq p_{i,1} \) from condition (5.10) which lead to \( \sum_{q=3}^{K+1} p_{j,q} \leq \sum_{q=3}^{K+1} p_{i,q} \) by successively applying (5.11) and (5.12), with \( p_{j,3} < p_{i,3} \), which is a contradiction to (5.22). Hence (5.14) hold for all \( r (r=0\sim K-1) \) and then (5.9) hold for all \( k (k=2\sim K+1) \) and theorem holds for all \( m (m \geq 2) \).

Q.E.D.

For \( m=2 \), this theorem coincides with well known Johnson’s theorem for two machines [15] and in case where \( m=3 \) in this theorem we obtain theorem 3.

Theorem 6 is efficiently applied to the reduction of the number of sequences in branch-and-bound algorithms and other algorithms for obtaining the optimal sequence for \( m \) machines \((m \geq 3)\) in flow shop where no passing is allowed. cf. [11].

Also this theorem gives some ways of obtaining the approximate sequence similar as Johnson approximation [14] for any \( m \) machines \((m \geq 3)\) in such flow shop. (cf. section 6.3)

5.2 Generalization of General Theorem 5

Next theorem 7 is a generalization of theorem 5 to \( m \) machines \((m \geq 2)\), which includes the theorem 6 and generalization of theorem
4 presented in next section 5.3 as its special cases.

**Theorem 7.** Let job \( i \) and \( j \) be adjacent two jobs.

I. In case where relation (A): \( p_{i,k_{0}+1} < p_{i,k_{0}}, p_{i,k_{0}+n_{0}} < p_{i,k_{0}+n_{0}+1}; \) does not hold for any \( k_{0} (k_{0} = 1 - m - 2) \) and for any \( n_{0} (n_{0} = 1 - m - k_{0} - 1) \) for each \( k_{0} \), if next \((m-1)\) inequalities:

\[
\min \{ p_{i,k}, p_{j,k+1} \} \leq \min \{ p_{j,k}, p_{i,k+1} \} \quad (k = 1 - m - 1)
\]

hold, then job \( i \) must precedes job \( j \) under transitive property in order to construct the optimal sequence for \( m \) machines \((m \geq 2)\).

When equalities hold for all inequalities, either ordering is optimal.

II. In case where above relation (A) holds for a certain \( k_{0} (k_{0} = 1 - m - 2) \) and a certain \( n_{0} (n_{0} = 1 - m - k_{0} - 1) \) for this \( k_{0} \), if additional inequalities to (5.23):

\[
\min \left[ \sum_{q=u}^{v} p_{i,q}, \sum_{q=u+1}^{v+1} p_{j,q} \right] \leq \min \left[ \sum_{q=u}^{v} p_{j,q}, \sum_{q=u+1}^{v+1} p_{i,q} \right]
\]

\((1 \leq u < v \leq m - 1)\)

hold, then the same conclusions as that in I hold.

**Proof.** By theorem 6 it is sufficient to show that (5.9) hold for case I.

Case I. (1) In case \( p_{i,k} \leq p_{j,k+1} \) for each \( k (k = 1 - m - 1) \), it becomes \( p_{i,k} \leq p_{i,k+1} \) from (5.23) and so \( \sum_{q=k-r-1}^{k-1} p_{i,q} \) is a smallest term in (5.9) for all \( r \) and \( k \). Hence (5.9) hold.

(2) In case \( p_{i,k} \leq p_{j,k+1} (k = 1 - k_{1} - 1) \), \( p_{j,k_{1}+1} < p_{i,k_{1}} \), it becomes \( p_{j,k_{1}+1} \leq p_{i,k}, p_{j,k} \) for each \( k (k_{1} < k \leq m - 1) \) by assumption and then \( \sum_{q=k-r-1}^{k-1} p_{i,q} \) or \( \sum_{q=k-r-1}^{k_{1}-1} p_{i,q} + \sum_{q=k_{1}+1}^{k} p_{j,q}, \sum_{q=k-r}^{k} p_{j,q} \) is a smallest term in (5.9).

(3) In case \( p_{j,1} < p_{i,1} \), it holds always \( p_{j,k+1} \leq p_{i,k}, p_{j,k} \) for each \( k (2 \leq k \leq m - 1) \) by assumption and \( \sum_{q=k-r}^{k} p_{j,q} \) is a smallest term in (5.9).

Q.E.D.

Relation (A) is schematically shown in Fig. 2. Theorem 7 can be used as a criterion in algorithm for determining the optimal sequence.
The Order of n Items Processed on m Machines [III] 179

\[ i \quad p_{i,k_0} \quad p_{i,k_0+1} \quad \ldots \quad p_{i,k_0+n_0} \leq p_{i,k_0+n_0+1} \]

\[ j \quad p_{j,k_0} \geq p_{j,k_0+1} \quad \ldots \quad p_{j,k_0+n_0} \quad p_{j,k_0+n_0+1} \]

Fig. 2. Relation (A).

for \( m \) machines \((m \geq 3)\) in flow shop under consideration.

5.3 Generalization of Theorem 4 under Johnson Working Rule

It will be shown that generalization of theorem 4 \((m=3)\) which is the case where Johnson working rule for two machines case is used, is a special case of general theorem 7.

Generalization of theorem 4 to \( m \) machines case \((m \geq 2)\) is stated as below:

**Corollary.** If job \( i \) directly precedes job \( j \) on each adjacent two machines \( M_k, M_{k+1} \) under Johnson working rule by using

\[
(5.23) \quad \min[p_{i,k}, p_{j,k+1}] \leq \min[p_{j,k}, p_{i,k+1}]
\]

for each \( k \) \((k=1 \sim m-1)\), then job \( i \) directly precedes job \( j \) on \( m \) machines case, that is, if an optimal sequence on each adjacent two machines \( M_k, M_{k+1} \) \((k=1 \sim m-1)\) coincides with each other, then this sequence becomes an optimal sequence on \( m \) machines, when Johnson working rule is employed in flow shop where no passing is allowed.

**Proof.** It is sufficient to show that case II in theorem 7 does not occur. If \((m-1)\) inequalities \((5.23)\) hold in case II and Johnson working rule is used for two jobs \( i, j \) where \( i \) is smaller than \( j \) \((j \) is smaller than \( i)\), then, as known from Fig. 2 and \((5.23)\), job \( j \) precedes job \( i \) on two machines \( M_k, M_{k+1} \); \( k=k_0, \) or \( k_0+1, \ldots, k_0+n_0-1 \) \((k=k_0+n_0, \) or \( k_0+n_0-1, \ldots, k_0+1)\) which contradicts to the assumption.

Q.E.D.
6. Approximate Algorithms

The problem is to decide the approximate sequence which has sufficiently small makespan on \( m \) machines in flow shop where no passing is allowed.

The methods of approximation are classified into following three types, that is,

1) Determination of the position of \( n_k \) jobs at stage \( k \) to construct the approximate sequence of \( n \) jobs. Usually the case where \( n_k=1 \) is considered.

2) Systematic reformation of the order of some adjacent jobs in the suitably determined initial sequence of \( n \) jobs.

3) Separation of the set of \( n \) jobs into \( r \) subsets followed by suitable connection of the optimal or approximate subsequence of each subset.

Next, some approximate algorithms related to these types will be presented by using the inequalities in theorem 6.

6.1 Approximate Algorithms 1

Let \( n \) jobs be named along 1, 2, \( \ldots \), \( n \) as usual.

Step 1. Each of the optimal sequences obtained by each inequality of (5.1) and (5.2) is determined by general working rule as the same way for two machines case, under transitive property of the order of jobs determined by each inequality.

Step 2. For each pair \((i, j)\) among \( n \) jobs, frequency of the order \( ij \) and frequency of the order \( ji \) in all optimal sequences determined at step 1 are counted respectively. Then the order among two orders \( ij, ji \) which has larger number of frequency is determined. If the number of frequency is equal to each other, then both orders \( ij, ji \) are taken.

Example \((m=3, n=6)\). For three machines case, inequalities become (4.8), (4.12) and (4.18) in theorem 3, a special case of theorem 6.
The Order of \( n \) Items Processed on \( m \) Machines [III]

<table>
<thead>
<tr>
<th>( i )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_{i,1} )</td>
<td>6</td>
<td>12</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>( p_{i,2} )</td>
<td>7</td>
<td>2</td>
<td>6</td>
<td>11</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>( p_{i,3} )</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

Processing Time

Then optimal sequences by (4.8), (4.12) and (4.18) become as below respectively (Step 1):

(1) \( [643152] \) (2) \( 235641 \) (3) \( [345612] \)

\( [643512] \) \( [354612] \)

And the order of two jobs of each pair \((i, j)\) is determined as below. (Step 2):

(1, 2) 12 (2, 4) 42 (3, 5) 35 (3, 6) 36
(1, 3) 31 (3, 4) 34 (4, 5) 45, 54 (4, 6) 64
(2, 2) 32 (1, 5) 51 (1, 6) 61 (5, 6) 56
(1, 4) 41 (2, 5) 52 (2, 6) 62

Step 3. Each job \( i \) which is increasing from \( i=2 \) to \( i=n \) is classified into one of two subsets \( N_1, N_2 \) defined as follows. Subset \( N_1 \) is composed of the job \( i \) which has the unique order for all job \( j (<i) \). Subset \( N_2 \) is composed of the job \( i \) which has two orders \( ij, ji \) for certain job \( j (<i) \). Job 1 belongs to \( N_1 \) only when unique order among the pair \((1, 2)\) is determined.

Under the above preparations, the procedures in next step 4 construct the approximate sequence successively.

Step 4. First the order of the jobs in \( N_1 \) along the increasing number of job is determined by following each unique order of two jobs \((i, j)\).

Next the position of each job in \( N_2 \) is determined along the increasing number of job by following each unique or twofold order of two jobs \((i, j)\) such that already determined order of jobs is not changed. Lastly sequence of \( n \) jobs can be obtained.

If any sequence of \( n \) jobs cannot be determined by the above
ways, then the possible sequences that satisfy almost all unique or twofold orders are taken. Then the sequence having minimum makespan among them is the approximate sequence.

Example. (Continued)

\[ N_1 = \{1, 2, 3, 4, 6\}, \quad N_2 = \{5\}. \] (Step 3).

Construction of the approximate sequence is made as below (Step 4):

<table>
<thead>
<tr>
<th>Stage</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_1 ):</td>
<td>12</td>
<td>312</td>
<td>3412</td>
<td>36412</td>
</tr>
</tbody>
</table>

\( N_2 \): (Stage 5). Order 45 contradicts to the unique orders 56, 64 and then the order 54 determines one approximate sequence 356412 with makespan 57 which is just an optimal sequence in this example.

6.2 Remarks

1) Product-latine method [16] of the matrix for Hamiltonian path can be used after step 2 in the above algorithm 1 by taking both order \( ij \), \( ji \) as the element corresponding to twofold order of \( (i, j) \).

2) Mean approximate ratio for 15 examples of three machines and six jobs was 98.45% where

\[
\begin{align*}
\text{(Approximate ratio)} & = \frac{\text{(Optimal time)}}{\text{(Approximate time)}} \times 100\%. \\
\end{align*}
\]

3) Additional steps to make the makespan of the sequence of \( n \) jobs obtained in step 4 more smaller by exchanging suitable jobs of this sequence may be considered.

6.3 Other Related Algorithms

1) In the above algorithm 1, only some of the inequalities (5.1), (5.2) can be taken instead of all inequalities (5.1), (5.2) and then approximate sequence will be obtained by following the same steps.

For example, Johnson approximation for three machines [14] is obtained by using only the inequality (4.18) in the above example and general formula of this type for \( m \) machines case is next inequality
where \( u=1 \) and \( v=m-1 \) in (5.2):

\[
(6.1) \quad \min \left[ \sum_{k=1}^{m-1} p_{i,k}, \sum_{k=2}^{m} p_{j,k} \right] \leq \min \left[ \sum_{k=r}^{m-1} p_{j,k}, \sum_{k=2}^{m} p_{i,k} \right].
\]

But two additional inequalities:

\[
(6.2) \quad \min \left[ \sum_{k=1}^{m-2} p_{i,k}, \sum_{k=2}^{m-1} p_{j,k} \right] \leq \min \left[ \sum_{k=1}^{m-2} p_{j,k}, \sum_{k=2}^{m-1} p_{i,k} \right],
\]

\[
(6.3) \quad \min \left[ \sum_{k=2}^{m-1} p_{i,k}, \sum_{k=3}^{m} p_{j,k} \right] \leq \min \left[ \sum_{k=2}^{m-1} p_{j,k}, \sum_{k=3}^{m} p_{i,k} \right].
\]

for instance may yield better approximate sequence.

2) If, in any above algorithm, we use only step 1 and then decide the approximate sequence as the sequence with least makespan among all optimal sequences for each inequality corresponding to (5.1) and/or (5.2), or similar inequalities, we obtain another algorithm. Algorithm in [13] can be considered as this type.

3) An algorithm to decide one job of the approximate sequence successively from the first job is formulated by using the relations

\[
(6.4) \quad T_k(ij) \leq T_k(ji) \quad \text{for all } k (k=2 \sim m)
\]

that are the same as (3.3) and (5.6), or one relation

\[
(6.5) \quad T_m(ij) \leq T_m(ji)
\]

where each \( T_k(ij) \), \( T_k(ji) \) can be calculated by recurrence relations (3.1), (3.2) and so on successively.

This algorithm is stated as below:

At each stage \( k (k=1 \sim n-1) \) there has been already decided the presubsequence \( i_1 \cdots i_{k-1} \) with \((k-1)\) jobs, then for each pair \((i, j)\) of two jobs among remained \((n-k+1)\) jobs, the order \(ij\) is determined by relations (6.4) where both orders \(ij\), \(ji\) are taken when inequalities with reverse direction hold for the pair \((i, j)\), or by relation (6.5).

Next one job \(i_k\) which precedes all other jobs \(j\) becomes \(k\)th job of the approximate sequence. If such job does not exist, the jobs \(i_k\) that have a large number of jobs preceded by \(i_k\) are considered as the \(k\)th jobs of the approximate sequence.
Then the approximate sequence with least makespan among all sequences of $n$ jobs obtained at stage $n$ is determined.

References


Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.
