J. Operations Research Soc. of Japan
Vol. 9, Nos. 3 & 4, July 1967

ON THE BOUND OF MAKESPANS AND ITS APPLICATION IN *M* MACHINE SCHEDULING PROBLEM

ICHIRO NABESHIMA

University of Electro-Communications

(Received February 28, 1967)

Preface

For so-called m machine scheduling problem, recently there has been applied branch and bound method to min-makespan problem both in case where no passing is allowed [1] [2] [3] and in case where passing is allowed [4]. In those cases, to make more exact the value of the lower bound of makespans of sequences with definite presubsequence is very important in order to obtain an optimal solution using branch and bound algorithm (B. B. algorithm) by checking as smaller number of nodes as possible, under consideration on the quantity of calculations on the other hand.

In this paper, such more exact lower bound (revised lower bound) than already given in Refs. [1]~[4] is presented for each case by using Johnson's criterion for two machines [5] (§ 1) and, for the purpose of estimating better lower bound for the case where no passing is allowed, other devices for obtaining some different types of lower bound of makespans of sequences with definite backsubsequence and/or definite presubsequence will be shown with applications to B. B. algorithm (§ 2). Next, upper bound of makespans of sequences with definite presubsequence in case where no passing is allowed is presented with application to B. B. algorithm for max-makespan problem (§ 3). In each of these sections numerical examples will be shown in order to show the effectiveness of each bound. Finally, additional remarks mill be shown, especially concerning the sensitivity of the "algorithm" for each problem (§ 4).

§ 1. Revised Lower Bound of Makespans of Sequences with Definite Presubsequence

In papers [1]~[4] already published up to now concerning branch and bound algorithm for optimal sequencing of n jobs through m machines along same machines order for each job, lower bound of makespans of sequences with definite presubsequence for each node has the terms that represent the sum of the processing time of each job belonging to set of unordered remained jobs at every machine where idle time of each machine by these processing doesn't been taken into account. But, estimation of this idle time can be taken into consideration by appling Johnson's criterion [5] for two machines case as shown in the following. Further let m machines be named by $M_1, M_2, \cdots, M_m$ and be used in this order for any job and processing time of job i on M_k be $m_{k,i}$ ($i=1\sim n, k=1\sim m$).

1.1. Case where no passing is allowed

1.1.1. Revised lower bound

First it's considered the case where no passing of job is allowed. Let J_r ($r=1\sim n-1$) be a definite presubsequence of r jobs among n jobs that are processed on m machines and $T_k(J_r)$ be the completion time of this sequence J_r on machine M_k ($k=1\sim m, r=1\sim n-1$) and $\bar{J}_r$ be the set of all unordered remaining $(n-r)$ jobs after the processing of J_r.

Then, for each two machines M_k, M_{k+1} ($k=1\sim m-1$) adjoining each other, let $i_{r+1}^k\, i_{r+2}^k \cdots i_n^k$ be the sequence of $(n-r)$ jobs in $\bar{J}_r$ which is determined by next Johnson's criterion (1.1) for independent two machines M_k, M_{k+1}: that is, for any two jobs i,j in $\bar{J}_r$, if it holds inequality

$$\min\,[m_{k,i}, m_{k+1,j}] \leqq \min\,[m_{k,j}, m_{k+1,i}], \tag{1.1}$$

then job i must precede job j in order to minimize the makespan of the sequence of $(n-r)$ jobs in $\bar{J}_r$ on M_k, M_{k+1} alone.

Hence, this sequence $i_{r+1}^k, i_{r+2}^k \cdots i_n^k$ must be processed on M_k, M_{k+1} along this order after the time $T_k(J_r)$ on M_k and also $T_{k+1}(J_r)$ on M_{k+1}

and let $T^k_{k+1}(\bar{J}_r)$ be elapsed time of the processing of $i^k_{r+1}\, i^k_{r+2}\cdots i_n{}^k$ on M_{k+1} after the time $T_{k+1}(J_r)$ [cf. Fig. 1]. As shown in Fig. 1, completion time T_{k+1} of the sequence $i^k_{r+1}\, i^k_{r+2}\cdots i_n{}^k$ on M_{k+1} is obviously a possible earliest completion time of any sequence of $(n-r)$ jobs in $\bar{J}_r$ on M_{k+1}.

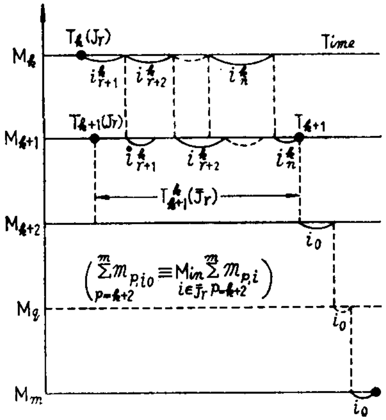

Fig. 1. Processing of $\bar{J}_r$ on M_{k+1}.

So that it must be obtained next revised lower bound LB(J_r) (1.2) of the makespans of every sequences of n jobs with definite presubsequence J_r:

$$\mathrm{LB}(J_r)=\max\left(\begin{array}{l} T_2(J_r)+T_2{}^1(\bar{J}_r)+\min\limits_{i\in\bar{J}_r}\sum\limits_{p=3}^{m} m_{p,\,i}, \\[2ex] T_3(J_r)+T_3{}^2(\bar{J}_r)+\min\limits_{i\in\bar{J}_r}\sum\limits_{p=4}^{m} m_{p,\,i}, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ T_{m-1}(J_r)+T^{m-2}_{m-1}(\bar{J}_r)+\min\limits_{i\in\bar{J}_r} m_{m,\,i}, \\[2ex] T_m(J_r)+T^{m-1}_m(\bar{J}_r). \end{array}\right) \qquad (1.2)$$

$$(r=1\sim n-1).$$

Here, LB(J_r) is an increasing function of r for $J_{r_1}\subset J_{r_2}$ $(r_1<r_2)$ and LB(J_{n-1}) is equal to real total elapsed time (makespan) TE(J_{n-1}) of a sequence uniquely determined by presubsequence J_{n-1}.

Also, $T_k(J_r)\,(k=1\sim m, r=1\sim n-1)$ is calculated by next recurrent relation where $J_r\equiv J_{r-1}\,i_r\,(i_r$ is the last job of $J_r)$ and $T_0(J_r)\equiv 0, T_k(J_0)\equiv 0$:

$$T_k(J_r)=\max\,[T_{k-1}(J_r), T_k(J_{r-1})]+m_{k,\,i_r}\ (k=1\sim m, r=1\sim n-1). \tag{1.3}$$

Especially for obtaining the valve of each $T^k_{k+1}(\bar{J}_r)\equiv T^k_{k+1}(i^k_{r+1}\,i^k_{r+2}\cdots i_n{}^k)$ $(k=1\sim m-1, r=1\sim n-1)$, next recurrent relation (1.4) similar to (1.3) can be used:

$$\begin{aligned} T^k_{k+1}(i^k_{r+1}\cdots i^k_{r+l})=\max\,[&T^k_{k+1}(i^k_{r+1}\cdots i^k_{r+l-1}), T_k(J_r)\\ &+\sum_{j=1}^{l} m_{k,\,i^k_{r+j}}]+m_{k+1,\,i^k_{r+l}} \qquad (1.4)\\ &\qquad\qquad (l=1\sim n-r)\end{aligned}$$

where $T^k_{k+1}(i^k_r)\equiv T_{k+1}(J_r)$ $(l=1)$.

1.1.2. Comparison with lower bound in refs. [3] [4] and B. B. algorithm

Revised lower bound (1.2) is more exact than the lower bound already given [3] [4] as follows:

$$\mathrm{LB}(J_r)=\max\left(\begin{array}{l} T_1(J_r)+\sum\limits_{i\in\bar{J}_r} m_{1,\,i}+\min\limits_{i\in\bar{J}_r}\sum\limits_{p=2}^{m} m_{p,\,i},\\ T_2(J_r)+\sum\limits_{i\in\bar{J}_r} m_{2,\,i}+\min\limits_{i\in\bar{J}_r}\sum\limits_{p=3}^{m} m_{p,\,i},\\ T_3(J_r)+\sum\limits_{i\in\bar{J}_r} m_{3,\,i}+\min\limits_{i\in\bar{J}_r}\sum\limits_{p=4}^{m} m_{p,\,i},\\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\ T_{m-1}(J_r)+\sum\limits_{i\in\bar{J}_r} m_{m-1,\,i}+\min\limits_{i\in\bar{J}_r} m_{m,\,i},\\ T_m(J_r)+\sum\limits_{i\in\bar{J}_r} m_{m,\,i}. \end{array}\right) \quad (r=1\sim n-1) \tag{1.5}$$

That is to say, since $T^k_{k+1}(\bar{J}_r)\geqq\sum\limits_{i\in\bar{J}_r} m_{k+1,\,i}$, if it's neglected the first term in maximum bracket of (1.5), $\mathrm{LB}(J_r)$ of (1.2) is larger than that of (1.5) and more efficient in the sense that $T^k_{k+1}(\bar{J}_r)$ characterizes a possible minimum sum of the idle time of M_{k+1} by processing of any backsubsequence of unordered remained $(n-r)$ jobs in $\bar{J}_r$ on machine M_{k+1} $(k=1\sim m-1)$ and the calculations of the valve of $T^k_{k+1}(\bar{J}_r)$ which is the same as that of $T_{l+1}(J_r)$ isn't so complicated. In the next section it will be shown this facts by solving some examples using B. B. algorithm with lower bound (1.2) and (1.5) respectively, resulting that the number of nodes by (1.2) is smaller than that by (1.5).

As B. B. algorithm with revised lower bound (1.2) is the same as that with (1.5) [1]~[4], there shall be no language of it.

1.1.3. Numerical examples

In this section some numerical examples are solved by branch and bound algorithm with revised lower bound (1.2) and lower bound (1.5) respectively.

Then, efficiency of the revised lower bound becomes clear.

Example 1. $(m=3, n=6)$ [1]

Processing Time (hrs.)

i	1	2	3	4	5	6
$m_{1,i}$	6	12	4	3	6	2
$m_{2,i}$	7	2	6	11	8	14
$m_{3,i}$	3	3	8	7	10	12

In order to decide a subsequence $i^k_{r+1}\, i^k_{r+2} \cdots i_n^k$ of $(n-r)$ jobs in $\bar{J}_r$ on M_k, M_{k+1} for each node (J_r), it's sufficient to decide an optimal sequence of n jobs on two machines M_k, M_{k+1} in advance.

In this example, for M_1, M_2 an optimal sequence is say 643152 and for M_2, M_3 it's 235641. Then, calculations of revised lower bound LB(J_r) of each node (J_r) is made as shown in the following for three nodes. That is, for a node $(J_1)\equiv(3)$, since $i_2^1\, i_3^1 \cdots i_6^1=64152$ and $i_2^2\, i_3^2 \cdots i_6^2=25641$, each term in maximum bracket of LB(3) can be calculated as below:

Completion time of a sequence $J_r\, i^1_{r+1}\, i^1_{r+2} \cdots i_n^1$ on M_2.

Order of jobs	3	6	4	1	5	2	
(M_1	4	6	9	15	21	33)	$T_2 + \min\limits_{i \in \bar{J}_r} m_{3,i} = 52+3=55$
M_2	10	24	35	42	50	52	

Completion time of a sequence $J_r\, i^2_{r+1}\, i^2_{r+2} \cdots i_n^2$ on M_3.

Order of jobs	3	2	5	6	4	1	
(M_2	10	12	20	34	45	52)	LB(3) = max [55, 56] = 56
M_3	18	21	31	46	53	56	

Next for a node $(J_2) \equiv (35)$, since $i_3^1\, i_4^1\, i_5^1\, i_6^1 = 6412$ and $i_3^2\, i_4^2\, i_5^2\, i_6^2 = 2641$, LB(35) can be calculated as below briefly:

	3	5	6	4	1	2	
(M_1	4	10	12	15	21	33)	52+3=55
M_2	10	18	32	43	50	52	
	3	5	2	6	4	1	
(M_2	10	18	20	34	45	52)	LB(35) = max [55, 56] = 56
M_3	18	28	31	46	53	56	

Another example is shown for LB(356) as below; here they hold $i_4^1\, i_5^1\, i_6^1 = 412$ and $i_4^2\, i_5^2\, i_6^2 = 241$

	3	5	6	4	1	2	
(M_1	4	10	12	15	21	33)	
M_2	10	18	32	43	50	52	52+3=55
	3	5	6	2	4	1	
(M_2	10	18	32	34	45	52)	
M_3	18	28	44	47	54	57	LB(356) = max [55, 57] = 57

By similar calculations of each value of LB(J_r), scheduling tree for example 1 becomes as in Fig. 2 where upper number at each node denotes a revised lower bound and lower number in parenthesis denotes a lower bound (1.5) already given [1].

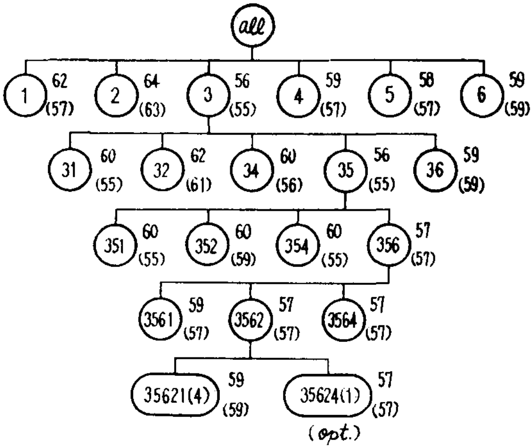

Fig. 2. Scheduling Tree of Example 1.

Number of nodes by revised lower bound (1.2) becomes 20 which is smaller than 58 by lower bound [1.5] [1], and optimal sequences are 365241 and 356412, 356421 with 57 hrs. further obtained if node (3564) may be branched.

Another two examples have next processing time respectively:

Example 2. ($m=5, n=6$) [3]

i	1	2	3	4	5	6
$m_{1,i}$	5	6	30	2	3	4
$m_{2,i}$	8	30	4	5	10	1
$m_{3,i}$	20	6	5	3	4	4
$m_{4,i}$	15	7	9	28	1	1
$m_{5,i}$	5	17	10	8	15	4

Example 3. ($m=3, n=6$)

i	1	2	3	4	5	6
$m_{1,i}$	5	3	12	2	9	11
$m_{2,i}$	9	8	10	6	3	1
$m_{3,i}$	6	2	4	12	7	3

Then, let Nr=Number of nodes by revised lower bound (1.2)

No=Number of nodes by lower bound (1.5)

Na=Number of nodes by lower bound (1.5) with additional two terms $h^{(1)}, h^{(2)}$ [3] for three machines case:

$$h^{(1)}=T_1(J_r)+\sum_{p=1}^{3} m_{p,k_0}+\sum_{i\in\bar{J}_r-k_0} \min(m_{1,i}, m_{3,i}),$$

where $\sum_{p=1}^{3} m_{p,k_0} \equiv \max_{i\in\bar{J}_r} \sum_{p=1}^{3} m_{p,i}$

and $\quad h^{(2)}=T_2(J_r)+\sum_{p=2}^{3} m_{p,k_1}+\sum_{i\in\bar{J}_r-k_1} \min(m_{2,i}, m_{3,i}),$

where $\sum_{p=2}^{3} m_{p,k_1} = \max_{i\in\bar{J}_r} \sum_{p=2}^{3} m_{p,i}.$

Then, results are shown in Table 1.

Table 1: Number of Nodes in Scheduling Tree.

Example	Nr	No	Na
1	20	58	/
2	64	74	/
3	24	37	37

1.1.4. Remarks

Some remarks concerning this section will be itemized as follows:

1. Generalization of the additional terms [3] ($m=3$) to lower bound (1.5) is given as below for m machines case ($m\geqq 3$).

$$h^{(q)}=T_q(J_r)+\sum_{p=q}^{m} m_{p,k_{q-1}}+\sum_{i\in\bar{J}_r-k_{q-1}} \min(m_{q,i}, m_{m,i}),$$

where

$$\sum_{p=q}^{m} m_{p,k_{q-1}}=\max_{i\in\bar{J}_r} \sum_{p=q}^{m} m_{p,i}. \quad (q=1\sim m-1)$$

Here, only for $\mathrm{LB}(J_r)=\max_{q=1\sim m-1} [h^{(q)}]$ it holds $\mathrm{LB}(J_{n-1})\leqq \mathrm{TE}(J_{n-1})$.

2. Another way to eliminate the number of nodes in scheduling tree is to use the next theorem which determines the definite order of two neighbouring jobs regardless of their position in sequence: that is,

Theorem.[1] In m machines case, for each neighbouring two jobs i, j, if next $\frac{1}{2}m(m-1)$ inequalities (1) and (2) hold;

$$\min[m_{k,i}, m_{k+1,j}] \leqq \min[m_{k,j}, m_{k+1,i}] \quad (k=1\sim m-1) \qquad (1)$$

$$\min[\sum_{k=p}^{q} m_{k,i}, \sum_{k=p+1}^{q+1} m_{k,j}] \leqq \min[\sum_{k=p}^{q} m_{k,j}, \sum_{k=p+1}^{q+1} m_{k,i}], \qquad (2)$$
$$(p<q;\ p=1\sim m-2, q=2\sim m-1)$$

then job i must always precede job j regardless of their position.

Corolary. In three machines case, inequalities (1) and (2) in the theorem become next forms:

$$\min[m_{1,i}, m_{2,j}] \leqq \min[m_{1,j}, m_{2,i}], \min[m_{2,i}, m_{3,j}] \leqq \min[m_{2,j}, m_{3,i}] \qquad (1)$$

$$\min[m_{1,i}+m_{2,i}, m_{2,j}+m_{3,j}] \leqq \min[m_{1,j}+m_{2,j}, m_{2,i}+m_{3,i}]. \qquad (2)$$

Determination of this definite order ij is very simple because each inequality in (1) and (2) has transitive property. By using this theorem, if definite order ij is determined, then any nodes that contain the order ji or shall contain the order ji afterwards can be omitted in branching a node. For example, scheduling tree of the example ($m=3, n=6$) in Ref. [3] (p. 184) which has 367 nodes by lower bound (1.5) and 65 nodes by (1.5) with additional two terms $h^{(1)}, h^{(2)}$, has 49 nodes by revised lower bound (1.2) appling this corolary and about 93 nodes only by revised lower bound.

1.2. Case where passing is allowed

For this case, former paper [4] has presented some branch and bound algorithms for optimal sequencing of n jobs through $m(m \geqq 4)$ machines where passing of job is allowed. There, formulation of the lower bound

1) The proof of this theorem will be shown in paper to be published in future [6].

of each node hasn't taken into account the sum of idle time of each machine by processing of each job belonging to the set of unordered remained jobs. In the following it can be taken into consideration by using johnson's criterion for two machines in order to make more exact that lower bound.

For the present case the order of n jobs may not be the same for each of m machines, but for obtaining optimal solution it can be assumed that the order of n jobs is the same for first two machines M_1, M_2 and for last two machines M_{m-1}, M_m respectively.

First, some terminologies must be defined as follows.

Let $J_r^{12}, J_r^k\ (k=3\sim m-2), J_r^{m-1,m}$ be definite subsequence of r jobs among n jobs that are processed on machine M_1 and M_2, $M_k\ (k=3\sim m-2)$, M_{m-1} and M_m respectively and let

$$(J_r)=\begin{pmatrix} J_r^{12} \\ J_r^{3} \\ J_r^{4} \\ \vdots \\ J_r^{m-2} \\ J_r^{m-1,m} \end{pmatrix} \qquad (r=1\sim n-1) \tag{1.6}$$

denotes the set of all sequences of n jobs that have definite subsequence $J_r^{12}, J_r^k\ (k=3\sim m-2), J_r^{m-1,m}$ as their first r jobs processed on M_1 and M_2, $M_3\ (k=3\sim m-2)$, M_{m-1} and M_m respectively.

Next let $T_k(J_r^{12})\ (k=1,2)$, $T_k(J_r^k)\ (k=3\sim m-2)$, $T_k(J_r^{m-1,m})\ (k=m-1,m)$ be the earliest completion time of the sequence $J_r^{12}, J_r^k\ (k=3\sim m-2), J_r^{m-1,m}$ on $M_k\ (k=1,2)$, $M_k\ (k=3\sim m-2)$, $M_k\ (k=m-1,m)$ respectively if necessary by considering the following sequences of some unordered remained jobs of the former sequence J_r^k and let $l_{r,k}$ be the last job of $J_r^k\ (k=1\sim m)$ $(l_{r,1}=l_{r,2}=l_{r,12}, l_{r,m-1}=l_{r,m}=l_{r,m-1,m},$ the same for $J_r^k)$ and $\bar{J}_r^{12}, \bar{J}_r^k\ (k=3\sim m-2), \bar{J}_r^{m-1,m}$ be the set of all unordered remained jobs after the processing of presubsequence $J_r^{12}, J_r^k\ (k=3\sim m-2), J_r^{m-1,m}$ respectively.

1.2.1. Value of the earliest completion time $T_k(J_r^k)$

Each value of $T_k(J_r^{12})\,(k=1,2)$, $T_k(J_r^k)\,(k=3\sim m-2)$ and $T_k(J_r^{m-1,m})$ $(k=m-1,m)$ is determined as follows:

1. $T_k(J_r^{12})\ (k=1,2)$

$$T_1(J_r^{12})=\sum_{i\in J_r^{12}} m_{1,i},$$

$$T_2(J_r^{12})=\max\begin{bmatrix} T_1(J_r^{12}), \\ T_2(J_r^{12}-l_{r,12}) \end{bmatrix}+m_{2,l_{r,12}}, \tag{1.7}$$

$$(r=1\sim n-1)$$

where $J_r^{12}-l_{r,12}$ denotes the sequence obtained from J_r^{12} by excluding its last job $l_{r,12}$.

2. $T_k(J_r^k)\,(k=3\sim m-1)$.

As each J_r^k is defined independently for each other, there may be jobs of J_r^k not belonging to some of the former $J_r^{12}, J_r^3, \cdots, J_r^{k-1}$.

Hence for example, let i_3^{12} be a job in $J_r^3\cap\bar{J}_r^{12}$ having a smallest position number in J_r^3 and $i_4^{3,12}$, $i_{4,12}^3$ be a job in $J_r^4\cap\bar{J}_r^3\cap\bar{J}_r^{12}$, $J_r^4\cap\bar{J}_r^3\cap J_r^{12}$ respectively both having a smallest presition number in J_r^4, then let i_4^3 be a job in $J_r^4\cap\bar{J}_r^3$ having a smallest position number in J_r^4; that is, a job equal to either $i_4^{3,12}$ or $i_{4,12}^3$. Generally, let $i_{k,p-2}^{k-1,\cdots,p-1}$ be a job in $J_r^k\cap\bar{J}_r^q\cap J_r^{p-2}$ for all $q=p-1\sim k-1$ and for each $p\,(3\leqq p\leqq k)$ having a smallest position number in J_r^k where for example $i_{4,1}^{3,2}=i_4^{3,12}\,(k=4, p=3)$ and i_k^{k-1} be a job in $J_r^k\cap\bar{J}_r^{k-1}$ having a smallest position number in J_r^k; that is, a job equal to either of $i_{k,p-2}^{k-1,\cdots,p-1}$.

Then, for the determination of the value of $T_3(J_r^3)$ which is by definition the earliest completion time of J_r^3 on M_3, subsequence of jobs in $J_r^3\cap\bar{J}_r^{12}$ must be processed on M_1 and M_2 after J_r^{12}. As i_3^{12} has a smallest position number in $J_r^3\cup\bar{J}_r^{12}$, by considering the idle time of M_3 caused by the processing of the sequence $\{i_3^{12}\cdots l_{r,3}\}$, $T_3(J_r^3)$ must be replaced by $\underline{T}_3(J_r^3)$ as in the next form: [cf. Fig. 3]

$$T_3(J_r^3) \geqq \underline{T}_3(J_r^3) = \max\begin{bmatrix} \max\begin{bmatrix} T_1(J_r^{12}) + m_{1,\, i_3^{12}}, \\ T_2(J_r^{12}) \end{bmatrix} + m_{2,\, i_3^{12}}, \\ T_3^*(J_r^3 - \{i_3^{12} \cdots l_{r,3}\}) \end{bmatrix} + \sum_{i \in \{i_3^{12} \cdots l_{r,3}\}} m_{3,\, i}, \tag{1.8}$$

where $\{i_3^{12} \cdots l_{r,3}\}$ denotes the subsequence of J_r^3 which begins from i_3^{12} and ends at $l_{r,3}$, but when i_3^{12} doesn't exist it means $\{i_3^{12} \cdots l_{r,3}\} \equiv l_{r,3}$ and $T_3^*(J_r^3 - \{i_3^{12} \cdots l_{r,3}\})$ which denotes the completion time on M_3 of a subsequence $J_r^3 - \{i_3^{12} \cdots l_{r,3}\}$ contained in $J_r^{12} \cap J_r^3$ can be calculated by using the relation (a) which is the same as equation (1.3) in sec. 1.1.1:

$$T_k(i_{j,\,k}) = \max\begin{bmatrix} T_{k-1}(i_{j,\,k}), \\ T_k(i_{j-1,\,k}) \end{bmatrix} + m_{k,\, i_{j,k}}, \tag{a}$$

where $T_k(i_{j,\,k})$ denotes the completion time on M_k of jth job $i_{j,\,k}$ of the sequence on M_k.

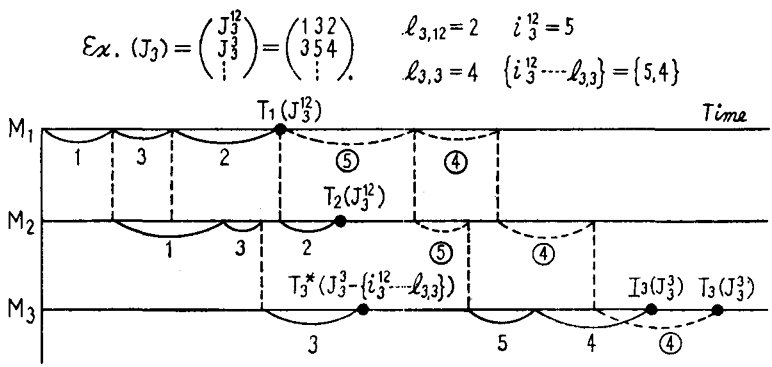

Fig. 3. Example of $\underline{T}_3(J_r^3)$ and $T_3(J_r^3)$ where $n=5$, $r=3$.

Next,, it holds next form for $T_4(J_r^4)$ by similar reasons as above:

$$T_4(J_r^4) \geqq \underline{T}_4(J_r^4) = \max\left(\begin{array}{l} \max\left[\begin{array}{l} \max\left[\begin{array}{l} T_1(J_r^{12}) + m_{1,\, i_4^{3,12}}, \\ T_2(J_r^{12}) \end{array}\right] + m_{2,\, i_4^{3,12}}, \\ \underline{T}_3(J_r^3) \end{array}\right] + m_{3,\, i_{4,12}^3}, \\ T_4^*(J_r^4 - \{i_4^3 \cdots l_{r,4}\}) \end{array}\right) + \sum_{i \in \{i_4^3 \cdots l_{r,4}\}} m_{4,\, i}, \qquad (r=1\sim n-1) \tag{1.9}$$

where they must be defined that if $i_4^3 = i_4^{3,12}$ then it holds $i_4^3 \equiv i_4^{3,12} \equiv i_{4,12}^3$ and if $i_4^3 \equiv i_{4,12}^3$ then it holds $m_{k,\, i_4^{3,12}} \equiv 0$ $(k=1,2)$ [cf. Fig. 4], and $T_4^*(J_r^4 - \{i_4^3 \cdots l_{r,4}\})$ having the same meaning as in (1.8) can be calculated by using (a).

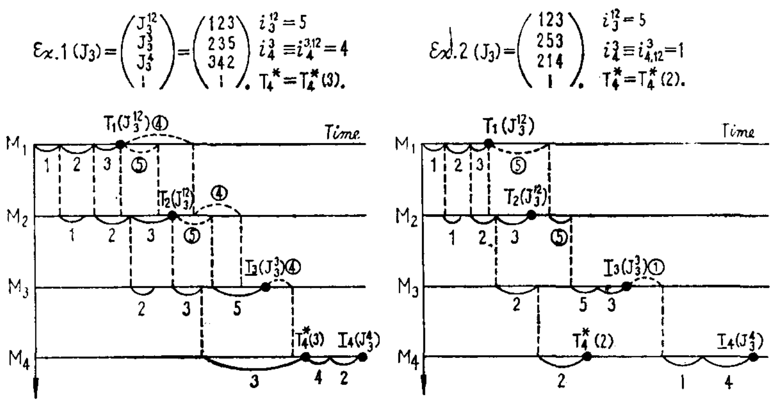

Fig. 4. Special Examples of $\underline{T}_4(J_r^4)$ where $n=5$, $r=3$.

Generally, by the same reasons, for determination of the value of $T_k(J_r^k)$ $(k=4\sim m-1,\ r=1\sim n-1)$, it holds next forms for each k and r:

$$T_k(J_r^k) \geqq \underline{T}_k(J_r^k) = \max\left[\begin{array}{l} T_{k,\,k-1} + m_{k-1,\, i_{k,k-2}^{k-1}}, \\ T_k^*(J_r^k - \{i_k^{k-1} \cdots l_{r,k}\}) \end{array}\right] + \sum_{i \in \{i_k^{k-1} \cdots l_{r,k}\}} m_{k,\, i}, \tag{1.10}$$

where maximum operations $T_{k,\,p}$ $(p=2\sim k-1)$ are defined as follows to simplify the term in maximum bracket of (1.10) and calculated for increasing p to obtain $T_{k,\,k-1}$,

$$
\left.
\begin{aligned}
T_{k,2} &\equiv \max\begin{bmatrix} T_1(J_r^{12}) + m_{1,\, i_k^{k-1,\cdots,3,12}}, \\ T_2(J_r^{12}) \end{bmatrix} \quad (p=2) \\
T_{k,p} &\equiv \max\begin{bmatrix} T_{k,\,p-1} + m_{p-1,\, i_{k,p-2}^{k-1,\cdots,p-1}}, \\ \underline{T}_p(J_r^p) \end{bmatrix} \quad (p=3\sim k-1)
\end{aligned}
\right\} \quad \text{(b)}
$$

and they must be defined that if $i_k^{k-1} \equiv i_{k,q-2}^{k-1,\cdots,q-1}$ then each $m_{p,\, i_{k,p-2}^{k-1,\cdots,p-1}} \equiv 0$ $(3 \leqq p \leqq q-1)$ and $i_k^{k-1} \equiv i_{k,q-2}^{k-1,\cdots,q-1} \equiv i_{k,p-2}^{k-1,\cdots,p-1}$ $(q+1 \leqq p \leqq k)$ and moreover $T_k^*(J_r^k - \{i_k^{k-1} \cdots l_{r,k}\})$ denotes the completion time on M_k of a sequence $J_r^k - \{i_k^{k-1} \cdots l_{r,k}\}$ of J_r^k which can be calculated by using (a) from temporarily determined completion time on M_{k-1} of each job of this sequence for calculation of $T_{k-1}(J_r^{k-1})$ by (1.10) or by (1.8) for $k=4$.

3. $T_m(J_r^{m-1,m})$. Lastly it holds

$$
T_m(J_r^{m-1,m}) \geqq \underline{T}_m(J_r^{m-1,m}) = \max\begin{bmatrix} \underline{T}_{m-1}(J_r^{m-1,m}), \\ T_m^*(J_r^{m-1,m} - l_{r,\,m-1,\,m}) \end{bmatrix} + m_{m,\, l_{r,\,m-1,\,m}}, \tag{1.11}
$$

where $T_m^*(J_r^{m-1,m} - l_{r,\,m-1,\,m})$ is calculated by using (a) as above.

1.2.2. Revised lower bound

Next, Johnson's eriterion for two machines is applied to each two machines M_k, M_{k+1} $(k=1\sim m-1)$ with the set of $(n-r)$ unordered remained jobs $\bar{J}_r^k, \bar{J}_r^{k+1}$ respectively.

Let they defined that

$$
\bar{J}_r\binom{k}{k+1} = \bar{J}_r^k \cap \bar{J}_r^{k+1}, \quad \bar{J}_r\binom{k}{-} = \bar{J}_r^k \cap J_r^{k+1}, \quad \bar{J}_r\binom{-}{k+1} = J_r^k \cap \bar{J}_r^{k+1},
$$

where $\bar{J}_r\binom{m-1}{-} \equiv \phi, \bar{J}_r\binom{-}{m} \equiv \phi$ $(k=m-1)$.

Then, all jobs in $\bar{J}_r\binom{k+1}{k}$ must be optimally ordered by Johnson's criterion (1.1) for two machines M_k, M_{k+1} as in sec. 1.1.1.

After the time $\underline{T}_k(J_r^k)$ on M_k, first all jobs in $\bar{J}_r\binom{k}{-}$ are processed along the same ordering as in J_r^{k+1}, and after the time $\underline{T}_{k+1}(J_r^{k+1})$ on M_{k+1}, first all jobs in $\bar{J}_r\binom{-}{k+1}$ are processed along the same ordering as in J_r^k, then after the processing of all jobs in $\bar{J}_r\binom{k}{-}$ on M_k and all jobs in $\bar{J}_r\binom{-}{k+1}$ on

M_{k+1}, a sequence of $\bar{J}_r\binom{k+1}{k}$ defined by Johnson's criterion is processed on M_k, M_{k+1}.

Then, let $T^k_{k+1}(\bar{J}^{k+1}_r)$ be the elapsed time of the processing of all jobs in $\bar{J}_r(\overline{k+1}), \bar{J}_r\binom{k}{k+1}$ after $\underline{T}_{k+1}(J^{k+1}_r)$ on M_{k+1}. [cf. Fig. 5]

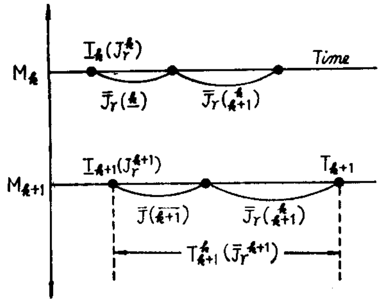

Fig. 5. Processing of $\bar{J}^{k+1}_r$ on M_{k+1}.

Let as shown in Fig. 5, completion time of these jobs on M_{k+1} be T_{k+1} then T_{k+1} is a possible earliest completion time of any sequence of all jobs in J^{k+1}_r on M_{k+1}.

Hence, it holds next revised lower bound for each node (J_r) $(r=1\sim n-1)$:

$$\mathrm{LB}(J_r)=\max\left\{\begin{array}{l} T_2(J_r^{12})+T_2^1(\bar{J}_r^{12})+\sum\limits_{k=3}^{m-2}\min\limits_{i\in\bar{J}_r{}^k} m_{k,i}+\min\limits_{i\in\bar{J}_r{}^{m-1,m}}\sum\limits_{k=m-1}^{m} m_{k,i}, \\ \underline{T}_3(J_r^3)+T_3^2(\bar{J}_r^3)+\sum\limits_{k=4}^{m-2}\min\limits_{i\in\bar{J}_r{}^k} m_{k,i}+\min\limits_{i\in\bar{J}_r{}^{m-1,m}}\sum\limits_{k=m-1}^{m} m_{k,i}, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \underline{T}_{m-2}(J_r^{m-2})+T_{m-2}^{m-3}(\bar{J}_r^{m-2})+\min\limits_{i\in\bar{J}_r{}^{m-1,m}}\sum\limits_{k=m-1}^{m} m_{k,i}, \\ \underline{T}_{m-1}(J_r^{m-1,m})+T_{m-1}^{m-2}(\bar{J}_r^{m-1,m})+\min\limits_{i\in\bar{J}_r{}^{m-1,m}} m_{m,i}, \\ \underline{T}_m(J_r^{m-1,m})+T_m^{m-1}(\bar{J}_r^{m-1,m}). \qquad (r=1\sim n-1) \end{array}\right\} \tag{1.12}$$

$\mathrm{LB}(J_r)$ is an increasing function of r for each $J_{r_1}\subset J_{r_2}$ $(r_1<r_2)$ and

LB(J_{n-1}) is equal to real total elapsed time TE(J_{n-1}) of a set of sequences of n jobs on each machines that is uniquely determined by J_{n-1}.

1.2.3. Comparison with lower bound in Ref. [4] and B. B. algorithms

Revised lower bound (1.12) is more exact than the lower bound already given [4] as follows:

$$\mathrm{LB}(J_r)=\max\left(\begin{array}{l} T_1(J_r^{12})+\sum_{i\in\bar{J}_r^{12}} m_{1,i}+\sum_{k=2}^{m-2}\min_{i\in\bar{J}_r^k} m_{k,i}+\min_{i\in\bar{J}_r^{m-1,m}}\sum_{k=m-1}^{m} m_{k,i}, \\ T_2(J_r^{12})+\sum_{i\in\bar{J}_r^{12}} m_{2,i}+\sum_{k=3}^{m-2}\min_{i\in\bar{J}_r^k} m_{k,i}+\min_{i\in\bar{J}_r^{m-1,m}}\sum_{k=m-1}^{m} m_{k,i}, \\ \underline{T}_3(J_r^3)+\sum_{i\in\bar{J}_r^3} m_{3,i}+\sum_{k=4}^{m-2}\min_{i\in\bar{J}_r^k} m_{k,i}+\min_{i\in\bar{J}_r^{m-1,m}}\sum_{k=m-1}^{m} m_{k,i}, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \underline{T}_{m-2}(J_r^{m-2})+\sum_{i\in\bar{J}_r^{m-2}} m_{m-2,i}+\min_{i\in\bar{J}_r^{m-1,m}}\sum_{k=m-1}^{m} m_{k,i}, \\ \underline{T}_{m-1}(J_r^{m-1,m})+\sum_{i\in\bar{J}_r^{m-1,m}} m_{m-1,i}+\min_{i\in\bar{J}_r^{m-1,m}} m_{m,i}, \\ \underline{T}_m(J_r^{m-1,m})+\sum_{i\in\bar{J}_r^{m-1,m}} m_{m,i}, \end{array}\right)$$

$$(r=1\sim n-1) \qquad (1.13)$$

That is to say, since $T_{k+1}^k(\bar{J}_r^{k+1}\geqq\sum_{i\in\bar{J}_r^{k+1}} m_{k+1,i}$, if it's neglected the first term in maximum bracket of (1.13), LB(J_r) of (1.12) is larger than that of (1.13) and more efficient in the sense that $T_{k+1}^k(\bar{J}_r^{k+1})$ characterize a possible minimum sum of the idle time of M_{k+1} caused by processing of any backsequence of unordered remained $(n-r)$ jobs in $\bar{J}_r^{k+1}$ on machine M_{k+1} $(k=1\sim n-1)$ and the calculations of the value of $T_{k+1}^k(\bar{J}_r^{k+1})$ which is the same as that of $\underline{T}_{k+1}(J_r^{k+1})$ say isn't so complicated.

In the next section, it will be shown this facts by solving an example using B. B. alyorithm with lower bound (1.12) and (1.13) respectively.

Branch and bound algorithms.

Five B. B. algorithms has presented in former paper [4]. Principal algorithms are as follows:

Algorithm 1. The procedure is to start from a node (J_0) representing all possible sequences of the amounts $(n!)^{m-2}$, then to devide this node into n^{m-2} subclasses (nodes (J_1)) according to whether the first job of J_1^{12} and J_1^k $(k=3\sim m-2)$ and $J_1^{m-1,m}$ is $1, 2, \cdots, n$. And for each of them, by using (1.7)~(1.12), (a), (b), LB(J_1) is calculated and one of the nodes (J_1) having minimum LB(J_1) is devided into $(n-1)^{m-2}$ subclasses (nodes (J_2)) having the same first job as this node (J_1) on each machine, according to whether the second job of J_1^{12} and J_2^k $(k=3\sim m-2)$ and $J_2^{m-1,m}$ is $1, 2, \cdots n$ except the same number as this branched node (J_1). And then LB(J_2) is calculated for each of them as above and one of the nodes (J_2) having minimum LB(J_2) is devided into $(n-2)^{m-2}$ subclasses (nodes (J_3)) having the same former two jobs as this node (J_2) on each machine, according to whether the third job of J_3^{12} and J_3^k $(k=3\sim m-2)$ and $J_3^{m-1,m}$ is $1, 2, \cdots, n$ except the same numbers as this branched node (J_2).

Proceeding by the same way to the nodes (J_{n-1}) of the amounts 2^{m-2}, LB(J_{n-1}) of each of them is calculated as above. Let MLB(J_{n-1}) denotes the minimum of these LB(J_{n-1}), then already formed nodes (J_r) $(r=1\sim n-1)$ of order one for which it holds inequality LB(J_r) $\geqq$ MLB(J_{n-1}) are discarded.

In this situation if there are no nodes (J_r) $(r=1\sim n-1)$ such that LB(J_r) $<$ MLB(J_{n-1}), then a set of sequences of n jobs on each machine that is uniquely determined by J_{n-1} which gives MLB(J_r) is an optimal solution, being LB(J_{n-1})$\equiv$TE(J_{n-1}). Othermise, the same proredure as above is applied to the remaning nodes of order one by branching a node having largest r among the minimum of their LB(J_r).

By proceeding by this way, finally it can be found a node (J_{n-1}) having the minimum of LB(J_r) among the remaining nodes (J_r) $(r=1\sim n-1)$ of order one and a set of sequences uniquely determined by this node is an optimal solution.

Algorithm 2. (algorithm 3. in Ref. [4])

The procedure that is different from algorithm 1 is to replace the term $T_k^*(J_r^k-\{i_k^{k-1}\cdots l_{r,k}\})$ by the term $\underline{T}_k(J_r^k-\{i_k^{k-1}\cdots l_{r,k}\})$ in revised lower bound $\mathrm{LB}(J_r)$ $(k=3\sim m, r=1\sim n-1)$ where each value of $\underline{T}_k(J_r^k-\{i_k^{k-1}\cdots l_{r,k}\})$ is known from the result of the calculations of the lower bound of a former node connected with this node (J_r), having $J_r^k-\{i_k^{k-1}\cdots l_{r,k}\}$ as presubsequence on M_k, and to calculate $\mathrm{TE}(J_{n-1})$ by using (a) for a node (J_{n-1}) having minimum $\mathrm{LB}(J_{n-1})$ obtained by the same way as in algorithm 1 and to discard all nodes (J_r) $(r=1\sim n-1)$ of order one having $\mathrm{LB}(J_r)\geqq\mathrm{TE}(J_{n-1})$ in the case when it holds $\mathrm{TE}(J_{n-1})\leqq\mathrm{LB}(J_{n-1})$ for each node of the remained nodes (J_{n-1}) on the same branch as a node having minimum $\mathrm{LB}(J_{n-1})$, or otherwise to calculate $\mathrm{TE}(J_{n-1})$ of some nodes (J_{n-1}) having smaller $\mathrm{LB}(J_{n-1})$ than the firstly calculated $\mathrm{TE}(J_{n-1})$ and to determine the least $\mathrm{TE}(J_{n-1})$ among them in order to compare with $\mathrm{LB}(J_r)$ for each of all remained nodes (J_r) $(r=1\sim n-1)$ of order one, and to follow the saıne steps untill a node (J_{n-1}) determining an optimal solution is found as in algorithm 1.

Practically it may be more efficient to use algarithm 2 than the other if balance between the quantity of calculations of the value of lower bound $\mathrm{LB}(J_r)$ for each node (J_r) and the number of nodes of scheduling tree is taken into consideration.

1.2.4. Numerical example

In this section, algorithm 2 with revised lower bound (1.12) and lower bound (1.13) already given in Ref. [4] are applied to four machines case respectively. For this case, earliest completion time $T_k(J_r^{12})$ $(k=1,2)$, $T_k(J_r^{34})$ $(k=3,4)$ of a subsequence J_r^{12}, J_r^{34} on machines M_1 and M_2, M_3 and M_4 respectively become next forms:

$$\begin{cases} T_1(J_r^{12})=\sum_{i\in J_r} m_{1,i}, \\ T_2(J_r^{12})=\max\begin{bmatrix} T_1(J_r^{12}), \\ T_2(J_r^{12}-l_{r,12}) \end{bmatrix}+m_{2,l_{r,12}} \end{cases} \tag{1.14}$$

$$T_3(J_r^{34}) \geqq \underline{T}_3(J_r^{34}) = \max\begin{bmatrix} \max\begin{bmatrix} T_1(J_r^{12})+m_{1,\, i_{34}^{12}}, \\ T_2(J_r^{12}) \end{bmatrix} + m_{2,\, i_{34}^{12}}, \\ \underline{T}_3(J_r^{34} - \{i_{34}^{12} \cdots l_{r,\,34}\}) \end{bmatrix} + \sum_{i \in \{i_{34}^{12} \cdots l_{r\cdot 34}\}} m_{3,\, i} \tag{1.15}$$

$$T_4(J_r^{34}) \geqq \underline{T}_4(J_r^{34}) = \max\begin{bmatrix} \underline{T}_3(J_r^{34}), \\ \underline{T}_4(J_r^{34} - l_{r,\,34}) \end{bmatrix} + m_{4,\, l_{r,34}}. \tag{1.16}$$

And revised lower bound $\mathrm{LB}(J_r)$ of a node (J_r) of a tree is as follows:

$$\underset{(r=1\sim n-1)}{\mathrm{LB}(J_r) = \max}\begin{pmatrix} T_2(J_r^{12}) + T_2^1(\bar{J}_r^{12}) + \min\limits_{i \in \bar{J}_r^{34}} \sum\limits_{p=3} m_{p,\, i}, \\ \underline{T}_3(J_r^{34}) + T_3^2(\bar{J}_r^{34}) + \min\limits_{i \in \bar{J}_r^{34}} m_{4,\, i}, \\ \underline{T}_4(J_r^{34}) + T_4^3(\bar{J}_r^{34}). \end{pmatrix} \tag{1.17}$$

This $\mathrm{LB}(J_r)$ is an increasing function of r for $J_{r_1} \subset J_{r_2}$ $(r_1 < r_2)$ and it holds $\mathrm{LB}(J_{n-1}) \leqq \mathrm{TE}(J_{n-1})$.

Following the algorithm 2, next example can be easily solved for each of two lower bounds.

Example $(m=4,\ n=4)$. [4].
Processing Time (hrs.)

i	1	2	3	4
$m_{1,i}$	4	5	4	6
$m_{2,i}$	3	4	6	1
$m_{3,i}$	4	4	14	3
$m_{4,i}$	7	1	4	8

Scheduling tree by the procedure of the algorithm 2 with revised lower bound becomes the next form [Fig. 6] having 42 nodes smaller than 46 nodes by lower bound already given [4]. Upper number and lower number in parenthesis labeled at each node denotes revised lower bound and old lower bound respectively.

In this example each $\mathrm{LB}(J_3)$ $(n-1=3)$ has just coincided with $\mathrm{TE}(J_3)$ calculated by using (a).

Examples of the calculations of $\mathrm{LB}(J_r)$ are shown below for nodes $\mathrm{LB}(\begin{smallmatrix}1\\3\end{smallmatrix})$, $\mathrm{LB}(\begin{smallmatrix}13\\14\end{smallmatrix})$.

LB$\binom{1}{3}$.

Job	1	3	2	4		
M_1	4	8	13	19		
M_2	7	14	18	20		$20+5=25$
Job	1	(3)				
M_2	7	14		4	2	
Job			(1)	15	19	
M_3		28	32	35	39	$39+1=40$
Job	3	4	1	2		
M_3	28	31	35	39		$\mathrm{LB}\binom{1}{3} = \max\begin{bmatrix}25\\40\\48\end{bmatrix}=48$
M_4	32	40	47	48		

LB$\binom{13}{14}$.

Job	1	3	2	4		
M_1	4	8	13	19		
M_2	7	14	18	20		$20+5=25$
Job	1	3	(4)		2	
M_1	4	8	14			
M_2	7	14	15		19	
Job	1		4	3	2	
M_3	11		18	32	36	$36+1=37$
Job	1	4	3	2		
M_3	11	18	32	36		$\mathrm{LB}\binom{13}{14} = \max\begin{bmatrix}25\\37\\37\end{bmatrix}=37.$
M_4	18	26	36	37		

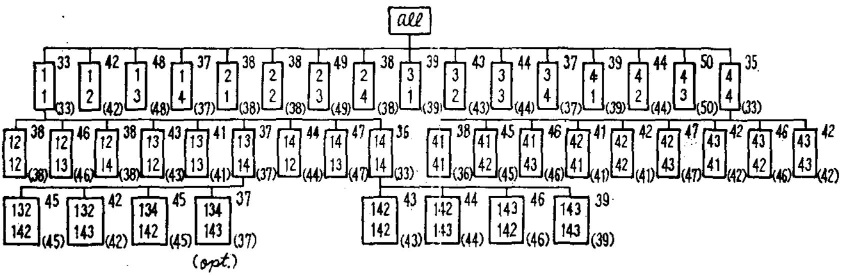

Fig. 6. Scheduling Tree of the Example.

A node $(J_3)=\binom{134}{143}$ having $\mathrm{LB}(J_3)\equiv\mathrm{TE}(J_3)=37$ gives an optimal solution with a sequence (1342) on M_1 and M_2 and a sequence (1432) on M_3 and M_4 and minimal total elapsed time is 37 hrs..

§ 2. Other types of lower bound in case where no passing is allowed

In this section, only the case where no passing is allowed will be considered. And, for the purpose of estimating a better lower bound of makespans, two types of lower bound under definite backsubsequence and under definite pre—and back-subsequences respectively are constructed.

2.1. Sequences with definite backsubsequence

Let LB(R_r) be the lower bound of makespans of sequences with definite backsubsequence R_r of r jobs ($r=1\sim n-1$) and $\overline{R}_r$ be the set of unordered remained $(n-r)$ jobs.

As in sec. 1, let $i_1^k\, i_2^k \cdots i_{n-r}^k$ be a sequence of all jobs in $\overline{R}_r$ which is determined by Johnson's criterion (1.1) for two machines M_k, M_{k+1} ($k=1\sim m-1$), then after the time $T_k^0 \equiv \min_{i\in \overline{R}_r} \sum_{p=1}^{k-1} m_{p,i}$ ($T_1^0\equiv 0$), on each M_k ($k=1\sim m-1$), the above sequence $i_1^k\, i_2^k \cdots i_{n-r}^k$ is processed on M_k and M_{k+1} such that idle time of M_k desn't exist.

Next, let $T_{k+1}^k(\overline{R}_r)$ be the elapsed time of the processing of this sequence on M_{k+1} after the time T_k^0. [cf. Fig. 7]

Let E_{k+1}^{n-r} be the completion time on M_{k+1} of a last job i_{n-r}^k of this sequence as shown in Fig. 7. Next, after the time $T_k^{n-r} \equiv \min_{i\in\overline{R}_r} \sum_{p=1}^{k-1} m_{p,i} + \sum_{q=1}^{n-r} m_{k,i_q^k}$ on M_k, definite backsubsequence R_r must be processed on M_k, $M_{k+1}, \cdots, M_m$; that is, each job of R_r must be processed on M_k continuously without idle time of M_k and then each job of R_r must be processed on the following machines $M_{k+1}, M_{k+2}, \cdots, M_m$ by using (a) such that the first job of R_r must be processed on M_{k+1} after the time E_{k+1}^{n-r}, on each M_q ($q=k+2\sim m$) after the time $E_{k+1}^{n-r} + \sum_{p=k+2}^{q} \min_{i\in\overline{R}_r} m_{p,i}$.

Let $E_m{}^k(R_r)$ be the elapsed time on M_m from the time $F^{n-r}_{k,m} \equiv E^{n-r}_{k+1} + \sum_{p=k+2}^{m} \min_{i \in \bar{R}_r} m_{p,\,i}$ to the completion time $G_m{}^k(R_r)$ of the last job of R_r.

Then, completion time on M_m of the last job of any sequence of n jobs with definite backsubsequence R_r isn't earlier than the time $G_m{}^k(R_r)$.

Hence, next lower bound $\mathrm{LB}(R_r)$ of the makespans of sequences with definite backsubsequence R_r is obtained:

$$
\begin{cases}
\mathrm{LB}(R_r) = \max \left(\begin{array}{l}
T_2{}^1(\bar{R}_r) + \sum\limits_{p=3}^{m} \min\limits_{i \in \bar{R}_r} m_{p,\,i} + E_m{}^1(R_r), \\[2ex]
\min\limits_{i \in \bar{R}_r} m_{1,\,i} + T_3{}^2(\bar{R}_r) + \sum\limits_{p=4}^{m} \min\limits_{i \in \bar{R}_r} m_{p,\,i} + E_m{}^2(R_r), \\[2ex]
\min\limits_{i \in \bar{R}_r} \sum\limits_{p=1}^{2} m_{p,\,i} + T_4{}^3(\bar{R}_r) + \sum\limits_{p=5}^{m} \min\limits_{i \in \bar{R}_r} m_{p,\,i} + E_m{}^3(R_r), \\[1ex]
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\[1ex]
\min\limits_{i \in \bar{R}_r} \sum\limits_{p=1}^{m-3} m_{p,\,i} + T^{m-2}_{m-1}(\bar{R}_r) + \min\limits_{i \in \bar{R}_r} m_{m,\,i} + E^{m-2}_{m}(R_r), \\[2ex]
\min\limits_{i \in \bar{R}_r} \sum\limits_{p=1}^{m-2} m_{p,\,i} + T^{m-1}_{m}(\bar{R}_r) + E^{m-1}_{m}(R_r).
\end{array} \right) \\
\qquad (r = 1 \sim n-2) \\[2ex]
\mathrm{LB}(R_{n-1}) = T_2{}^1(i_1) + \sum\limits_{p=3}^{m} \min\limits_{i \in \bar{R}_{n-1}} m_{p,\,i} + E_m{}^1(R_{n-1}) \quad \text{[First term in max. bracket]} \\[2ex]
\qquad\qquad \equiv T_2{}^1(i_1) + \sum\limits_{p=3}^{m} m_{p,\,i_1} + E_m{}^1(R_{n-1}), \\[1ex]
\text{where } \bar{R}_{n-1} \equiv i_1.
\end{cases}
\tag{2.1}
$$

Another lower bound can be obtained by using only the first term in maximum bracket of $\mathrm{LB}(R_r)\,(r=1 \sim n-2)$; that is,

$$
\mathrm{LB}(R_r) = T_2{}^1({}_r\bar{R}) + \sum_{p=3}^{m} \min_{i \in \bar{R}_r} m_{p,\,i} + E_m{}^1(R_r) \quad (r = 1 \sim n-1)
\tag{2.2}
$$

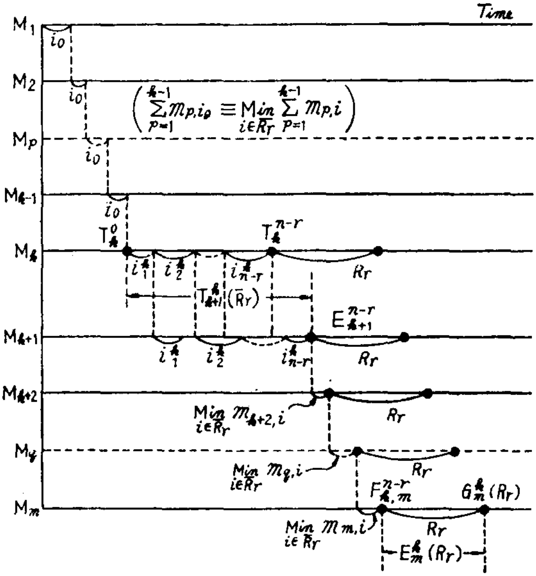

Fig. 7.

Always it holds (2.2) $\leqq$ (2.1). Each of the above two lower bounds (2.1), (2.2) is an increasing function of r where $R_{r_1} \subset R_{r_2}$ $(r_1 \subset r_2)$ and $\mathrm{LB}(R_{n-1})$ is equal to the real total elapsed time $\mathrm{TE}(R_{n-1})$ of a sequence which is uniquely determined by R_{n-1}.

Moreover, another less exact lower bound in comparison with lower bound (2.1) can be obtained as below by similar constructions as that of lower bound $\mathrm{LB}(J_r)$ (1.5) in sec. 1.1.2.

As in the case of lower bound (2.1), after the time $T_k^{n-r} \equiv \min_{i\in\bar{R}_r} \sum_{p=1}^{k-1} m_{p,i} + \sum_{i\in\bar{R}_r} m_{k,i}$ on each M_k $(k=1\sim m)$, definite backsubsequence R_r must be processed on $M_k, M_{k+1}, \cdots, M_m$ by the same ways as to lower bound (2.1) [cf. Fig. 8], such that the first job of R_r must be processed on M_k after the time T_k^{n-r} continuously and on M_q $(q=k+1\sim m)$ after the time $T_k^{n-r} + \sum_{p=k+1}^{q} \min_{i\in\bar{R}_r} m_{p,i}$. Let the time $F_{k,m}^{n-r}$ be defined as below: $F_{k,m}^{n-r} \equiv$

$T_k^{n-r}+\sum_{p=k+1}^{m}\min_{i\in\bar{R}_r} m_{p,i}$, then let $\underline{E}_m{}^k(R_r)$ be the elapsed time on M_m from the time $\underline{F}_{k,m}^{n-r}$ to the completion time $\underline{G}_m{}^k(R_r)$ of the last job of R_r where it's holds that the time $\underline{G}_m{}^k(R_r)$ isn't larger than the time $G_m{}^k(R_r)$ in Fig. 7.

Hence next simple lower bound can be obtained:

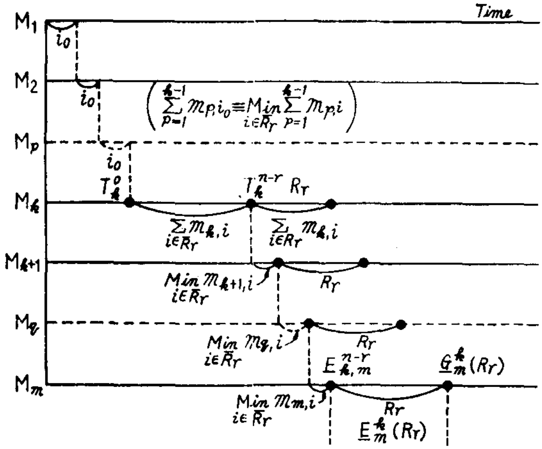

Fig. 8.

$$\mathrm{LB}(R_r)=\max\left(\begin{array}{l}
\sum_{i\in R_r} m_{1,i}+\sum_{p=2}^{m}\min_{i\in\bar{R}_r} m_{p,i}+\underline{E}_m{}^1(R_r),\\[2ex]
\min_{i\in\bar{R}_r} m_{1,i}+\sum_{i\in\bar{R}_r} m_{2,i}+\sum_{p=3}^{m}\min_{i\in\bar{R}_r} m_{p,i}+\underline{E}_m{}^2(R_r),\\[2ex]
\min_{i\in\bar{R}_r}\sum_{p=1}^{2} m_{p,i}+\sum_{i\in\bar{R}_r} m_{3,i}+\sum_{p=4}^{m}\min_{i\in\bar{R}_r} m_{p,i}+\underline{E}_m{}^3(R_r),\\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
\min_{i\in\bar{R}_r}\sum_{p=1}^{m-2} m_{p,i}+\sum_{i\in\bar{R}_r} m_{m-1,i}+\min_{i\in\bar{R}_r} m_{m,i}+\underline{E}_m^{m-1}(R_r),\\[2ex]
\min_{i\in\bar{R}_r}\sum_{p=1}^{m-1} m_{p,i}+\sum_{i\in\bar{R}_r} m_{m,i}+\sum_{i\in\bar{R}_r} m_{m,i}.
\end{array}\right)$$

$(r=1\sim n-1)$ (2.3)

Also next more simple lower bound which is the first term in maximum bracket of the lower bound (2.3) can be obtained:

$$\mathrm{LB}(R_r)=\sum_{i\in\bar{R}_r} m_{1,i}+\sum_{p=2}^{m}\min_{i\in\bar{R}_r} m_{p,i}+\underline{E}_m{}^1(R_r). \quad (r=1\sim n-1) \tag{2.4}$$

Each of $\mathrm{LB}(R_r)$ (2.3), (2.4) is an increasing function of r where $R_{r_1}\subset R_{r_2}$ $(r_1<r_2)$ and $\mathrm{LB}(R_{n-1})$ is equal to the real total elapsed time $\mathrm{TE}(R_{n-1})$ of a sequence of n jobs uniquely determined by R_{n-1}.

Remark: The other lower bound $\mathrm{LB}(R_r)$ can be obtained if the first term in maximum bracket of the lower bound (2.1) or (2.3) is contained in its maximum bracket.

2.1.2. B. B. algorithm

B. B. algorithm can be constructed along similar procedures as the algorithm in sec. 1.1.2 by determining one job upward from last job of the backsequence at each step.

2.1.3. Numerical example

Next example ($m=3, n=6$) will be solved by B. B. algorithm with lower bound $\mathrm{LB}(R_k)$ (2.1).

Example

Processing Time (hrs.)

i	1	2	3	4	5	6
$m_{1,i}$	3	3	8	7	10	12
$m_{2,i}$	7	2	6	11	8	14
$m_{3,i}$	6	12	4	3	6	2

In three machines case, lower bound $\mathrm{LB}(R_r)$ (2.1) becomes next forms:

$$\mathrm{LB}(R_r) = \max_{(r=1\sim 4)} \begin{bmatrix} T_2{}^1(\overline{R}_r)+\min\limits_{i\epsilon\overline{R}_r} m_{3,\,i}+E_3{}^1(R_r), \\ \min\limits_{i\epsilon\overline{R}_r} m_{1,\,i}+T_3{}^2(\overline{R}_r)+E_3{}^2(R_r). \end{bmatrix}$$

$$\mathrm{LB}(R_5)=T_2{}^1(i_1)+m_{3,\,i_1}+E_3{}^1(R_5)$$
$$=\sum_{p=1}^{3} m_{p,\,i_1}+E_3{}^1(R_5).\quad (\overline{R}_5\equiv i_1)$$

In this example, a sequence determined by Johnson's criterion (1.1) is 146532 for M_1 and M_2 and 215346 for M_2 and M_3. Then, examples of the calculations of the value of $\mathrm{LB}(R_r)$ are as follows:

First, for LB(3), $i_1{}^1\, i_2{}^1\, i_3{}^1\, i_4{}^1\, i_5{}^1=14652$ and $i_1{}^2\, i_2{}^2\, i_3{}^2\, i_4{}^2\, i_5{}^2=21546$

Job	1	4	6	5	2	3
M_1	3	10	22	32	35	43
M_2	10	21	36	44	46	52
M_3			46+2=48			56

M_1 3 Job	2	1	5	4	6	3
M_2	5	12	20	31	45	51
M_3	17	23	29	34	47	55

LB(3)=max [56, 55]=56

Next, for LB(53), $i_1{}^1\, i_2{}^1\, i_3{}^1\, i_4{}^1=1462$ and $i_1{}^2\, i_2{}^2\, i_3{}^2\, i_4{}^2=2146$

Job	1	4	6	2	5	3
M_1	3	10	22	25	35	43
M_2	10	21	36	38	46	52
M_3			38+2=40		52	56

M_1 3 Job	2	1	4	6	5	3
M_2	5	12	23	37	45	51
M_3	17	23	26	39	51	55

LB(53)=max [56, 55]=56

By the same ways, scheduling tree can be constructed as in Fig. 9 where number rabeled at each node denotes the lower bound (2.1) of each node.

Optimal sequence is 142653 and 214653, 124653 (57 hrs.) further obtained if a node (4653) may be branched. Total number of the nodes of this tree is a possible least number 20.

Remarks

(1). If lower bound (2.3) is used, then inspite of its simpler calculations of the value of $\mathrm{LB}(R_r)$, number of nodes in tree will much more increase.
(2). The above optimal sequences have definite backsubsequence 653 in common, hence, for this example lower bound $\mathrm{LB}(R_r)$ is more efficient

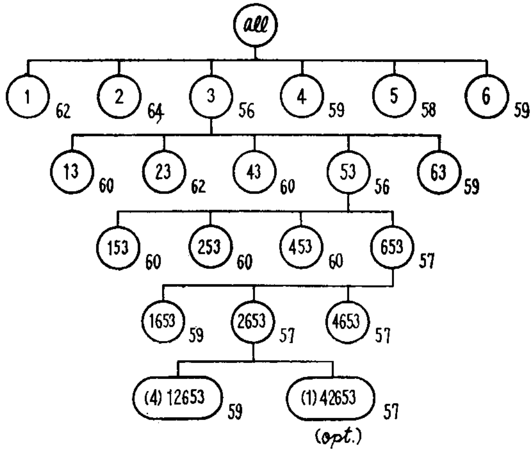

Fig. 9. Scheduling Tree of the Example.

than lower bound LB(J_r) in sec. 1.1.1. But lower bound LB(J_r) is more efficient than lower bound LB(R_r) for example 1 in sec. 1.1.3 on the other hand.

2.2. Sequence with definite pre-and back-subsequences

As mentioned in the remark (2) in former section, either one among the two lower bound LB(J_r), LB(R_r) is more efficient according to the type of the example. So that, construction of the lower bound LB(J_r, R_S) of the makespans of sequences with definite presubsequence J_r and definite backsubsequence R_S may have some meaning.

2.2.1. Lower bound LB(J_r, R_S)

The above lower bound LB(J_r, R_S) $(r+S=1\sim n-1)$ is constructed as follows:

Let S_{n-r-S} be the set of unordered remained $(n-r-S)$ jobs. First for the case where $rS \neq 0$, let $T_k(J_r)$ be the completion time of presubsequence J_r on M_k $(k=1\sim m)$ and a sequence $i_1^k\, i_2^k \cdots i_{n-r-S}^k$ be a sequence of all jobs in S_{n-r-s} which is determined by Johnson's criterion (1.1) for two machines M_k, M_{k+1} $(k=1\sim m-1)$.

Then, after the time $T_k(J_r)$ on each M_k $(k=1\sim m-1)$, the above sequence $i_1^k\, i_2^k \cdots i_{n-r-s}^k$ is processed on M_k, M_{k+1} such that idle time of M_k doesn't exist and let $T_{k+1}^k(S_{n-r-s})$ be the elapsed time of the processing of this sequence on M_{k+1} after the time $T_{k+1}(J_r)$ and the completion time of this sequence on M_{k+1} be E_{k+1}^{n-s} [cf. Fig. 10].

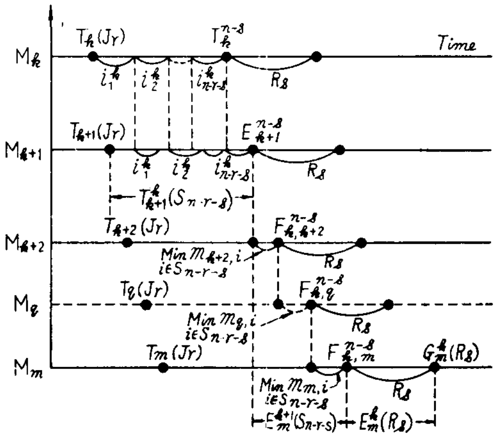

Fig. 10.

Next, after the time $T_k^{n-s} \equiv T_k(J_r) + \sum_{p=1}^{n-r-S} m_{k,\, i_p^k}$ on M_k, definite back-subsequence R_s must be processed on $M_k, M_{k+1}, \cdots, M_m$: that is, each job of R_s must be processed on M_k continuously without idle time of M_k and then each job of R_s must be processed on the following machines $M_{k+1}, M_{k+2}, \cdots, M_m$ successively such that the first job of R_s is processed on M_{k+1} after the time E_{k+1}^{n-s} and on M_{k+2} after the time $F_{k,k+2}^{n-s} \equiv \max[E_{k+1}^{n-s}, T_{k+2}(J_r)] + \min_{i \in S_{n-r-s}} m_{k+2,\, i}$ and on each M_q $(q=k+3\sim m)$ after the time $F_{k,q}^{n-s} \equiv \max[F_{k,q-1}^{n-s}, T_q(J_r)] + \min_{i \in S_{n-r-s}} m_{q,\, i}$ successively. Then, let $E_m^{k+1}(S_{n-r-s})$ be the elapsed time from the time E_{k+1}^{n-s} on M_{k+1} to the time $F_{k,m}^{n-s}$ on M_m

and $E_m^k(R_s)$ be the elapsed time on M_m after the time $F_{k,m}^{n-s}$ to the completion time $G_m^k(R_s)$ of the last job of R_s.

So, completion time on M_m of the last job of any sequence of n jobs with definite presubsequence J_r and definite backsubsequence R_3 isn't earlier than the time $G_m^k(R_s)$.

If $r=0$, then it's defined that $\mathrm{LB}(J_0, R_s)\equiv\mathrm{LB}(R_s)$: lower bound in sec. 2.1.1.

And if $s=0$, then it's defined that $\mathrm{LB}(J_r, R_0)\equiv\mathrm{LB}(J_r)$: revised lower bound in sec. 1.1.1.

Hence, next lower bound $\mathrm{LB}(J_r, R_s)$ can be obtained:

$$\left\{\begin{array}{l}
\mathrm{LB}(J_r, R_s)=\max\left(\begin{array}{l}
T_2(J_r)+T_2^1(S_{n-r-s})+E_m^2(S_{n-r-s})+E_m^1(R_s),\\
T_3(J_r)+T_3^2(S_{n-r-s})+E_m^3(S_{n-r-s})+E_m^2(R_s),\\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
T_{m-1}(J_r)+T_{m-1}^{m-2}(S_{n-r-s})+E_m^{m-1}(S_{n-r-s})+E_m^{m-2}(R_s),\\
T_m(J_r)+T_m^{m-1}(S_{n-r-s})+E_m^{m-1}(R_s).
\end{array}\right)\\
(r+s=1\sim n-2,\ rs\neq 0)\\
\mathrm{LB}(J_r, R_s)=T_2(J_r)+T_2^1(i_1)+\sum_{p=3}^{m} m_{p,i_1}+E_m^1(R_s):\ \text{first term in max.}\\
\quad\text{bracket,}\ (r+s=n-1, rs\neq 0)\quad \text{where}\ S_1\equiv i_1.\\
\mathrm{LB}(J_r, R_0)\equiv\mathrm{LB}(J_r)\ (1.2)\ (s=0)\\
\mathrm{LB}(J_0, R_s)\equiv\mathrm{LB}(R_s)\ (2.1)\ (r=0)
\end{array}\right. \qquad (2.5)$$

Another lower bound $\mathrm{LB}(J_r, R_s)$ can be obtained by using only the first term in maximum bracket of (2.5):

$$\left.\begin{array}{l}
\mathrm{LB}(J_r, R_s)=T_2(J_r)+T_2^1(S_{n-r-s})+E_m^2(S_{n-r-s})+E_m^1(R_s).\\
\qquad\qquad\qquad\qquad (r+s=1\sim n-1,\ rs\neq 0)\\
\mathrm{LB}(J_r, R_0)\equiv\mathrm{LB}(J_r)\ (1.2),\ \mathrm{LB}(J_0, R_s)\equiv\mathrm{LB}(R_s)\ (2.1).
\end{array}\right\} \qquad (2.6)$$

Moreover, another less exact lower bound LB(J_r, R_s) in comparison with lower bound (2.5) can be obtained as below by similar constructions as that of lower bound LB(J_r) (1.5) and LB(R_r) (2.3): [cf. Fig. 11]

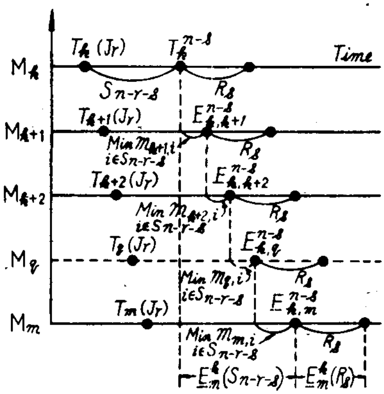

Fig. 11.

As in the case of lower bound (2.5), after the time $T_k^{n-s} \equiv T_k(J_r) + \sum_{i \in S_{n-r-s}} m_{k,i}$ on each M_k ($k=1 \sim m$), definite backsubsequence R_s must be processed on $M_k, M_{k+1}, \cdots, M_m$, that is, each job of R_s must be processed on M_k continuously without idle time of M_k and then each job of R_s must be processed on the following machines $M_{k+1}, M_{k+2}, \cdots, M_m$ successively such that the first job of R_s is processed on M_{k+1} after the time $\underline{F}_{k,k+1}^{n-s} \equiv \max[T_k^{n-s}, T_{k+1}(J_r)] + \min_{i \in S_{n-r-s}} m_{k+1,i}$ and on each M_q ($q=k+2 \sim m$) after the time $\underline{F}_{k,q}^{n-s} \equiv \max[\underline{F}_{k,q-1}^{n-s}, T_q(J_r)] + \min_{i \in S_{n-r-s}} m_{q,i}$.

Then, let $\underline{E}_m{}^k(S_{n-r-s})$ be the elapsed time after the time T_k^{n-s} on M_k to the time $\underline{F}_{k,m}^{n-s}$ on M_m and $\underline{E}_m{}^k(R_s)$ be the elapsed time on M_m after the time $\underline{F}_{k,m}^{n-s}$ to the completion time $\underline{G}_m{}^k(R_s)$ of the last job of R_s.

Hence, next lower bound can be obtained:

$$\left\{\begin{array}{l} \mathrm{LB}(J_r, R_s)=\max\left(\begin{array}{l} T_1(J_r)+\sum\limits_{i\in S_{n-r-s}} m_{1,i}+\underline{E}_m{}^1(S_{n-r-s})+\underline{E}_m{}^1(R_s), \\ T_2(J_r)+\sum\limits_{i\in S_{n-r-s}} m_{2,i}+\underline{E}_m{}^2(S_{n-r-s})+\underline{E}_m{}^2(R_s), \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ T_{m-1}(J_r)+\sum\limits_{i\in S_{n-r-s}} m_{m-1,i}+\underline{E}_m^{m-1}(S_{n-r-s})+\underline{E}_m^{m-1}(R_s), \\ T_m(J_r)+\sum\limits_{i\in S_{n-r-s}} m_{m,i}+\sum\limits_{i\in R_s} m_{m,i}. \end{array}\right) \\ (r+s=1\sim n-2 \\ rs\neq 0) \\ \mathrm{LB}(J_r, R_s)=T_1(J_r)+m_{1,i_1}+\underline{E}_m{}^1(i_1)+\underline{E}_m{}^1(R_s).\ (r+s=n-1,\ rs\neq 0,\ S_1\equiv i_1) \\ \mathrm{LB}(J_r, R_0)\equiv \mathrm{LB}(J_r)\ (1.5)\ (s=0),\ \mathrm{LB}(J_0, R_s)\equiv \mathrm{LB}(R_s)\ (2.3)\ (r=0). \end{array}\right. \tag{2.7}$$

Another lower bound can be obtained by using only the first term in maximum bracket of (2.7): that is,

$$\left.\begin{array}{l} \mathrm{LB}(J_r, R_s)=T_1(J_r)+\sum\limits_{i\in S_{n-r-s}} m_{1,i}+\underline{E}_m{}^1(S_{n-r-s})+\underline{E}_m{}^1(R_s). \\ \qquad\qquad (r+s=1\sim n-1,\ rs\neq 0) \\ \mathrm{LB}(J_r, R_0)\equiv \mathrm{LB}(J_r)\ (1.5),\ \mathrm{LB}(J_0, R_s)\equiv \mathrm{LB}(R_s)\ (2.3). \end{array}\right\} \tag{2.8}$$

Each of the above lower bounds $\mathrm{LB}(J_r, R_s)$ (2.5)~(2.8) is an increasing function of r and s where $J_{r_1}\subset J_{r_2}$ $(r_1<r_2)$ and $R_{s_1}\subset R_{s_2}$ $(s_1<s_2)$ and $\mathrm{LB}(J_r, R_s)$ $(r+s=n-1)$ is equal to the real total elapsed time $\mathrm{TE}(J_r, R_s)$ of a sequence uniquely determined by this subsequence (J_r, R_s) $(r+s=n-1)$.

2.2.2. B.B. algorithm

The procedure is almost the same as that of the B. B. algorithms in the above sections, except that; [I]. nodes $(J_1), (J_1, R_1), (J_2, R_1), (J_2, R_2), \cdots, (J_r, R_s)$ $(r+s=n-1)$ or [II]. nodes $(R_1), (J_1, R_1), (J_1, R_2), (J_2, R_2)\cdots, (J_r, R_s)$ $(r+s=n-1)$ are constructed successively.

2.2.3. Numerical examples compared with other lower bounds

Example 1 ($m=3, n=6$) [Example 1 in sec. 1.1.3.].

Processing Time (hrs.)

i	1	2	3	4	5	6
$m_{1,i}$	6	12	4	3	6	2
$m_{2,i}$	7	2	6	11	8	14
$m_{3,i}$	3	3	8	7	10	12

Scheduling tree of example 1 is shown in Fig. 12.

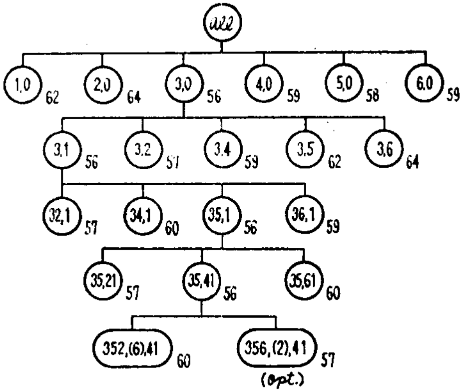

Fig. 12. Scheduling Tree of Example 1.

Number of nodes is a possible least number 20 which is the same as the result in example 1 for revised lower bound LB(J_r) and smaller than result by LB(J_r) (1.5) in sec. 1.1.3 and shall be smaller than the result by LB(R_r) in sec. 2.1.1.

Example 2 ($m=3, n=6$) [Example 3 in sec. 1.1.3]

Processing Time (hrs.)

i	1	2	3	4	5	6
$m_{1,i}$	5	3	12	2	9	11
$m_{2,i}$	9	8	10	6	3	1
$m_{3,i}$	6	2	4	12	7	3

Scheduling tree of example 2 is shown in Fig. 13.

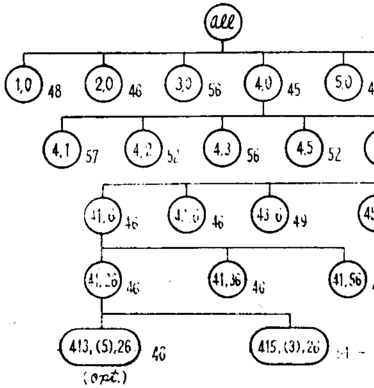

Fig. 13. Scheduling Tree of Example 2.

Number of nodes is a possible least number 20 which is smaller than the results by revised lower bound $LB(J_r)$ (1.2) and lower bound $LB(J_r)$ (1.5) respectively as shown in sec. 1.1.3.

Moreover, if all optimal sequences have to be found, then number of nodes is 28 for $LB(J_r, R_s)$ and at least 42 even for revised lower bound $LB(J_r)$ (1.2).

Remarks: Another B. B. algorithms with $LB(J_r, R_s)$ may be considered as below: [III]. First it's constructed $n(n-1)$ nodes (J_1, R_1), then if lowest bound is $LB(i_1, i_2)$ and $LB(i, i_2)$ is smaller than $LB(i, j)$ for almost all $i(i=1\sim n)$ and $j \neq i_2$, then next it's constructed nodes (i_1, R_2) where last job of R_2 is i_2 and vise versa. [IV]. Policy of branching is to branch from a node of order one with lowest bound.

§ 3. Upper Bound in Case Where no Passing is Allowed

In this section upper bound of makespans of sequences with definite presubsequence is constructed by using Johnson's criterion for two machines and applied to B. B. algorithm for max-makespan problem in case where no passing is allowed.

3.1. Upper bound of makespans of sequences with definite presubsequence

As before, let J_r be definite presubsequence of r jobs and $\bar{J}_r$ be the set of unordered remained $(n-r)$ jobs and $T_k(J_r)$ be completion time of J_r on M_k $(k=1\sim m)$. Also, let $i_1^k\, i_2^k \cdots i_{n-r}^k$ be a sequence of all jobs in $\bar{J}_r$ determined by Johnson's criterion (1.1) for each two machines M_k, M_{k+1} $(k=1\sim m-1)$, then a sequence $\omega_{k+1}^k \equiv i_{n-r}^k \cdots i_2^k\, i_1^k$ is a sequence which is optimal for max-makespan problem for each two machines M_k, M_{k+1} $(k=1\sim m-1)$.

So, in order to construct the upper bound of makespans of sequences of n jobs with definite presubsequence J_r, completion time T_{k+1}^{n-r} of each sequence $i_{n-r}^k \cdots i_2^k\, i_1^k$ on M_{k+1} $(k=1\sim m-1)$ must be defined as shown in the following: [cf. Fig. 14]

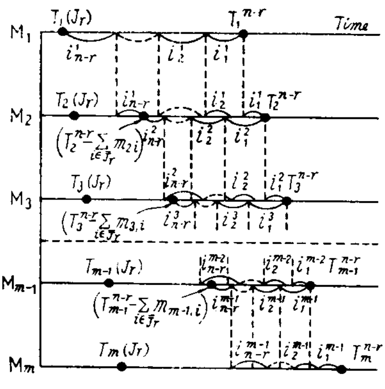

Fig. 14. $UB(J_r)=T_m^{n-r}$.

T_2^{n-r}=completion time of a sequence ω_2^1 on M_2 in case when ω_2^1 is processed on M_1, M_2 after the time $T_1(J_r)$ on M_1, $T_2(J_r)$ on M_2 respectively.

T_3^{n-r}=completion time of a sequence ω_3^2 on M_3 in case when ω_3^2 is processed on M_2, M_3 after the time $T_2^{n-r}-\sum_{i\in\bar{J}_r} m_{2,i}$ on M_2, $T_3(J_r)$ on M_3 respectively.

generally,

T_q^{n-r}=completion time of a sequence ω_q^{q-1} on M_q in case when ω_q^{q-1} is processed on M_{q-1}, M_q after the time $T_{q-1}^{n-r}-\sum_{i\in \bar{J}_r} m_{q-1,i}$ on M_{q-1}, $T_q(J_r)$ on M_q respectively ($q=2\sim m$), where $T_1^{n-r}\equiv T_1(J_r)+\sum_{i\in \bar{J}_r} m_{1,i}$.

Then, for the processing of any sequence of $(n-r)$ jobs in $\bar{J}_r$, its completion time on M_2 isn't after the time T_2^{n-r} and even if its starting time on M_2 be the time $T_2^{n-r}-\sum m_{2,i}$, its completion time on M_3 isn't after the time T_3^{n-r} and by the same reasons its completion time on M_m isn't after the time T_m^{n-r}. So that the time T_m^{n-r} is an upper bound $\mathrm{UB}(J_r)$ of makespans of sequences of n jobs with J_r as their definite presubsequence: that is,

$$\mathrm{UB}(J_r)=T_m^{n-r} \quad (r=1\sim n-1). \tag{3.1}$$

Obviously, $\mathrm{UB}(J_r)$ is a decreasing function of r where $J_{r_1}\subset J_{r_2}$ $(r_1<r_2)$ and $\mathrm{UB}(J_{n-1})$ is equal to real total elapsed time $\mathrm{TE}(J_{n-1})$ of a sequence uniquely determined by J_{n-1}.

Remark. $\mathrm{UB}(R_r)$ and $\mathrm{UB}(J_r, R_s)$ can be constructed by similar devices as in former and present sections.

3.2. B. B. Algorithm for max-makespan problem

Upper bound $\mathrm{UB}(J_r)$ defined in former section can be applied to the following cases where no passing is allowed: that is,

1. By using $\mathrm{UB}(J_r)$ together with $\mathrm{LB}(J_r)$, interval of variability of makespans of sequences with definite presubsequence J_r can be estimated.
2. By using B. B. algorithm with $\mathrm{UB}(J_r)$, optimal sequence can be obtained for max-makespan problem and then largest makespan can be recognized.

Next, B. B. algorithm for max-makespan problem is presented.

B. B. algorithm for max-makespan problem

The procedure is almost the same as that of B. B. algorithm for min- makespan problem in sec. 1.1 except that maximum $\mathrm{UB}(J_r)$ must be branched at each stage and return to a former node (J_r) $(1 \leqq r < n-1)$ must be done when it holds $\mathrm{UB}(J_r) > \mathrm{UB}(J_{n-1})$.

3.3. Numerical example for max-makespan problem

Example $(m=3, n=6)$ [Example 1 in sec. 1.1.3]

Processing Time (hrs.)

i	1	2	3	4	5	6
$m_{1,i}$	6	12	4	3	6	2
$m_{2,i}$	7	2	6	11	8	14
$m_{3,i}$	3	3	8	7	10	12

In this example, by reversing the order determined by Johnson's criterion for two machines, they hold $\omega_2{}^1=251346$ and $\omega_3{}^2=146532$. An example of the calculations of $\mathrm{UB}(J_r)$ will be shown for UB(1) as below:

Job	1	2	5	3	4	6
M_1	6	18	24	28	31	33
M_2	13	20	32	38	49	63

Job	1	4	6	5	3	2
M_2	(22)	33	47	55	61	63
M_3	16	40	59	69	77	80

$T_2^{n-r} - \sum_{i \in J_r} m_{2,i} = 63-41=22.$ $\mathrm{UB}(1)=80.$

Scheduling tree of example is shown in Fig. 15.

Number of nodes in scheduling tree is 39 and all optimal sequences are 214635, 214653 with 80 hrs..

Remarks: inverse sequence of the optimal sequence for max-makespan problem isn't always optimal sequence for min-makespan problem and vise versa. For example, an inverse 536412 of an optimal sequence 214635 for this example has makespan 59 hrs. which is larger than min-makespan 57 hrs. as shown in example 1 in sec. 1.1.3, and an inverse 142653 of an optimal sequence 356241 for min-makespan problem (example 1 in sec. 1.1.3) isn't an optimal sequence for max-makespan problem (example in this section) because its makespan is 70 hrs.

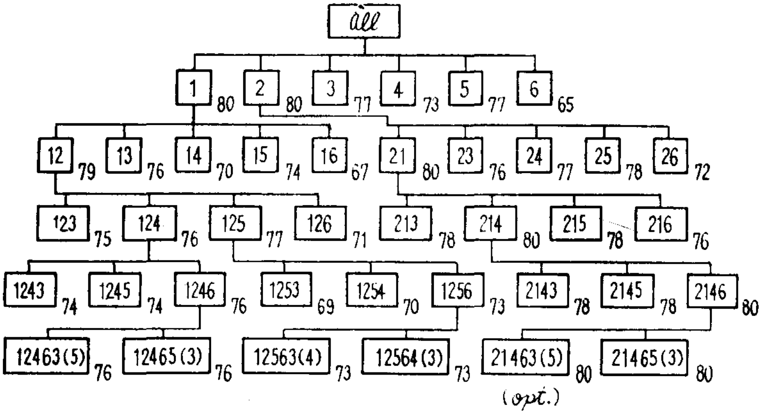

Fig. 15. Scheduling Tree of the Example.

§ 4. Additional Remarks

Each of the B. B. algorithm with lower hound LB(J_r) in sec. 1.1, 1.2, LB(R_r) in sec. 2.1, LB(J_r, R_s) in sec. 2.2, and upper bound UB(J_r) in sec. 3.1 respectively can be programmed for computer, to these this paper doesn't refer. But they will be presented next remarks about the efficiency and sensitivity concerning the B. B. algorithm with each of the above bounds.

4.1. Efficiency of each lower bound defined in the former sections

Revised lower bound LB(J_r) in secs. 1.1 and 1.2 is clearly more efficient than lower bound already given in Refs. [1]~[4], especially for small machine's number. According to an example, each one among revised lower bouud LB(J_r) and LB(R_r) and LB(J_r, R_s) in case where no passing is allowed is more efficient than the others as shown by the example in each section.

But average number of nodes for many examples and nnmber of nodes in case when all optimal sequences must be obtained may be smaller for LB(J_r, R_s) than for the others. On the other hand, complexity of the calculations of the value of lower bound of each node may slightly increase for LB(J_r, R_s). So that, if balance between the number of nodes in scheduling tree in various cases and the quantity of the calculations of the value of lower bound at each node is concerned, revised lower bound LB(J_r) or LB(J_r, R_s) may be more efficient than the others.

Also, if B. B. algorithms with lower bound LB(J_r) and upper bound UB(J_r) are applied together, then range of makespans of sequences with definite presubsequence J_r can be estimated and this may be useful for determining approximate solutions or altimately optimal solutions for min-(max-)makespan problem.

4.2. Sensitivity of the "algorithm" for min-or max-makespan problem

There will happen the cases where some jobs are omitted from or new jobs participate in a present lot of jobs. In these situations, it must be again applied B. B. algorithm to a new lot of jobs from the beginning of its procedure in order to repeat whole steps. Also, even when it must be found an optimal sequence of the subset of n jobs, the situation is the same. That is, in so far as the sensitivity of the "algorithm" is concerned, B. B. algorithm isn't so effective in spite of its speed of computations for each problem.

REFERENCES

[1] Lomnicki Z. A., "A Branch-and-Bound Algorithm for the Exact Solution of the Three machine Scheduling Problem," Oper. Res. Quart., 16, 1 (1965), 89—100.

[2] Ignall E. and L. Schrage, "Application of the Branch and Bound Technique to some Flow-Shop Scheduling Problems," Oper. Res. 13, 3 (1965) 400—412

[3] Brown A.P.G. and Z.A. Lomnicki, " Some Applications of the " Branch-and-Bound " Algorithm to the machine Scheduling Problem," Oper. Res. Quart., 17, 2 (1966), 173—186.
[4] Nabeshima I., " Branch and Bound Algorithms for Optimal Sequening of Jobs through machines Where Passing is Allowed," Rep. Univ. Electro—Comm. **21** (Sci. & Tech. Sect.), (1966), 63—73.
[5] Johnson S. M., " Optimal Two and Three Stage Production Schedule with Setup Time Included," Nav. Res. Log. Quart., 1, 1 (1954), 61—68.
[6] Nabeshima I., " The Order of n Items Processed on m Machines [III]," to appear.