A Method of Flow-Shop Re-Scheduling Dealing with Variation of Productive Capacity

Kenzo KURIHARA
Yusuke OWADA
Yann-liang LI
Kazuaki MASUDA
Department of Industrial Engineering & Management, Kanagawa University
3-27-1, Rokkakubashi, Kanagawa-ku, Yokohama, Kanagawa 221-8686, Japan

and

Nobuyuki NISHIUCHI
Faculty of System Design, Tokyo Metropolitan University
6-6, Asahigaoka, Hino, Tokyo 191-0065, Japan

ABSTRACT

We can make optimum scheduling results using various methods that are proposed by many researchers. However, it is very difficult to complete the processes on time without delaying the schedule. There are two major causes that disturb the planned optimum schedules; they are (1)the variation of productive capacity, and (2)the variation of products' quantities themselves. In this paper, we deal with the former variation, or productive capacities, at flow-shops. When production machines in a shop go out of order at flow-shops, we cannot continue to operate the productions and we have to stop the production line. To the contrary, we can continue to operate the shops even if some workers absent themselves. Of course, in this case, the production capacities become lower, because workers need to move from a machine to another to overcome the shortage of workers and some shops cannot be operated because of the worker shortage.
We developed a new re-scheduling method based on Branch-and-Bound method. We proposed an equation for calculating the lower bound for our Branch-and-Bound method in a practical time. Some evaluation experiments were done using practical data of real flow-shops. We compared our results with those of another simple scheduling method, and we confirmed the total production time of our result is shorter than that of another method by 4%.

Keywords: Flow-Shop Scheduling, Branch-And-Bound Method, Variation of Productive Capacity.

1. INTRODUCTION

There are two major causes that disturb the planned optimum schedules; they are (1)the variation of productive capacity, and (2)the variation of products' quantities themselves. In this paper, we will deal with the former variation, or productive capacities, at flow-shop works. Various scheduling methods have been proposed by making stochastic models on uncertain factors, such as processing time and appointed date of delivery[1][2]. We can also deal with the variance of productive capacity, such as machine troubles and worker absence, by similar stochastic models. This kind of stochastic model may be able to decrease the delays of scheduling against some variation of productive capacity. For example, a scheduling method for a machining center was reported to minimize the total cost considering the occurrence of machine troubles[3]. As shown in this paper, stochastic models are useful for rather long time range scheduling. However, these models are not effective for short time range scheduling, because we cannot absorb the delays within the short time. For the short time range planning, some research results are reported based on heuristic approach [4].

In this paper, we will propose a flow-shop scheduling method dealing with variation of productive capacity. In general, previous flow-shop scheduling methods were developed under the precondition that each shop can be operated independently. However, the shortage of the productive capacity break the precondition, and we cannot accept the independency of shop operations.

In order to solve the above scheduling problem, we will introduce a new concept, or virtual work, that consumes machine time like normal works but has no load on machines. By introducing this idea, we developed a new re-scheduling method based on Branch-and-Bound method. Some evaluation experiments are done using practical data of real flow-shop works. We compared our results with those of another simple scheduling method, and we confirmed the total production time of our result is shorter than that of another method by 4%.

2. FLOW-SHOP RE-SCHEDULING

There are two major factors that decide the productive capacity; they are workers and machines. In our research, we will focus on the workers and we will propose a re-scheduling method when a worker is absent unexpectedly.

Let's consider that one worker is necessary for each shop during the operation. Therefore, when a worker is absent, one of the shops has to be idle. Practically, workers should hop from a shop to another in order to continue the production at the flow-shop. In other words, one of shops has to take a rest because of worker shortage at the same time. In this condition, we want to re-schedule to minimize the total duration of productions at the flow-shops.

We will propose a new re-scheduling method under the above conditions. The problem that we want to solve is defined as follows.

Preconditions

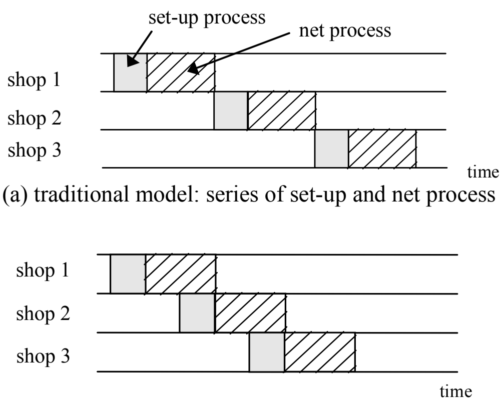
1) Single machine: One machine is set in each shop.
2) No passing: Each process is not allowed to pass other preceding process.
3) No interruption: Operation cannot be interrupted after it started at a shop.
4) Starting time of each process: Set-up processes at the second shop through the last one should be started later than the starting time at the first shop.
5) Parallel processing of set-up processes: Each process consists of two sub-processes, one is a set-up process and the other is an actual main net process. As shown in Fig.1, the set-up process can be operated in parallel with the actual net process at the previous shop.
6) Worker absence: A worker will be absent during some designated duration. During the duration, one shop must stop its operation because of worker shortage.

Objective function

The objective is to minimize the total processing time. We want to decide the processing order of processes in order to finish all the processes as early as possible.

Control variables

We can control only the processing order of processes. The products are processed in the same order at all the shops.



(a) traditional model: series of set-up and net process

(b) proposed model: parallel of set-up and net process

Fig.1 Flow-shop scheduling problem

3. RE-SCHEDULING ALGORITHM

Notations

n : Number of products

m : Number of shops

$\tau_{j,k}$: Set-up process time of product j at shop k (j=1,2,…,n)(k=1,2,3,…,m)

$t_{j,k}$: Net process time of product j at shop k (j=1,2,3,…,n)(k=1,2,3,…,m)

J_r : Sequence of products whose orders are already determined (r = 1,2,…,n)

$\overline{J}_r$: Set of products whose orders has not been determined (r = 1,2,…,n)

$X_{j_r,k}$: Idle time that occurs before the r–th product processing at shop k

$T_k(J_r)$: Finishing time of defined processing order J_r at shop k

A_s : Time when a worker absence happens

A_f : Time when a worker absence clears

d_k : Process time of virtual work at shop k (k=1,2,…,m)

Fundamental Idea

When a worker is absent, one of m shops cannot be operated. In general, this un-operated period flows from the upper shop to the lower one. Considering this characteristic, we introduced a new concept, or virtual product that requires shop occupation but has no load on the shop. Based on this idea, un-operated periods are assigned as shown in Fig.2. We may assign the un-operated period to some idle periods between processes. In this case, we don't need to make un-operated period in parallel. This fact shows that the virtual product may disturb to make optimal schedule. However, mathematically optimum schedules may require the workers to move from one shop to another frequently without considering the workers efficiency. Therefore, mathematically optimum schedules are not always optimal in practical. We can expect efficient scheduling results for workers if we apply our virtual product concept. That's because we can give a simple rule for workers; each worker should simply move to the previous shop, when finishing a process and the shop is in un-operated period.

We will apply the Branch-And-Bound method (BAB) for re-scheduling. Though BAB can search the optimum solution, it demands a lot of calculation time. In this paper, we will propose a lower bound calculation method in practical calculation time.

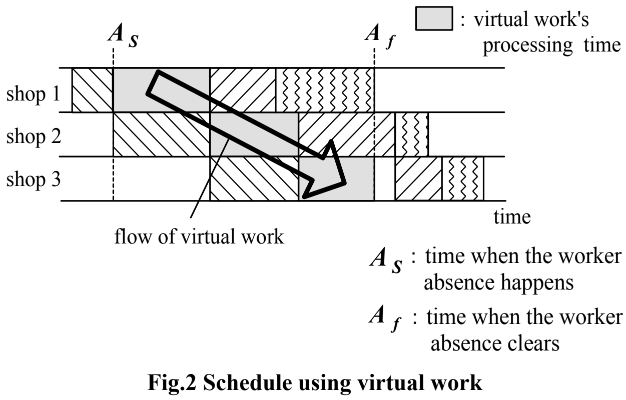


Fig.2 Schedule using virtual work

Selection of Works that Require Re-scheduling

The following equation is satisfied for the planned schedule before the occurrence of worker shortage.

$$T_1(J_r) = T_1(J_r - j_r) + \tau_{j_r,1} + t_{j_r,1} \qquad (1)$$

where,

$$T_k(j_0) = 0 \qquad (k = 1,2,3,\cdots,m-1)$$

First of all, we need to search r_0 that satisfies the equation $T_1\left(J_{r_0}\right) \le A_s < T_1\left(J_{r_0+1}\right)$ based on the above equation (1). Since there is no worker shortage till this process r_0, the fist process through the process r_0 are operated following the planned schedule we already have. The product set $\overline{J}_{r_0}$, that consists of product ($r_0 + 1$) and the later products, is necessary to be re-scheduled, because the process starting time of ($r_0 + 1$) is later than the time A_s, or the beginning time of the worker shortage. When a process finishes its processing at the first shop just after the time A_f, or the worker shortage ending, we can stop assigning the virtual products.

According to this scheduling method, we may not get the optimum solution, because we omitted the re-scheduling of J_{r_0}. The reason why we omitted the re-scheduling is to avoid the confusions, which may cause by the sudden schedule change at the production line.

Lower Bounds Calculation

In the Branch-and-Bound method, it is necessary to develop an efficient lower bound calculation equation $LB(J_r)$. We have developed the following equation (see Eq.(2) and Fig.3) that makes it possible to search the optimum solution in practical calculation time. The equation consists of 3 terms as shown in Eq.(2).

$$LB\left(J_r\right) = \underset{1 \le k \le m}{Max}\left(\ 1^{st}\ term + \ 2^{nd}\ term + 3^{rd}\ term\ \right) \qquad (2)$$

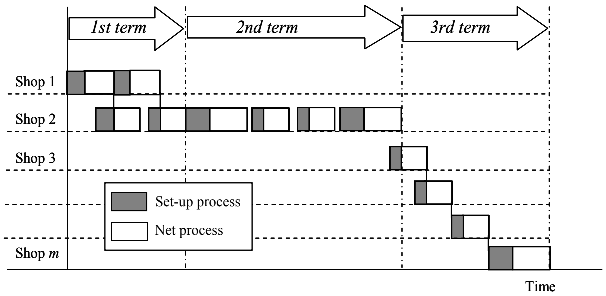


Fig.3 Explanation of lower bound

1st term calculation: This term calculates the process completion time of product processing order set $\boldsymbol{J_r}$ at shop k.

1) Process assignment before the worker shortage starts.

The processing order J_{r_0} is the same as the original scheduling results as follows.

$$T_1\left(J_r\right) = T_1\left(J_r - j_r\right) + \tau_{j_r,1} + t_{j_r,1} \qquad \left(\mathrm{J_r} \subset J_{r0}\right)$$

$$T_k\left(J_r\right) = Max\left(T_{k-1}(J_r),\ T_k(J_r - j_r) + \tau_{j_r,k}\right) + t_{j_r,k} \qquad (3)$$
$$\left(\mathrm{J_r} \subset J_{r0}\right)(k = 2,3,\cdots,m)$$

where,

$$T_k(\ j_0\ ) = 0 \qquad (k = 1,2,3,\cdots,m)$$

2) Process assignment during the worker shortage period (part 1).

The following procedure is for the first virtual product, so the procedure is applied till the first virtual product finishes the last shop processing.

The product set $\overline{J}_{r_0}$, that should be assigned after the worker shortage started, is scheduled producing a virtual product and deciding the processing time of the virtual product as follows. We have to generate one virtual product at arbitral one shop during the worker shortage period. In order to check if a virtual product should be generated or not, the finishing time for the last product of product set J_r at the shop 1 is calculated.

Let's note B_S as the starting time of virtual product. When the finishing time of the product $\boldsymbol{j_r}$, that is the latest ordered product of J_r, satisfies the Eq.(4), a virtual product must be assign before the product j_r.

$$\boldsymbol{B_s \le T_1(J_r) < A_f} \qquad (4)$$

For the first virtual work, $B_S = A_s$ is given.

Let's consider that the first virtual product should be assigned after the product set $J_{r'}$. The virtual product has to flow from upper shop to the lower shop without idle times, because the virtual product processing time represents the time of worker absence. In order not to make idle time, the finishing time of virtual product at every shop can be calculated by Eq.(5). For the last shop, we cannot decide the finishing time at this moment, because we have to decide the time based on the starting time of the following product at the last shop. The details for the decision process will be written later.

We can calculate the finishing times for the processes that follow *r'*-th process using the following equations recursively.

$$T_1\left(J_r\right) = T_1\left(J_r - j_r\right) + \tau_{j_r,1} + t_{j_r,1}$$
$$T_k\left(J_r\right) = Max\left(T_{k-1}(J_r),\ T_k(J_r - j_r) + \tau_{j_r,k}\right) + t_{j_r,k} \qquad (5)$$
$$(k = 2,3,\cdots,m)$$

where

$$T_k(\ j_0\ ) = 0 \qquad (k = 1,2,3,\cdots,m)$$

$$T_k(J_{r'} + j_d) = Max[Min(T_{k+1}(J_{r'}),\quad A_f),\quad T_k(J_{r'})] \qquad (6)$$
$$(k = 1,2,3,\cdots,m-1)$$

The assignment of processes is continued till the starting time of a process reach the finishing time of the virtual product at the last shop. We can calculate the finishing time using the following Eq.(7).

$$T_{mk}\left(J_{r'+1}\right) - \tau_{j_{r'+1},m} - t_{j_{r'+1},m}$$
$$= Max\left(T_{m-1}(J_{r'+1}) - \tau_{j_{r'+1},m}\ ,\ T_m(J_{r'})\right) \qquad (7)$$

Consequently, the above recursive calculation is continued for the product set $J_{r''}$ that satisfies the following Eq.(8).

$$T_1(J_{r''}) \le T_m(J_{r'+1}) < T_1(J_{r''+1}) \qquad (8)$$

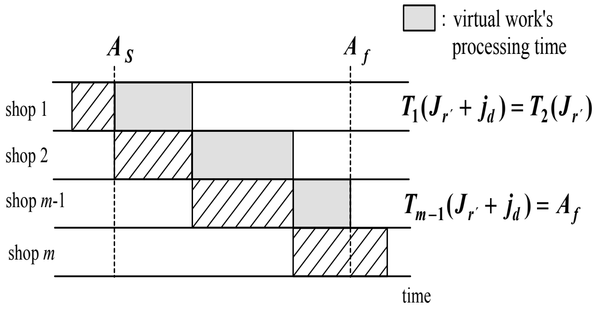


Fig.4 Finishing time of a virtual work at each shop

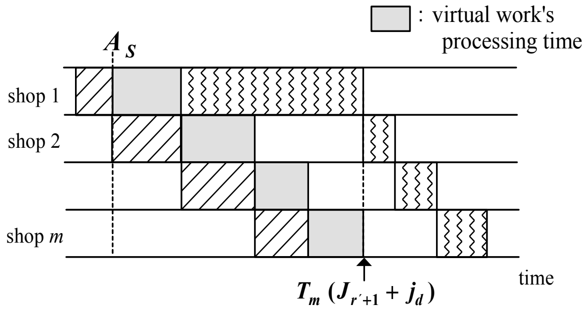


Fig.5 Finishing time of a virtual work at shop *m*

3) Work assignment during the worker shortage period (part 2). According to the above discussion, the starting time of the next virtual product can be decided by the following Eq.(9). The time is the same as finishing time of the previous virtual product at the last shop.

$$B_s = T_m\left(J_{r'+1}\right) - \tau_{j_{r'+1},m} - t_{j_{r'+1},m} \tag{9}$$

The following procedure is the same as the above procedure written in section 2). Whenever a new virtual product is assigned, $\boldsymbol{B_S}$ is re-calculated based on Eq.(9).

4) Work assignment after the worker shortage period.
For the rest products that remain after the above procedures are finished, we can calculate the finishing times for the remaining products using the following equations recursively.

$$T_1(J_r) = T_1(J_r - j_r) + \tau_{j_r,1} + t_{j_r,1}$$

$$T_k(J_r) = Max\left(T_{k-1}(J_r),\ T_k(J_r - j_r) + \tau_{j_r,k}\right) + t_{j_r,k} \qquad (k = 2,3,\cdots,m) \tag{10}$$

where,

$$T_k(\ j_0\ ) = 0 \qquad\qquad (k = 1,2,3,\cdots,m)$$

2nd term calculation: This term calculates the total processing time for the product set $\boldsymbol{\bar{J}_r}$, that contains unassigned products at this point, at the shop k. Even if the processes are processed in parallel, we cannot decrease the net total processing time. Therefore, the total processing time for the product set $\boldsymbol{\bar{J}_r}$ can be calculated as follows.

$$\text{total processing time} = \sum_{j\in \bar{J}_r}\left(\tau_{j,k} + t_{j,k}\right) \tag{11}$$

In our research, we improved the accuracy of the lower bound calculation by including the estimated lower bound of the idle times. The idle time at shop k is considered to be longer than that of two-shop model, or k-1 and k. The idle time will never emerged at shop k-1 for the two-shop model. To the contrary, in the case of multi-shop model, the idle time may be also produced at k-1. This idle time may longer the idle time at shop k, but it will never shorten the idle time. Therefore, the expected minimum idle time for two-shop model gives us the lower bound for the multi-shop model. The expected minimum idle times for two-shop model is calculated as follows.

$$2^{nd}\ term = \sum_{j\in \bar{J}_r}\left(\tau_{j,k} + t_{j,k}\right) + \underset{\text{all order of } \bar{J}_r}{Min}\left(\sum_{j_p\in \bar{J}_r}\left(X_{j_p,k}\right)\right) \tag{12}$$

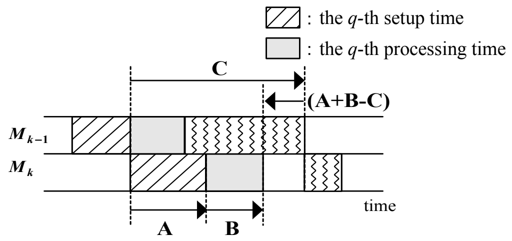
In the above equation, $X_{j_p,k}$ represents the idle time that may occur before the p-th process at shop k. The second term of Eq.(12) is calculated as follows.

Let's consider two-shop model that consists of only k and k-1. We denote that $\boldsymbol{\bar{J}_q}$ (where, $q = 1,\cdots,(n-r)$) is the subset of $\bar{J}_r$, and that $\{J_q\}$ is the set of elements included in $\boldsymbol{J_q}$. Using these notations, the idle time $\sum_{j_p\in\{J_q\}}(X_{j_p,k})$ for $\boldsymbol{J_q}$ at shop k can be calculated as follows. In this paper, we use the notion of $I_2(J_q)$ instead of $\sum_{j_p\in\{J_q\}}(X_{j_p,k})$ just for simplification.
The staying time of the q-th process in the product set $\boldsymbol{J_q}$ can be calculated by Eq.(13), because there is no idle time for the first shop k-1.

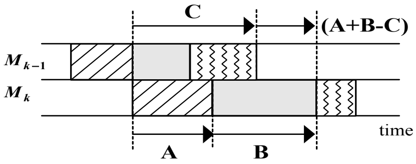
$$\begin{aligned}&\left(\text{Finishing time of the } q-\text{th work at } M_k\right)\\ &\qquad -\left(\text{Finishing time of the } q-\text{th work at } M_{k-1}\right)\\ &=\left[\left\{I_2(J_q) + \sum_{p=1}^{q}\left(\tau_{j_p,k} + t_{j_p,k}\right)\right\} - \sum_{p=1}^{q}\left(\tau_{j_p,k-1} + t_{j_p,k-1}\right)\right]\\ &=\left[I_2(J_q) + \sum_{p=1}^{q}\left(\left(\tau_{j_p,k} + t_{j_p,k}\right) - \left(\tau_{j_p,k-1} + t_{j_p,k-1}\right)\right)\right]\end{aligned} \tag{13}$$

This staying time can be calculated using the following equation derived from the relationship shown in Fig.6.

$$\begin{aligned}&I_2(J_q) + \sum_{p=1}^{q}\left(\left(\tau_{j_p,k} + t_{j_p,k}\right) - \left(\tau_{j_p,k-1} + t_{j_p,k-1}\right)\right)\\ &= t_{j_p,k} + Max\begin{pmatrix} I_2(J_{q-1}) + \sum_{p=1}^{q-1}\left(\left(\tau_{j_p,k} + t_{j_p,k}\right) - \left(\tau_{j_p,k-1} + t_{j_p,k-1}\right)\right)\\ + \tau_{j_p,k} - \left(\tau_{j_p,k-1} + t_{j_p,k-1}\right),\ 0\end{pmatrix}\\ &= Max\left(I_2(J_q) + \sum_{p=1}^{q}\left(\left(\tau_{j_p,k} + t_{j_p,k}\right) - \left(\tau_{j_p,k-1} + t_{j_p,k-1}\right)\right),\ \tau_{j_p,k}\right)\end{aligned} \tag{14}$$



(a) $I_2(J_q)+\sum_{p=1}^{q}\left(\left(\tau_{j_p,k}+t_{j_p,k}\right)-\left(\tau_{j_p,k-1}+t_{j_p,k-1}\right)\right)\le 0$



(b) $I_2(J_q)+\sum_{p=1}^{q}\left(\left(\tau_{j_p,k}+t_{j_p,k}\right)-\left(\tau_{j_p,k-1}+t_{j_p,k-1}\right)\right)\ge 0$

Notes;

$$A = I_2(J_{q-1})+\sum_{p=1}^{q-1}\left(\left(\tau_{j_p,k}+t_{j_p,k}\right)-\left(\tau_{j_p,k-1}+t_{j_p,k-1}\right)\right)$$

$$B = \tau_{j_p,k}$$

$$C = \tau_{j_p,k-1}+t_{j_p,k-1}$$

Fig.6 Mechanism to generate the staying times

Subtract the term $\sum_{p=1}^{q}\left(\left(\tau_{j_p,k}+t_{j_p,k}\right)-\left(\tau_{j_p,k-1}+t_{j_p,k-1}\right)\right)$ from the both term, and we can get the following equation.

$$I_2(J_q) = Max\left(I_2(J_{q-1}),\ \sum_{p=1}^{q}\left(\left(\tau_{j_p,k-1}+t_{j_p,k-1}-\tau_{j_p,k}\right)-\sum_{p=1}^{q-1}\left(t_{j_p,k}\right)\right)\right) \tag{15}$$

Substitute $K_2(q)=\sum_{p=1}^{a}\left(\tau_{j_p,k-1}+t_{j_p,k-1}-\tau_{j_p,k}\right)-\sum_{p=1}^{a-1}t_{j_p,k}$ in Eq.(15), we can get the following equation.

$$I_2(J_q)= Max\left(I_2(J_{q-1}),K_2(q)\right) \tag{16}$$

The following equations are obtained by using the obvious relationship $I_2(J_0)=0$.

$$I_2(J_1)=Max(0,K_2(1))$$

$$\begin{aligned} I_2(J_2) &= Max\left(I_2(J_1),K_2(2)\right) \\ &= Max\left(Max\left(0,K_2(1)\right),K_2(2)\right) \\ &= Max\left(0,K_2(1),K_2(2)\right) \end{aligned} \tag{17}$$

We can think that $K_2(0)=0$ is presumed, and get the following equation.

$$I_2(J_q)= \underset{0\le a\le q}{Max}\left(K_2(a)\right) \tag{18}$$

The above equation doesn't include the idle times that are made in front of virtual products, it is difficult to estimate the lower bound caused by the virtual products. As described before, the virtual product processing times are determined based on the starting/finishing times of other real processes. Therefore, in our research, we estimated the idle times caused by the virtual products as zero. Finally, we get the following Eq.(19) as the second term of Eq.(2).

$$2^{nd}\ term=\sum_{j\in J_r}\left(\tau_{j,k}+t_{j,k}\right)+\underset{\text{allorderof}\,J_r}{Min}\left(\underset{0\le p\le(n-r)}{Max}\left(K_2(n-r)\right)\right) \tag{19}$$

where,

$$K_2(a)=\sum_{p=1}^{a}\left(\tau_{j_p,k-1}+t_{j_p,k-1}-\tau_{j_p,k}\right)-\sum_{p=1}^{a-1}t_{j_p,k}\qquad j_p\in\{J_{n-r}\}$$

3rd term calculation: This term calculates the lower bound of the total processing time, that is necessary to be processed from the (k+1)-th shop to the m-th shop, for un-assigned product set $\bar{J}_r$. Since all the set-up processes may be processed in parallel with actual main net processes, we can ignore the set-up times. As a result, we can estimate the total processing time using the following Eq.(20).

$$The\ \ 3rd\ \ term\ = \underset{j\in \bar{J}_r}{Min}\left(\sum_{q=k+1}^{m}\left(t_{j,q}\right)\right) \tag{20}$$

Scheduling based on BAB

BAB is a useful technique for the tree search. The method doesn't investigate all the combinations in the search space. The search space is divided into some subsets, and the optimum solution is searched efficiently by removing useless subsets, that have no possibility of containing the optimum solution, based on the estimated lower bound values.

There are two procedures for BAB; they are the vertical and the horizontal search. In the vertical search, it is possible to obtain a rather good solution even if the search is quitted at any time because of the long computing time. Therefore, in our research, we applied the vertical search procedure. The concrete procedure is as follows.

Step 1: Set a tentative lower bound f * of total processing time f as ∞.

Step 2: The product set $\boldsymbol{J_0}$ is the start state. Arbitral one product out of all n products can be the candidate as the first processing. So, the initial state is divided into n nodes that represent the n sub-sets. Then, calculate the lower bound of each node $\boldsymbol{LB(J_1)}$ using equations mentioned in the previous section.

Step 3: Search the nodes that have the minimum lower bound value. If more than 2 nodes are found, select the node that contains the smallest product number. Then, the node is divided into (n-1) nodes. Calculate the lower bound of each generated node $\boldsymbol{LB(J_2)}$ as before.

Step 4: Search the nodes that have the minimum lower bound value. And the same procedure as step 2 is done till you can reach the state $\boldsymbol{J_{n-1}}$. If the minimum lower bound value exceeds the tentative lower bound f * during this procedure, it means that the checking sub-set has no possibility of containing the optimum solution in the sub-set. In this case, stop the node expansion of the node, and proceed to step 6.

Table1 Setup and processing time of actual production data

job		machine1		machine2		machine3		machine4		machine5		machine6	
		setup	process	setup	process	setup	process	setup	process	setup	process	setup	process
A	A1	23	10	42	2	53	3	21	3	53	7	30	2
	A2	21	210	41	90	85	80	33	150	54	75	11	55
	A3	28	105	23	40	68	80	21	60	49	75	27	55
	A4	27	105	37	100	82	80	21	180	97	75	21	92
	A5	41	210	31	40	64	80	19	60	73	150	31	55
B	B1	36	53	31	20	52	27	14	34	51	75	16	18
	B2	34	106	42	40	49	54	12	7	68	150	24	37
	B3	23	105	33	120	39	160	27	101	74	225	24	147
	B4	33	210	39	160	110	213	34	180	61	300	21	183
	B5	27	210	42	160	82	160	25	240	84	300	18	220
	B6	25	79	42	30	62	40	30	15	67	113	11	28
C	C1	27	105	23	40	38	54	18	7	72	150	21	37
	C2	37	262	39	200	88	268	13	150	86	206	11	138
	C3	42	157	27	60	66	80	16	10	82	225	24	55
	C4	30	315	34	120	49	160	24	12	82	254	27	110

(minutes)

Step 5: The obtained product set J_{n-1} represents the (n-1) process processing order, so the un-assigned product is only one. Therefore, the product set J_{n-1} specify the concrete order of all n products. This means that the obtained product set J_{n-1} that has the lower bound is a candidate of the optimum solution if $LB(J_r) < f^*$ is satisfied by the node. In this case, set the value of f^* as $LB(J_r)$ of the candidate node. Otherwise, the obtained node is not a candidate of the optimum solution, and do not change the value of the tentative lower bound f^*.

Step 6: Search nodes whose lower bound value is lower than f^*. If there are some, select the nearest node from the latest expanded node and return to step 4. Otherwise, the node corresponding to f^* is optimum.

4. NUMERICAL EXPERIMENTS

Experimental conditions

We evaluated our method by comparing the results with those of another scheduling method. In this evaluation, we had 3 practical cases that were gotten from an actual production line. Each problem consists of 10 products and 6 machines, and we chose 10 products at random from 16 products shown in Table.1. Under the condition that a worker is absent for 16 hours at a designated time, we re-scheduled to finish all the products as early as possible by re-ordering the products.

We compared the results of our proposed method with those of another method. The method is an adjusting method that basically accepts the original schedule before the worker shortage. When a worker finished his process, the worker would check if there was a product waiting for processing at the present shop and at un-operated shop. If there is one at the un-operated shop, the worker moves to the shop. Otherwise, he/she stays at the present shop.

Table2 Scheduling results

case	method	scheduling result	total processing
1	our proposed	B2 - C3 - B3 - A4 - B5 - C4 - B4 - C2 - A3 - A2	55.9
	simple adjust	B2 - C3 - B3 - A4 - B5 - A3 - B4 - C2 - C4 - A2	58.1
2	our proposed	B1 - B2 - B3 - A4 - B5 - C4 - B4 - A3 - A2 - A5	53.2
	simple adjust	B1 - B2 - B3 - A4 - B5 - A3 - C4 - B4 - A2 - A5	55.4
3	our proposed	C1 - B2 - B3 - A4 - B5 - B4 - A3 - C2 - A5 - A1	51.3
	simple adjust	C1 - B2 - B3 - A4 - B5 - A3 - B4 - C2 - A5 - A1	53.5

(hours)

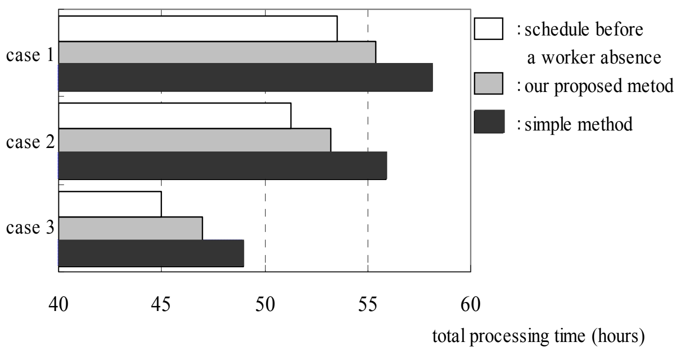


Fig.7 Re-scheduling results

Results

The results are shown in Table2 and Fig.7. Table 2 shows the concrete results of processing order for the 3 cases. You can see the chosen products for 3 cases. In the bar chart of Fig.7, each case shows three bars. The top bar shows the total processing time of the original schedule before the worker shortage, and the second one shows that of the proposed method. The final one shows that of the adjusting method shown above.

In all cases, the re-scheduled results demand longer processing time than the original one, because productive capacity becomes lower than before because of the worker absence. The objective of re-scheduling is to shorten the extra time caused by the worker absence.

In the case 1, the total processing time of the proposed method is 55 hours 54 minutes, and that of the adjusting method is 58 hours 6 minutes. Our method could shorten the time by 2 hour 12 minutes, or 3.8%. Since the reduction effect depends on the characteristics of the actual production data, we cannot guarantee the same reduction for all cases. Nevertheless, we could get the similar reduction effect in other 2 cases as shown in Fig.7. In these three cases, our proposed method could get better results than the adjusting method.

5. CONCLUSIONS

We developed a new re-scheduling method based on Branch-and-Bound method dealing with the decrease of workers. When one shop takes a rest for a while because of worker shortage, a no-load duration will emerge at the shop. In general, since there is no running stock at flow-shops, the no-load duration will flow from an upper stream shop to a lower stream one. Considering this feature of flow-shops, we set-up a new concept of a virtual product that has the similar characteristics as normal products but has no load in processing it. We will input the virtual product into the production line while the productive capacity is lower than usual. We proposed an equation for calculating the lower bound for our Branch-and-Bound method in practical calculation time.

Some evaluation experiments are done using actual data of real flow-shop products. We compared our results with those of another simple scheduling method, and we confirmed the total production time of our result is shorter than that of another method by 4%.

6. REFERENCES

[1] M.Haouari, T.Ladhari, "**Minimizing maximum lateness in a two-machine flow shop**", Journal of the Operational Research Society, Vol.51, No.9, 2000, pp.989-1100.

[2] M.Asano, M.Hino and H.Ohta, "**Scheduling Method Minimizing Inventory Cost with Some Constraints on Process Beginning Time and Appointed Date of Delibery**", Journal of JIMA, Vol.50, No.4, 1999, pp.226--234. (in Japanese)

[3] W.Zeng, H.Nagawasa, N.Nishiyama, "**Single Machine Scheduling for Minimizing the Expected Total Cost Subject to Machine Breakdowns**", *JSME Journal Series C*, vol.58, No.558, 1993, pp.309-315. (in Japanese)

[4] L.Refanidis, I.Vlahavas, "**Multi-objective heuristic state-space planning**", Artificial Intelligence, Vol.145, No.1-2, 2003, pp.1-32.