Production Control on Multi-project System (Optimization of Scheduling)*

By Tatsuo HIRAOKA**

This is a study on the scheduling of a multi-project system under restrictions of limited resources (manpower, equipment etc.). In this scheduling the job durations are controllable by the number of units of resources. A new approach tries to produce a strict optimizing method in which the computation is easier than that of my previous strict optimizing method. Here "a theory of a supplement schedule" is developed, which is not dealt with in any previous studies. Then this theory presents an algorithm which can be put to practical use.

1. Introduction

This is a study on the scheduling in an administrative organization which controls all the plural projects advancing in parallel. The schedule of the multi-project system is optimized under restrictions of the limited resources (such as, manpower, equipment etc.). This study considers a job duration controlled due to the quantity of resources which few previous studies have considered. Therefore, it is an important point to consider various job completion periods for the same job starting period. The previous studies on the resource-constrained multi-project scheduling in which a job duration is controllable by the quantity of the resources, include those of RAMPS^{(1)~(2)}, J.D. Wiest⁽³⁾, A.A. McGee & M.D. Markarian⁽⁴⁾, and that of the author⁽⁵⁾. These previous studies, except that of the author⁽⁵⁾, have not proposed a method of strict optimization. The author has proposed a method of strict optimization by a 0-1 integer programming. In this study a new approach tries to produce a strict optimizing method which is easily utilized in practice. First of all, "a theory of a supplement schedule" is developed here, which has been dealt with in no previous studies. And, this theory presents a strict optimizing algorithm. This theory is used to judge whether there is any possibility without further calculations of obtaining a better entire schedule from the partial schedule obtained in the calculation process. And, it is confirmed that such a strict optimizing algorithm as constructed here drastically reduces the computational time in comparison with the previous strict optimizing algorithm⁽⁵⁾. Then, as a result of an application to practical problems, it is confirmed that this study can be effective for practical use.

2. Model and Objective Function

2.1 Basic Conditions

(1) Define K as the number of projects. Each project has an available starting period (irrespective of the resource limitation). Assume that there are some projects whose available starting periods are the first period.

(2) To each job a wide variety of job speeds are given. And to each speed rank of each job, are given the job duration and the resource requirement per unit duration. Define I as the number of jobs (except "dummy" that is introduced when networks are constructed).

(3) The quantity of resources available in each period is limited. Define $RA_{t,m}$ as the number of units of resource m available in period t , and M as the number of resource types.

2.2 Model I and the Objective Function

Assume that each project has no specified completion deadline. The objective function is to minimize the duration of overall multi-project.

2.3 Model II and the Objective Function

Assume that each project has a specified completion deadline. Define $DU(k)$ as the specified completion deadline of project k . The objective function is to minimize the total delays in the completion deadline for all the projects.

3. Theory of Supplement Schedule

Let Sa be a partial schedule of job set $A (\neq \{\text{all the jobs}\})$, and $S\bar{a}$ be a partial schedule of job set $\bar{A} (\neq \{\text{all the jobs}\})$. Suppose A and $\bar{A}$ satisfy that $A \cup \bar{A} = \{\text{all the jobs}\}$ and $A \cap \bar{A} = \{\text{the empty set}\}$. Then if an entire schedule is obtained by combining Sa with $S\bar{a}$, define that " Sa and $S\bar{a}$ hold a mutual relation of a supplement schedule". However, when resource limitation, etc. prevent producing of an entire schedule despite the fact that A and $\bar{A}$ satisfy the relations mentioned above, it is not said that Sa and $S\bar{a}$ hold a mutual relation of a supplement schedule.

And let Sb be a partial schedule of job set $B (\neq \{\text{all the jobs}\})$ (including the case that B is an empty set and Sb is an empty schedule). When the supplement schedule of Sa is obtained from any entire schedules produced from Sb define that " Sb is always able to supply Sa with a supplement schedule".

[Theorem 1] Model I is supposed. Assume that Sb is always able to supply Sa with a supplement schedule, and the completion period of any job

* Received 6th October 1980.

** Research Student.
Faculty of Engineering, Univ. of Tokyo,
Hongo 7-3-1, Bunkyo-ku, Tokyo.

in S_a is smaller than the minimum value of the objective function of an entire schedule produced from S_a . Then, the value of the objective function of any entire schedule produced from S_b is not smaller than the minimum value of the objective function of an entire schedule produced from S_a . (Proof) Let $Fa1$ be the minimum value of the objective function of an entire schedule produced from S_a . From the assumption of the theorem, S_b is always able to supply S_a with a supplement schedule. Let $\hat{S}$ be an entire schedule produced from S_b , and $Fb1$ be the value of the objective function of $\hat{S}$. Let $\tilde{S}$ be an entire schedule obtained by combining S_a with its supplement schedule obtained from $\hat{S}$, and $Fc1$ be the value of the objective function of $\tilde{S}$. Then, $Fc1 \geq Fa1$. Let j_c be a job which is completed in period $Fc1$ in $\tilde{S}$. From the assumption of the theorem, any job is completed before period $Fa1$ in S_a . Therefore all of the jobs in A are completed before period $Fa1$ in $\tilde{S}$. Hence job j_c which is completed in period $Fc1$ ($Fc1 \geq Fa1$) in $\tilde{S}$ satisfies $j_c \notin A$. Since $\tilde{S}$ is the entire schedule obtained by combining S_a with its supplement schedule obtained from $\hat{S}$, the completion period of j_c in $\tilde{S}$ is equal to the completion period of j_c in $\hat{S}$ when $j_c \notin A$. Therefore, (the completion period of j_c in $\hat{S}$) = $Fc1$. Since $Fb1 \geq$ (the completion period of j_c in $\hat{S}$), $Fb1 \geq Fc1$. Finally $Fb1 \geq Fa1$, and Theorem 1 is proved. (Q.E.D.)

[Theorem 2] Model II is supposed. Assume that S_b is always able to supply S_a with a supplement schedule, and any project k satisfies either of the following two conditions. One is that no jobs in set A belong to project k and have successor jobs. The other is that, for all the entire schedules produced from S_b , the value of $\max\{(\text{the completion period of } k \text{ in the schedule}) - DU(k), 0\}$ is not smaller than the value of

$$\max\{(\text{the job completion period in } S_a) - DU(k), 0\}$$

of any job which belongs to project k and to set A and has no successor jobs. Then, the value of the objective function of any entire schedule produced from S_b is not smaller than the minimum value of the objective function of an entire schedule produced from S_a .

(Proof) Let $Fa2$ be the minimum value of the objective function of an entire schedule produced from S_a . From the assumption of the theorem, S_b is always able to supply S_a with a supplement schedule. Let $\hat{S}$ be an entire schedule produced from S_b , and $Fb2$ be the value of the objective function of $\hat{S}$. Let $\tilde{S}$ be an entire schedule obtained by combining S_a with its supplement schedule obtained from $\hat{S}$, and $Fc2$ be the value of the objective function of $\tilde{S}$. Then, $Fc2 \geq Fa2$. For each project k , define

$$\begin{aligned} \hat{a}_k &= \max\{(\text{the completion period of } k \text{ in } \hat{S}) - DU(k), 0\}, \\ \tilde{a}_k &= \max\{(\text{the completion period of } k \text{ in } \tilde{S}) - DU(k), 0\}, \\ P_k &= \begin{cases} \text{all of the jobs which belong to project } k \text{ and have} \\ \text{no successor jobs} \end{cases} \end{aligned}$$

Let $\tilde{j}$ be a job in P_k . When $\tilde{j} \in A$, since

$$\begin{aligned} \hat{a}_k &\geq \max\{(\text{the completion period of } \tilde{j} \text{ in } S_a) - DU(k), 0\} \\ &\text{from the assumption of the theorem,} \\ \hat{a}_k &\geq \max\{(\text{the completion period of } \tilde{j} \text{ in } \hat{S}) - DU(k), 0\}. \end{aligned}$$

When $\tilde{j} \notin A$, the completion period of $\tilde{j}$ in $\hat{S}$ is equal to the completion period of $\tilde{j}$ in $\tilde{S}$ from the definition that $\tilde{S}$ is the entire schedule obtained by combining S_a with its supplement

schedule obtained from $\hat{S}$. Therefore,

$$\begin{aligned} \hat{a}_k &\geq \max\{(\text{the completion period of } \tilde{j} \text{ in } \hat{S}) - DU(k), 0\} \\ &= \max\{(\text{the completion period of } \tilde{j} \text{ in } \tilde{S}) - DU(k), 0\}, \end{aligned}$$

when $\tilde{j} \notin A$.

$$\therefore \hat{a}_k \geq \max\{(\text{the completion period of } \tilde{j} \text{ in } \tilde{S}) - DU(k), 0\} \quad (\text{for all } \tilde{j} \in P_k)$$

Since

$$\hat{a}_k = \max_{\tilde{j} \in P_k} [\max\{(\text{the completion period of } \tilde{j} \text{ in } \tilde{S}) - DU(k), 0\}],$$

$$\hat{a}_k \geq \tilde{a}_k. \text{ Hence, } \sum_{k=1}^K \hat{a}_k \geq \sum_{k=1}^K \tilde{a}_k.$$

$$\text{Since } Fb2 = \sum_{k=1}^K \hat{a}_k \text{ and } Fc2 = \sum_{k=1}^K \tilde{a}_k, \quad Fb2 \geq Fc2.$$

Finally $Fb2 \geq Fa2$, and Theorem 2 is proved. (Q.E.D.)

4. Main Algorithm

The basic logic of the main algorithm is to improve the schedule gradually with a view to obtaining finally the strict optimal solution. The initial schedule is given — may be given arbitrarily — as an input data prior to the computation. This main algorithm contains the sub algorithm (Section 5) constructed by means of the theory of a supplement schedule which is mentioned above (Section 3). Only the limited information related to schedules and job sets is memorized to eliminate gradual memorizing computational processes (Refer to the main algorithm and the sub algorithm — Section 4.5). Such idea of the optimal memory location yields the result that the memory capacity theoretically increases to an extent not exceeding the first power order for each increase of I , K and M .

4.1 Main Algorithm (Model I)

Step 1 : For each job j , set the job completion deadline to

$$\{(\text{the value of the objective function of the present schedule}) - SLT(j) - 1\}.$$

Note that $SLT(j)$ is the lower bound of the duration required to accomplish all successor jobs of job j (if job j has no successor jobs, it is zero). ($SLT(j)$ — See Reference (5)).

Step 2 : i) Set the job set $\mathcal{Q}$ to the set of all the jobs whose completion periods do not occur prior to period T_1 . Note that T_1 is the earliest period among the completion periods of the jobs completed after their deadline.

ii) Set schedule S to the present schedule (the entire schedule). Set both of the sets $\mathcal{Q}'$ and $\mathcal{Q}''$ to the empty set.

Step 3 : i) If possible, to obtain a schedule better than schedule S (i.e. to reduce the value of the objective function) update starting-completion periods of jobs in $\mathcal{Q}$ by means of the sub algorithm (Section 5).

ii) If a schedule better than S is obtained (the value of the objective function is reduced), go back to Step 1 (however, stop the main algorithm in case the value of the objective function is equal to LBT , the lower bound of the duration of overall multi-project. LBT — See Reference (5)). Otherwise, proceed to Step 4.

Step 4 : If $\mathcal{Q} = \{\text{all the jobs}\}$, stop the main algorithm after setting the present schedule to schedule S (schedule S is a strict optimal solution). Otherwise, go back to Step 3 after updat-

ing (magnifying) $\mathcal{Q}$ as

$$\mathcal{Q} \leftarrow \mathcal{Q} \cup \left\{ \begin{array}{l} \text{all of the jobs whose completion periods are} \\ \text{latest, among the jobs not belonging to } \mathcal{Q}. \end{array} \right\}.$$

4.2 Main Algorithm (Model II)

Step 1 : For each job j , set the job completion deadline to

$$\{DU(\text{the project to which job } j \text{ belongs}) - SLT(j)\}$$

($SLT(j)$ - See Section 4.1).

Step 2 : i) Set the job set $\mathcal{Q}$ to the set of all the jobs whose completion periods do not occur prior to period T_2 . Note that T_2 is the maximum value of T_k for all of the projects $\hat{k}$ completed after their deadline and T_k is the earliest period among the completion periods of the jobs which belong to project $\hat{k}$ and satisfy

$$\begin{aligned} &(\text{job completion period}) - (\text{job completion deadline}) \\ &= (\text{the completion period of project } \hat{k}) - DU(\hat{k}). \end{aligned}$$

ii) Set schedule S to the present schedule (the entire schedule). Set both of the sets $\mathcal{Q}$ and $\mathcal{Q}'$ to the empty set.

Step 3 : i) If possible, to obtain a schedule better than schedule S (i.e. to reduce the value of the objective function) update starting-completion periods of jobs in $\mathcal{Q}$ by means of the sub algorithm (Section 5).

ii) If a schedule better than S is obtained (the value of the objective function is reduced), go back to Step 2 (however, stop the main algorithm in case the value of the objective function is equal to zero). Otherwise, proceed to Step 4.

Step 4 : If $\mathcal{Q} = \{\text{all the jobs}\}$, stop the main algorithm after setting the present schedule to schedule S (schedule S is a strict optimal solution). Otherwise, go back to Step 3 after updating (magnifying) $\mathcal{Q}$ as

$$\mathcal{Q} \leftarrow \mathcal{Q} \cup \left\{ \begin{array}{l} \text{all of the jobs whose completion periods are} \\ \text{latest, among the jobs not belonging to } \mathcal{Q}. \end{array} \right\}.$$

5. Sub Algorithm

The sub algorithm described here is constructed from the theory of a supplement schedule (Section 3).

5.1 Sub Algorithm ($\mathcal{Q} \neq \{\text{all the jobs}\}$)

Step 1 : Make all the jobs in set $\mathcal{Q}$ unscheduled, and set n and θ as $n \leftarrow 0$ and $\theta \leftarrow \{\text{the empty set}\}$.

Step 2 : i) Judge whether each schedule-

eligible job can be performed singly under restrictions where the constraints shown in Table 1^(*) are added. Note that a schedule-eligible job is an unscheduled job which has no unscheduled predecessor jobs. If all the schedule-eligible jobs satisfy the condition mentioned above, proceed to ii) after finding a tentative starting-completion period and the earliest starting period (under restrictions mentioned above) for each schedule-eligible job. Otherwise proceed to Step 7^(*). Note that the tentative starting-completion period is the starting-completion period in the minimum-completion-period performance (or the minimum-starting-period performance among the plural minimum-completion-period performances) of the job.

ii) Judge whether the following condition is satisfied. The condition is that, there is the warning job $j_w^{(*)}$ among the schedule-eligible jobs, j_w can be scheduled with its starting-completion period in schedule S , $t_w \geq TF$ and there are no schedule-eligible jobs which are not j_w and have the earliest starting periods (Step 2-i)) not later than t_w . If the condition is satisfied, proceed to Step 7^(*). Otherwise, proceed to Step 3 when $n > 0$ (proceed to Step 4 when $n = 0$). Note that t_w is the completion period of j_w in S and TF is the minimum value of the tentative completion period (Step 2-i)) for all the schedule-eligible jobs.

Step 3 : Proceed to Step 7^(*) if a job j_r satisfying the following condition is found among the scheduled jobs in set $\mathcal{Q}$ (i.e. $J_1, \dots, J_n$). Otherwise proceed to Step 4. The condition of job j_r is that the completion period of j_r can be reduced (or it is fixed and the starting period of j_r can be reduced) singly. The following constraints are added here.

$$\begin{cases} \hat{R}_{\ell,m} \leq R_{\ell,m} & (\text{if } \ell > TF) \\ \hat{R}_{\ell,m} \leq RA_{\ell,m} - \min_{t=\ell} TF \{RA_{t,m} - R_{t,m}\} & (\text{if } TS \leq \ell \leq TF) \\ \hat{R}_{\ell,m} \leq RA_{\ell,m} & (\text{if } \ell < TS) \end{cases}$$

Note that TS is the minimum value of the earliest starting period (Step 2-i)) for all the schedule-eligible jobs, $R_{\ell,m}$ is the present requirement of resource m in period ℓ , and $\hat{R}_{\ell,m}$ is the requirement of resource m in period ℓ when the starting-completion period of job j_r is updated (the updating is not actually made but it is assumed to have been made).

Step 4 : i) Judge whether the following two

Table 1 Added Constraints

Constraints	Contents
Constraint 1	The job completion period must not exceed the following upper bound. Model I : (job completion deadline) Model II : {(job completion deadline) + (value of objective function of schedule S) - 1 - $\sum \lambda_k$ } (if $K \geq 2$) $k \neq$ (the project to which) the job belongs {(job completion deadline) + (value of objective function of schedule S) - 1} (if $K = 1$)
	λ_k : The maximum value of the delay in the job completion deadline for all the scheduled jobs belonging to project k . If all the jobs belonging to project k are unscheduled, $\lambda_k = 0$. Note that the delay in the job completion deadline is described as $\max\{(\text{job completion period}) - (\text{job completion deadline}), 0\}$.
Constraint 2	The warning job ^(*) must not be performed with its starting-completion period in schedule S .

conditions are satisfied. One is that, there are no schedule-eligible jobs having a tentative completion period (Step 2-i)) later than TF and the earliest starting period (Step 2-i)) not later than TF . The other is that, all of the schedule-eligible jobs having a tentative completion period equal to TF can be scheduled with their tentative completion periods (Step 2-i)) (Resource constraints are satisfied when all of those jobs are scheduled). If both of them are satisfied, proceed to II). Otherwise, proceed to Step 5.

II) Judge whether the following condition is satisfied. The condition is that, when all of the schedule-eligible jobs having tentative completion periods (Step 2-i)) equal to TF are scheduled with their tentative starting-completion periods (Step 2-i)), the constraints shown in Table 1 are satisfied^{(*)5} and all the jobs in $\mathcal{Q}'$ are not scheduled with their starting-completion periods in schedule S . If it is satisfied, update θ as

$$\theta \leftarrow \theta \cup \left\{ \begin{array}{l} \text{all the schedule-eligible jobs having tentative} \\ \text{completion periods (Step 2-i)) equal to } TF \end{array} \right\}$$

and proceed to III). Otherwise proceed to Step 7^{(*)2}(*)6

III) Update n as $n \leftarrow n+1$.

IV) Choose job J_n from among the unscheduled jobs in set θ , and schedule it with its tentative starting-completion period (Step 2-i)). If set θ contains unscheduled jobs, go back to III). Otherwise proceed to Step 6.

Step 5 : First, update n as $n \leftarrow n+1$. Next, from among all the schedule-eligible jobs j , find a job whose $E(j)$ is maximum, and let it be J_n . $E(j)$ is defined as follows :

$$E(j) = \left\{ \begin{array}{l} \left(\begin{array}{l} \text{(the tentative completion period of } j \text{ sought for in} \\ \text{Step 2-i))} - \text{(the completion deadline of } j) \\ \text{(if the earliest starting period of } j \text{ sought} \\ \text{for in Step 2-i)) is later than } TF \end{array} \right) \\ \left(\begin{array}{l} \text{(the tentative completion period of } j \text{ sought for in} \\ \text{Step 2-i))} - \text{(the completion deadline of } j) + W \\ \text{(otherwise)} \end{array} \right) \end{array} \right\}$$

Note that W is a sufficiently large positive number. Then, schedule J_n with its tentative starting-completion period (Step 2-i)) and proceed to Step 6.

Step 6 : I) If all the jobs in $\mathcal{Q}$ are scheduled ($n=|\mathcal{Q}|$), stop the sub algorithm. (An entire schedule better than S is obtained). Otherwise proceed to II) when set $\mathcal{Q}'$ is not empty (go back to Step 2 when set $\mathcal{Q}'$ is empty).

II) If all the jobs in set $\mathcal{Q}'$ are scheduled with their starting-completion periods in the intermediate partial schedule S' ^{(*)7} proceed to IV)^{(*)2} Otherwise proceed to III).

III) Let r be the number of scheduled jobs among jobs $J_1, \dots, J_N$ ^{(*)7} Judge whether the following condition is satisfied. The condition is that, $r < n$, all of the jobs ($\in \mathcal{Q}'$) different from $J'_i (i=r+1, \dots, N)$ are scheduled with their starting-completion periods in the intermediate partial schedule S' , and jobs $J'_i (i=r+1, \dots, N)$ and all of the jobs ($\notin \mathcal{Q}'$) are unscheduled. If the condition is satisfied, set

$$J_{n+i-r} \leftarrow J'_i \quad (i=r+1, \dots, N) \quad \text{and} \\ \theta \leftarrow \theta \cup \{ \text{all the unscheduled jobs in set } \theta' \} \quad (*)7$$

schedule $J'_i (i=r+1, \dots, N)$ with their starting-completion periods in S' , update n as $n \leftarrow n+N-r$, and proceed to IV)^{(*)2} Otherwise go back to Step 2.

IV) Decide to stop the sub algorithm, and proceed to Step 7.

Step 7 : I) If $n=0$, $J_n \in \theta (n>0)$ or to stop the sub algorithm is decided in Step 6-IV), proceed to IV). Otherwise proceed to II).

II) If possible, update the starting-completion period of job J_n . The following two restrictions are added here. One is shown in Table 1. The other is that the completion period of job J_n must be increased (or it is fixed and the starting period of job J_n must be increased). And, under such restrictions, the minimum-completion-period performance (or the minimum-starting-period performance among the plural minimum-completion-period performances) of J_n is found. If the starting-completion period of job J_n is updated, go back to Step 6. Otherwise proceed to III).

III) Update n as $n \leftarrow n-1$, make J_n unscheduled and proceed to IV)^{(*)2}

IV) If $n=0$ or $J_i \in \theta (i=1, \dots, n)$ ^{(*)8} stop the sub algorithm, after setting $\mathcal{Q}$ to all of the jobs ($\in \mathcal{Q}$). (A schedule better than schedule S is not obtained. And no selection of the starting-completion periods of jobs in $\mathcal{Q}$ yields a schedule better than S .) Otherwise, find q that is the maximum number of $\hat{c}(J_i \notin \theta)$. When $n=q$, proceed to V)^{(*)2} When $n < q$, remove $J_i (i=q+1, \dots, n)$ from θ , make them unscheduled^{(*)2}(*)9 update n as $n \leftarrow q$, and proceed to V).

V) Set $\mathcal{Q}' \leftarrow \{ \text{all of the jobs } (\notin \mathcal{Q}) \} \cup \{ \text{jobs } J_1, \dots, J_n \}$. $N \leftarrow n$, $J'_i \leftarrow J_i (i=1, \dots, n)$ and $\theta \leftarrow \theta$. Set the intermediate partial schedule S' to the present partial schedule which is the partial schedule of $\mathcal{Q}'$. Then stop the sub algorithm. (A schedule better than S is not obtained, and it is impossible to obtain an entire schedule better than S from S').

5.2 Sub Algorithm ($\mathcal{Q} = \{ \text{all the jobs} \}$)

Steps 1~5 : See Steps 1~5 in the sub algorithm ($\mathcal{Q} \neq \{ \text{all the jobs} \}$ — Section 5.1).

Step 6 : I) If all the jobs in $\mathcal{Q}$ are scheduled ($n=|\mathcal{Q}|$), stop the sub algorithm. (An entire schedule better than S is obtained). Otherwise proceed to II) when set $\mathcal{Q}'$ is not empty (go back to Step 2 when set $\mathcal{Q}'$ is empty).

II) If all the jobs in set $\mathcal{Q}'$ are scheduled with their starting completion periods in the intermediate partial schedule S' ^{(*)7} proceed to Step 7^{(*)2} Otherwise proceed to III).

III) Let r be the number of scheduled jobs among jobs $J_1, \dots, J_N$ ^{(*)7} Judge whether the following condition is satisfied. The condition is that, $r < n$, all of the jobs ($\in \mathcal{Q}'$) different from $J'_i (i=r+1, \dots, N)$ are scheduled with their starting-completion periods in the intermediate partial schedule S' , and jobs $J'_i (i=r+1, \dots, N)$ and all of the jobs ($\notin \mathcal{Q}'$) are unscheduled. If the condition is satisfied, set $J_{n+i-r} \leftarrow J'_i (i=r+1, \dots, N)$ and $\theta \leftarrow \theta \cup \{ \text{all the unscheduled jobs in set } \theta' \}$ ^{(*)7}, schedule $J'_i (i=r+1, \dots, N)$ with their starting-completion periods in S' , update n as $n \leftarrow n+N-r$, and proceed to Step 7^{(*)2} Otherwise go back to Step 2.

Step 7 : I) If $n>0$ and $J_n \notin \theta$, proceed to II). Otherwise proceed to IV).

II) If possible, update the starting-completion period of job J_n . The following two restrictions are added here. One is shown in Table 1. The other is that the completion period of job J_n must be increased (or it is fixed and the starting period of job J_n must be increased). And,

under such restrictions, the minimum-completion-period performance (or the minimum-starting-period performance among the plural minimum-completion-period performances) of J_n is found. If the starting-completion period of job J_n is updated, go back to Step 6. Otherwise proceed to iii).

iii) Update n as $n \leftarrow n-1$, make J_n unscheduled and proceed to iv) (*2)

iv) If $n=0$ or $J_i \in \Theta (i=1, \dots, n)$ (*8) stop the sub algorithm. (A schedule better than schedule S is not obtained. And no selection of the starting-completion periods of jobs in $\mathcal{Q}$ ($=$ {all the jobs}) yields a schedule better than S .) Otherwise, find q that is the maximum number of $\hat{i}(J_i \notin \Theta)$. When $n=q$, go back to ii) (*2). When $n < q$, remove $J_i (i = q+1, \dots, n)$ from Θ , make them unscheduled (*2) (*9) update n as $n \leftarrow q$, and go back to ii).

6. Numerical Computations and Applications to the Practical Cases

The results of the numerical computations (*10) are shown in Table 2.

First, the comparison is made between the method developed in this study and the previous strict optimizing method of the author (*5). The computational results of problem Nos. 1~8 by both methods are shown in Table 2. The results demonstrate that this study drastically reduces the computational time in comparison with the previous strict optimizing method. The computa-

tions are made with the same computer.

Next, an application to practical job-order production is described. Problem Nos. 17~20 in Table 2 are practical problems of the scheduling in a specific company. In the multi-project system, a job duration is controlled by the quantity of resources. And a completion deadline is specified for each project. Therefore, these cases can be solved by Model II. The computational time is shown in Table 2. Hence it is confirmed that this study can be put to practical use effectively.

Finally, the author considers an increase in the computational time with an increase in the scale of the system (the number of jobs I , the number of projects K and the number of resource types M). The scheduling problem discussed in this study is an NP-problem, and there is no theoretical guarantee that computational time increases in polynomial order with an increase of the scale of the problem. But the results of the computations are as follows. Fig.1 shows that $[(\text{the computational time})/I^2]$ does not increase with an increase of I , Fig.2 shows that $[(\text{the computational time})/K]$ does not increase with an increase of K , and Fig.3 shows that $[(\text{the computational time})/M]$ does not increase with an increase of M . According to the results, it is confirmed that the computational time increases to an extent not exceeding the second power order with an increase of I , and to an extent not exceeding the first power order with an increase of K and M .

Table 2 Computational Results

Problem Number	Scale			Model	Computational time (sec.)	Computational time reported in Reference (5) (sec.)	Remarks
	I	K	M				
1	2	2	2	I	0.110	0.072	
				II	0.122	0.076	
2	4	2	2	I	0.237	0.429	
				II	0.170	1.012	
3	6	2	2	I	0.277	4.467	
				II	0.212	4.524	
4	6	2	3	I	0.291	4.340	
				II	0.230	5.618	
5	6	2	4	I	0.303	4.737	
				II	0.225	4.857	
6	8	2	2	I	0.669	(over 300)	
				II	0.314	237.161	
7	8	3	2	I	0.427	(over 300)	
				II	0.335	22.913	
8	8	4	2	I	0.580	214.152	
				II	0.721	296.984	
9	6	2	1	I	0.293		
				II	0.192		
10	8	1	2	I	0.317		
				II	0.321		
11	8	8	2	I	1.052		
				II	0.803		
12	10	2	2	I	1.247		
				II	0.387		
13	13	2	2	I	0.953		
				II	0.317		
14	15	2	2	I	2.765		
				II	0.342		
15	17	2	2	I	15.646		
				II	1.511		
16	20	2	2	I	0.459		
				II	7.432		
17	199	5	1	II	32.103		Practical case
18	241	5	1	II	83.845		Practical case
19	277	6	1	II	55.456		Practical case
20	293	6	1	II	70.527		Practical case

* I is the number of jobs, K is the number of projects and M is the number of resource types.
Model I•II:
See Section 2.

Fig. 1 Number of Jobs I and $\{(\text{Computational time})/I^2\}$

Fig. 2 Number of Projects K and $\{(\text{Computational time})/K\} (I=8, M=2)$

Fig. 3 Number of Resource Types M and $\{(\text{Computational time})/M\} (I=6, K=2)$

7. Conclusions

This is a study on the resource-constrained multi-project scheduling where a job duration is controllable by the quantity of resources. First, "a theory of a supplement schedule" is developed. And using this theory, a strict optimizing algorithm is developed.

The scheduling problem discussed in this study is an NP-problem and there is no theoretical guarantee that computational time increases in polynomial order with an increase of the scale of the problem. But, according to the results of the numerical computations, it is confirmed that the computational time increases to an extent not exceeding the second power order with an increase of the number of jobs and to an extent not exceeding the first power order with an increase in the number of projects and the number of resource types. The memory capacity theoretically increases to an extent not exceeding the first power order with an increase in the number of jobs, the number of projects and the number of resource types.

Then it is confirmed that this study drastically reduces the computational time in comparison with the previous strict optimizing method and can be put to practical use effectively.

Lastly the author wishes to express his thanks to Mr. Shigeru Watanabe, Rector of Tokyo Metropolitan Engineering Junior College, Mr. Hirofumi Miura, Professor and Mr. Hirochika Inoue, Assistant Professor at Tokyo University for their valuable advices.

(*1) Job j_w is called the warning job when j_w satisfies $j_w \in Q'$ and the following condition. The condition is that all of the jobs in set $Q'(\neq j_w)$

are scheduled with their starting-completion periods in schedule S (or there are no jobs different from j_w in set $\mathcal{Q}$).

(*2) It is impossible to obtain an entire schedule better than S from the present partial schedule (including the empty schedule).

(*3) Assume that $A=\mathcal{Q}$, Sa is a partial schedule of $A(=\mathcal{Q})$ where all the jobs in $A(=\mathcal{Q})$ are scheduled with their starting-completion periods in S , and Sb is the present partial schedule (including the empty schedule). Then A , Sa and Sb satisfy the condition of Theorem 1 or Theorem 2 related to the supplement schedule.

(*4) Assume that

$$A = \{\text{all the jobs scheduled at the present moment}\},$$

Sa is a partial schedule of A obtained by updating the starting-completion period of job j_r under the condition (considered in the sub algorithm—Step 3), and Sb is the present partial schedule. Then A , Sa and Sb satisfy the condition of Theorem 1 or Theorem 2 related to the supplement schedule.

(*5) In model I, the constraints shown in Table 1 are surely satisfied.

(*6) Consider the case where the sum of set $\tilde{\Theta}$ and all the jobs scheduled at the present moment is not $\{\text{all the jobs}\}$. Note that $\tilde{\Theta}$ is a set of all the schedule-eligible jobs whose tentative starting-completion periods are TF . Assume that

$$A = \{\text{all the jobs scheduled at the present moment}\} \cup \tilde{\Theta},$$

Sa is a partial schedule obtained by adding all of the jobs ($\in \tilde{\Theta}$) scheduled with their tentative starting-completion periods to the present partial schedule, and Sb is the present partial schedule (including the empty schedule). Then A , Sa and Sb satisfy the condition of Theorem 1 or Theorem 2 related to the supplement schedule.

(*7) S'_i, J'_i ($i=1, \dots, N$) and Θ' are sought for in the former calculation of the sub algorithm (see Section 4.5).

(*8) Consider the case where all of the jobs $J_1, \dots, J_n$ belong to Θ . Assume that

$$A = \{\text{all of the jobs } (\notin \mathcal{Q})\} \cup \{\text{jobs } J_1, \dots, J_i\},$$

Sa is a partial schedule of A where all the jobs in A are scheduled with their starting-completion periods at the present moment, and Sb is a partial schedule obtained by removing J_i from Sa . Then A , Sa and Sb satisfy the condition of Theorem 1 or Theorem 2 related to the supplement schedule (for $i \leq n$).

(*9) Consider the case where $q < n$. Assume that

$$A = \{\text{all of the jobs } (\notin \mathcal{Q})\} \cup \{\text{jobs } J_1, \dots, J_i\},$$

Sa is a partial schedule of A where all the jobs in A are scheduled with their starting-completion periods at the moment of beginning of Step 7—(iv), and Sb is a partial schedule obtained by removing J_i from Sa . Then A , Sa and Sb satisfy the condition of Theorem 1 or Theorem 2 related to the supplement schedule (for $q+1 \leq i \leq n$).

(*10) The computations were made at Computer Centre of the University of Tokyo (the computer model : HITAC 8800/8700).

References

- (1) Moshman, J., et al., Proc. of the Spring Joint Computer Conf., (1963), p.17.
- (2) Lambourn, S., Computer Jour., Vol.5, No.4 (1963), p.300.
- (3) Wiest, J.D., Management Science, Vol.13, No.6 (1967-2), p.B-359.
- (4) McGee, A.A. and Markarian, M.D., IRE Trans. on Engng. Management, (1962-9), p.104.
- (5) Hiraoka, T., Trans. of the JSME, Vol.46, No.403-C (1980-3), p.335.
- (6) Wagner, H.M., et al., Management Science, Vol.10, No.2 (1964-1), p.316.
- (7) Bennett, F., Unpublished Ph. D. thesis, Cornell Univ., (1966).
- (8) Johnson, T.J.R., Unpublished Ph. D. thesis, M.I.T. (1967).
- (9) Pritsker, A.A., et al., Management Science, Vol.16, No.1 (1969-9), p.73.
- (10) Tone, K., "PERT Text IV (in Japanese)" Toyo-Keizai, Inc. (1967).
- (11) E.W. Davis and Heidorn, G.E., Management Science, Vol.17, No.12 (1971-8), p.B-803.
- (12) Mason, A. and Moodie, C.L., Management Science, Vol.18, No.4 (1971-12), p.B-158.
- (13) Patterson, J.H. and Roth, G.W., AIIE Trans., Vol.8, No.4 (1976), p.449.
- (14) Bernington, G.E. and McGinnis, C.F., NSCO-IE Technical Report, N.C. State Univ., Raleigh, N.C. (1973).
- (15) Stinson, J.F., AIIE Trans, Vol.10, No.3 (1978), p.252.
- (16) Holloway, C.A. et al., Management Science, Vol.25, No.9 (1979-9), p.862.
- (17) Iri, M., Operations Research (in Japanese), Vol.25, No.3 (1980-3), p.187.