Translation from Transactions of IPSJ

Multi-Project Scheduling

Yoshitsugu Ohmae*, Susumu Hasegawa** and Minoru Okuda***

Project scheduling is the foundation of project management.

The well-known Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM) are representative of scheduling techniques. They are used to aid management of a single project without resource constraints.

Resource Allocation and Multi-Project Scheduling (RAMPS), which uses these techniques, has been presented as a technique for making multi-project schedules with restrictions on various resources. However, the definite technique that is the core of RAMPS has not yet been presented.

In this study, the proposed heuristic procedure for solving resource-constrained, multi-project scheduling selects the control factors, weights them properly by learning, and then determines the best scheduling pattern for execution of the schedule.

It has been clearly proven that this procedure achieves both a satisfactory rate of schedule duration and high human resource utilization.

In addition, a procedure is proposed for shortening the penalty delay of the designated project.

1. Introduction

Project scheduling and management is very important in executing large projects including the development of information systems. Generally, the well-known PERT and CPM are used to determine the execution time of a single project without resource constraints and relation to cost. In practical project scheduling, however, multi-project scheduling requires the same resources to be used for all projects and is executed in parallel and concurrently. Therefore, the problem is more complex.

In multi-project scheduling with resource constraints, the problem considered is one in which the resources are distributed among all the activities in such a way as to minimize an object function by the total time required (the scheduled period of the project), the total delay in the delivery date, and the completion time of the total project. Each activity in the projects is dealt with in a given order and is assigned a certain amount of work. However, effective scheduling must be determined when the activities cannot deal with non-concurrent resource constraints.

Several attempts have been made to develop an optimal algorithm for multi-project scheduling with resource constraints (such as the formulated method by Integer Linear Programming [1], the Branch and Bound Method [2], and Backtrack Programming [3]). However, scheduling of projects is a combinatorial problem. Mathematical programming can deal with only a limited scale of project, and it is hard to use in practice. Therefore, heuristic procedures using proper preference rules have been proposed for practical scheduling. Typical solutions include RAMPS [4], developed by C.E.I.R. Inc., and ISMPS [5], developed by Ohmae and Kim. However, in the former, the definite algorithm that determines the scheduling pattern, that is, the core of RAMPS, has not yet been presented.

This study improves on ISMPS, which is a PERT-type scheduling system. It can be applied to multiple projects with resource constraints; it selects control factors, weights them properly, sums up the total score, and then determines the best scheduling pattern for the execution of the schedule. The scheduling result changes largely according to the sorts of data used in the project and the scheduling purpose. Therefore, this study introduces a new concept where the weights of control factors are not fixed, the algorithm separates the computation process and learning process of the optimum weights, and the combination of the optimum weights is learned according to the specific scheduling purpose each time.

In ISMPS, the system engineer, programmer, and coder are considered as the resources; in this study, however, an only kind of working staff is considered as a resource. This means that the study deals with single resource constraints. Furthermore, a procedure is pro-

This is a translation of the paper that appeared originally in Japanese in Transaction of IPSJ, Vol. 31, No. 9 (1990), pp. 1375-1386.

*Department of Information Science, Kanagawa Institute of Technology.

**Hitachi, Ltd.

***NTT Network Systems Development Center.

posed for shortening the penalty delay of the designated project.

2. Fundamentals of Scheduling

The fundamentals of multi-project scheduling and a scheduling scheme are outlined below.

2.1 The Control Object of Project Scheduling

In this study, the control objectives of project scheduling are set up as follows:

1. Shortening the project scheduling period.

All projects should be finished as quickly as possible.

2. Meeting the delivery date of each project.

Each project is given a delivery date, so that the project will not be delayed.

3. Utilization of available resources.

Available resources are used in such a way that they do not become idle. Working staff are treated as a resource.

2.2 Control Factors and Their Effect

In this study, control factors are established as a management method for attaining an object. Important Control factors are shown in Table 1. These factors have been adopted in many previous studies. This study adopts the following four control factors, shown in Table 1:

① Float Days: FD (This adds TF to FF.)
② Latest Starting Time: LS
③ Shortest Activity: SA
④ Finished Work: FW

Other factors are rejected for the following reasons:

WC: This is regarded as the same as FW.
UR, SJ: These are taken into account in this study's algorithm.
RT: This is regarded as the same as SA. SA takes account of the critical path, whereas RT is a local part.
SC: If an activity having many successors is preferred, it is impossible to prevent bottlenecks.
IR: The basis on which each project is ranked is not clear.

2.3 Basic Element

The amount of work for each activity is calculated as follows:

work amount = number of resource units required for each speed × total time required (1)

This expression is widely used in practice. Each activity is selected with one of the three speeds (speed up, normal, slow down). The activity speed is changed by adjusting staff conflicts (Fig. 1). In this study, the starting time of each project is set in advance, and can be postponed but not brought forward.

2.4 Flow of Scheduling Process

A scheme of the scheduling flow is shown in Fig. 2. The learning and computation process is very important.

3. Algorithm of the Computation Process

A scheme of the computation process is shown in Fig. 3.

In this scheme, ② calculation of scores for each activity, ③ determination of a temporary speed for each activity, and ④ determination of an execution pattern of scheduling form the core of the algorithm, and are emphasized later in this paper.

This method is based on calculating each factor needed by PERT at one point of time. Therefore, the initial point of time for calculation is stated.

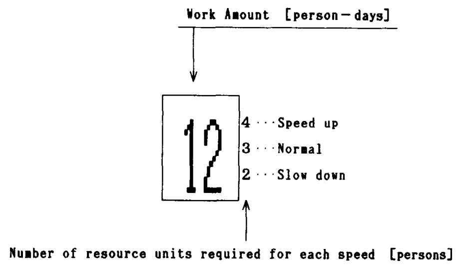

Fig. 1 Explanatory example of activity.

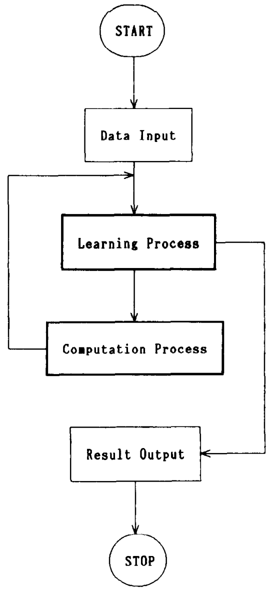

Fig. 2 Scheduling process flow.

Table 1 Control factors and their effects on the schedule.

Control factors	Notation	Methods of control	Effects on schedule				References
			SD	PB	PD	UR	
1. Total Float	TF	Higher priority given to an activity with little float	Y		Y		4),5),6),7),8),10)
2. Free Float	FF	Higher priority given to an activity with little float	Y	Y			4),5),6),7)
3. Latest Starting time	LS	Higher priority given to an activity for which this time is early			Y		5),9)
4. Work Continuity	WC	Higher priority given to an activity that is already executing, in order to restrain interruption		Y		Y	4),6),7)
5. Finished Work	FW	Higher priority given to an activity for which the finished work is large at the computation point of time		Y			8),10)
6. Unemployed Resources	UR	Higher priority given to assignment of unemployed resources				Y	4),6),7)
7. Scheduled Job	SJ	Higher priority given to starting as many activities as possible		Y			4),6),7)
8. Remaining Time	RT	Higher priority given to an activity for which the remaining work time is short	Y	Y			5)
9. Shortest Activity	SA	Higher priority given to an activity that is on the shortest critical path and less short work time	Y	Y	Y		8),10)
10. Successors	SC	Higher priority given to an activity that has many successors		Y			4),5),6),7)
11. Rate of Importance	IR	Higher priority given to an activity for which the rate of importance is high			Y		5)

(Notes)
Y : Adoption in Reference
SD : Shortening of Duration
PB : Prevention of Bottlenecks
PD : Prevention of Delay
UR : Utilization of Resources

3.1 Point of Time for Calculating Scheduling

The points of time for calculating scheduling are shown below:

① Point of time of initial calculation

Point of time at which scheduling starts (0 period)

② Completion of a project's activity

When one activity composing the project is finished, the successor of that activity is the object of the next schedule.

③ Start of a new project

All projects cannot always start at the same time, so the time at which a new project starts is the point of time for calculation.

④ Change in a project

The time at which a change occurs in the project being executed (for instance a change in period demands), or an express project preempts is the point of time for calculation.

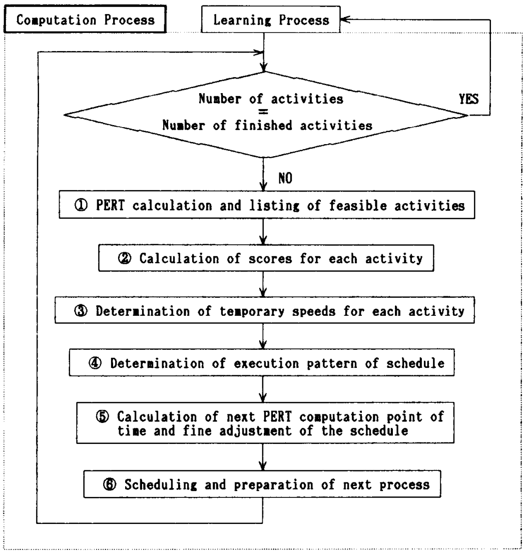

Fig. 3 Algorithm of computation process.

⑤ Regular calculation

The schedule is calculated at set intervals (for example, every week).

In this study, regular calculation is not performed. This is why the activity and resource amount changes in projects ①~④, and ⑤ is a non-changing point of time.

3.2 A Scheduling Execution Example

Figure 4 shows three projects, A, B, and C, which are taken as scheduling objects. Scheduling for these three projects is executed at normal speed without resource constraints. They are finished in 16, 11, and 14 days, respectively. Figure 5 is a load chart. On the seventh day, the number of persons required reaches a maximum of 25. Further, the resource utilization (subsequently referred to as the utilization) in only 58.5%, which is too low.

Table 2 shows an example in which resource constraints decrease from 25 to 15 persons for Fig. 4, and is a PERT calculation of the computation process on the seventh day. The delivery dates of projects A, B, and C (DP: Delivery dates of Projects) are 16, 11, and 14 days, respectively.

−1 in the PC column of Table 2 shows that the activity is finished; therefore, at this point of time, AN 1, 3, 9, 10, 11, 13, 16, 17, and 18 have finished.

Feasible activities are PC=0; therefore, AN 2, 4, 6, 12, 14, 19, 20, and 21 are feasible.

3.3 Calculation of Scores for Each Activity

Calculating the score consists of the following fundamental three steps:

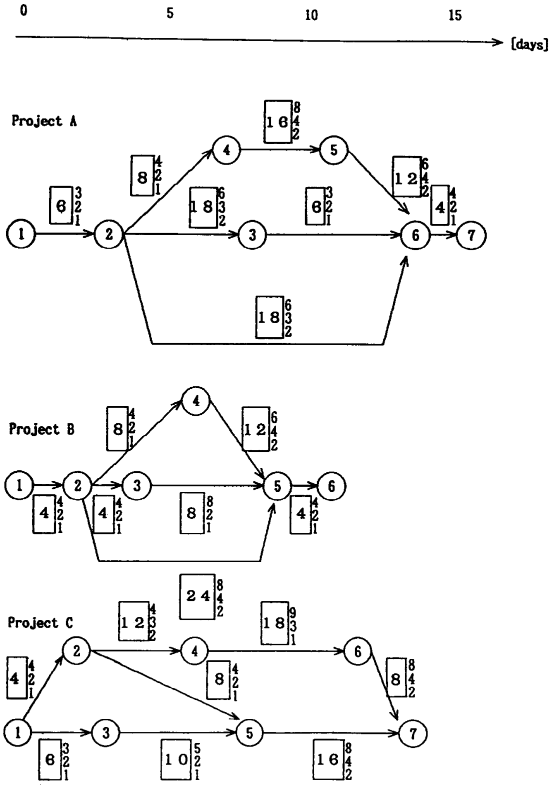

Fig. 4 Example of three projects.

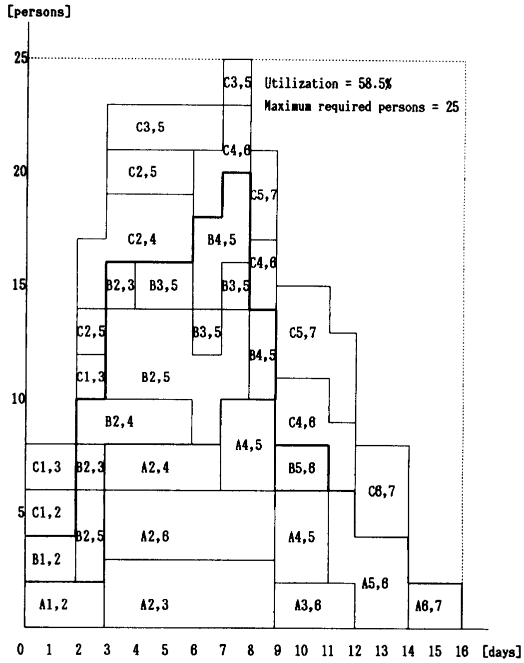

Fig. 5 Load chart for Fig. 4 at normal speeds.

Step 1: Calculation of the base scores for each control factor.

Step 2: Calculation of the scores for each control factor.

Step 3: Calculation of the scores for each activity.

The optimum weights for control factors LS, FD, FW, and SA are determined by the learning process. Therefore, for convenience' sake, the weights are as follows:

$$LSW=30,\ FDW=10,\ FWW=20,\ SAW=40$$

The last *W* of each control factor shows the weight, and the sum total of all the weights is 100.

$$LSW+FDW+FWW+SAW=100 \qquad (2)$$

Step 1: Calculation of the base scores for each control factor.

The base scores for each control factor are calculated as follows:

① Base score of LS: LSB

LSB = LS + delivery date of objective project − estimated completion date of objective project
= LS + DP − EP (3)

It is preferable that the LSB should be low.

② Base score of FD: FDB

$$FDB=(TF+FF)\times PSV \qquad (4)$$

Project Size Value (PSV) is used for comparison under the same conditions regardless of whether the project is big or small.

PSV = (maximum EP)/(EP of objective project) (5)

It is preferable that the FDB should be low.

③ Base score of FW: FWB

FWB = (finished work amount)/(work amount)
= FW/W (6)

It is preferable that the FWB should be high.

④ Base score of SA: SAB

SAB = required duration of the activity + estimated completion date of the target project − the point of time for calculation
= D + EP − PT

It is preferable that the SAB should be low.

Table 3 shows the results of calculating the base scores for each control factor relating to all feasible activities on the seventh day in Table 2.

Step 2: Calculation of the scores for each control factor.

The scores for each control factor are calculated as follows:

Scores for control factor = (weights for each control factor) × (ranking point by base score for each control factor) (7)

Table 2 PERT calculation of the computation process on the seventh day.

AN	PN	Ni	Nj	PC	D	W	FW	LS	TF	FF	EP	DP
1	1	1	2	-1	0	6	6	*	*	*	*	16
2	1	2	3	0	6	18	0	7	0	0	18	16
3	1	2	4	-1	0	8	8	*	*	*	*	16
4	1	2	6	0	6	18	0	10	3	3	18	16
5	1	3	6	1	3	6	0	13	0	0	18	16
6	1	4	5	0	4	16	0	9	2	0	18	16
7	1	5	6	1	3	12	0	13	2	2	18	16
8	1	6	7	3	2	4	0	16	0	0	18	16
9	2	1	2	-1	0	4	4	*	*	*	*	11
10	2	2	3	-1	0	4	4	*	*	*	*	11
11	2	2	4	-1	0	8	8	*	*	*	*	11
12	2	2	5	0	1	24	22	7	0	0	10	11
13	2	3	5	-1	0	8	8	*	*	*	*	11
14	2	4	5	0	1	12	10	7	0	0	10	11
15	2	5	6	2	2	4	0	8	0	0	10	11
16	3	1	2	-1	0	4	4	*	*	*	*	14
17	3	1	3	-1	0	6	6	*	*	*	*	14
18	3	2	4	-1	0	12	12	*	*	*	*	14
19	3	2	5	0	2	8	5	9	2	0	15	14
20	3	3	5	0	2	10	7	9	2	0	15	14
21	3	4	6	0	6	18	0	7	0	0	15	14
22	3	5	7	2	4	16	0	11	2	2	15	14
23	3	6	7	1	2	8	0	13	0	0	15	14

(Notes)
*: Activity already finished
AN: Activity number
Ni: Node i (start node)
PC: Number of preceding activities
W: Work amount
LS: Latest starting time
FF: Free float
DP: Delivery date of project
PN: Project number
Nj: Node j (arrival node)
D: Required duration
FW: Amount of finished work
TF: Total float
EP: Estimated completion date of project

Table 3 Base scores for each activity.

AN	LSB	FDB	FWB	SAB
2	5	0.0	0.000	17
4	8	6.0	0.000	17
6	7	2.0	0.000	15
12	8	0.0	0.917	4
14	8	0.0	0.833	4
19	8	2.4	0.625	10
20	8	2.4	0.700	10
21	6	0.0	0.000	14

The reason for introducing the ranking point by base score for each control factor is as follows.

If the ranking point is not used, then the base scores for each control factor are used directly, and the meaning of multiplying weights is lost, because the base scores are not normalized. Before multiplying weights, the base scores for each control factor are normalized. First of all, the method of normalization ranks the base scores for each control factor, and then sets each point. This method is the same as that used for dividing the prize money at a golf tournament, and allows the range of scores before multiplication of the weights to be unified. The range is from 1 to N (N is the number of feasible activities at the point of time for calculation). The sum total can also be unified.

$$\text{Sum total} = \sum_{K=1}^{N} K \ (K \text{ is an integer, } 1 \ldots N) \quad (8)$$

The ranking point is as follows:

$$\text{Ranking point} = \left(\sum_{K=S}^{F} K \right) \Big/ \text{SAME} \quad (9)$$

Table 4 Ranking points of LSB.

AN	LSB	UPER	SAME	F	S	Ranking point
2	5	0	1	8	8	8
4	8	3	5	5	1	3
6	7	2	1	6	6	6
12	8	3	5	5	1	3
14	8	3	5	5	1	3
19	8	3	5	5	1	3
20	8	3	5	5	1	3
21	6	1	1	7	7	7

$F = N - \text{UPER}$
$S = F - \text{SAME} + 1$
N: the number of feasible activities at the point of time for calculation
K: integer in the series $s, s+1, \ldots F$
UPER: the number base scores with higher priority than the calculation object.
SAME: the number of base scores identical to the calculation object (including itself).

Table 4 shows the result when this method is applied to the LSB in Table 3.

Table 5 shows the result when this method is applied to all of Table 3. The last P of each control factor in Table 5 shows the ranking point.

The scores for each control factor multiply this ranking point by the weights, and are shown in Table 6. The last S of each control factor shows the scores.

Step 3: Calculation of the scores for each activity.

The score for each activity is the sum total of the scores for each control factor, and is calculated as follows:

$$\text{Score for each activity} = \text{LSS} + \text{FDS} + \text{FWS} + \text{SAS} \quad (10)$$

Table 7 shows the result when this method is applied to all the activities in Table 6.

3.4 Determining a Temporary Speed for Each Activity

According to the scores for each factor calculated in the previous section, the temporary speeds are determined for each activity of scheduling. The speed value obtained here (speed-up: 3, normal: 2, slow-down: 1) is called a temporary speed, since it will be changed according to the restrictions on resources when the execution pattern of the schedule is determined. To determine the temporary speeds, the classification values that are the criteria for classifying the speeds for each activity are required. In this study, the largest score from among the feasible activities using the point of time for

Table 5 Ranking points for each factor.

AN	LSP	FDP	FWP	SAP
2	8.0	6.5	2.5	1.5
4	3.0	1.0	2.5	1.5
6	6.0	4.0	2.5	3.0
12	3.0	6.5	8.0	7.5
14	3.0	6.5	7.0	7.5
19	3.0	2.5	5.0	5.5
20	3.0	2.5	6.0	5.5
21	7.0	6.5	2.5	4.0

Table 6 Scores for each factor.

AN	LSS	FDS	FWS	SAS
2	240	65	50	60
4	90	10	50	60
6	180	40	50	120
12	90	65	160	300
14	90	65	140	300
19	90	25	100	220
20	90	25	120	220
21	210	65	50	160

Table 7 Scores for each activity.

AN	Score
2	415
4	210
6	390
12	615
14	595
19	435
20	455
21	485

calculation of the schedule is selected, and then one third and two thirds of that score are set as classification values In the following formulas, MS means the maximum score.

① speed 3: $2/3 \times MS \leqq$ score of activity
② speed 2: $1/3 \times MS <$ score of activity $< 2/3 \times MS$
③ speed 3: $1/3 \times MS \geqq$ score of activity

The higher the speed value of the activity classified, the higher the priority given to scheduling.

Table 8 shows the temporary speeds derived according to the above method.

3.5 Determining the Execution Pattern of a Schedule

(1) Determining an algorithm for the execution pattern of a schedule

The algorithm used to determine the execution pattern of a schedule consists of four calculation patterns, *A* to *D*, as shown in Fig. 6. These are needed to utilize the resource as effectively as possible.

In this case if the activity resource (the sum of the amount of resources utilized in all activities) is equal to the restricted resource amount, the temporary speeds become the speed values for each activity, and we go to the next routine (calculation of the next PERT computation point of time and fine adjustment of the schedule). Otherwise, the calculation patterns of the schedule are performed in alphabetical order unitl the activity resource is equal to the restricted resource amount, and then we go to the next routine. Here the fine adjustment is that if there are activities that can be slowed down without influencing the schedule, the resources assigned for these activities are provided to other activities so that they can be used more effectively.

If the last calculation in pattern *D* is performed and the activity resource is still not equal to the restricted resource amount, we set the speed values for each activity that meet the largest utilized resource amount in all calculation patterns without exceeding the restricted resource amount and then leave this routine.

(2) The calculation pattern structure of a schedule

The schedule calculation pattern consists of two algorithms for determining the priority order and resource adjustment. Each of these algorithms has two patterns. When they are combined, there are therefore four calculation patterns, *A* to *D*, as shown in Fig. 6. These are as follows:

① Calculation pattern *A*
priority order: pattern 1, adjustment of resource: pattern 1

② Calculation pattern *B*
priority order: pattern 2, adjustment of resource: pattern 1

③ Calculation pattern *C*
priority order: pattern 1, adjustment of resource: pattern 2

④ Calculation pattern *D*
priority order: pattern 2, adjustment of resource: pattern 2

Table 8 Temporary speeds for each activity.

AN	Score	Temporary
2	415	3
4	210	2
6	390	2
12	615	3
14	595	3
19	435	3
20	455	3
21	485	3

Here, in priority order pattern 1, each activity is given priority in descending order of its temporary speed, and activities that have the same temporary speed are given priority in descending order of the score for each activity. In priority order pattern 2, each activity is given priority in descending order of its temporary speed, as in pattern 1; however, activities that have the same temporary speed are given priority in descending order of the difference of the required unit resource amount between the present speed value (the temporary speed) and the speed value minus 1.

Furthermore, the resource adjustment algorithm is divided into the following two parts by comparing the activity resource sizes and the restricted resource amount.

1) When there are idle resources
(activity resource < the restricted resource amount)

In this case, the speed values for each activity must be increased to better use the remaining resources. First, the speed values for each activity are increased in ascending order of priority. Next, the additional utilized resources are input, and finally, the newly calculated activity resources are compared with the restricted resource amount. The pattern of speed values is saved for each activity that uses the largest resource within the restricted resource obtained as a result.

2) When there is a shortage of resources
(activity resource > the restricted resource amount)

In this case, the activity resource is decreased by slowing down the speed values for each activity. The speed values are decreased for each activity in descending order of priority and the decrease in resources is subtracted. The above procedure must be repeated until the activity resource is within the range of the restricted resource amount, and the pattern of speed values is saved for each activity that use the largest resource obtained as a result.

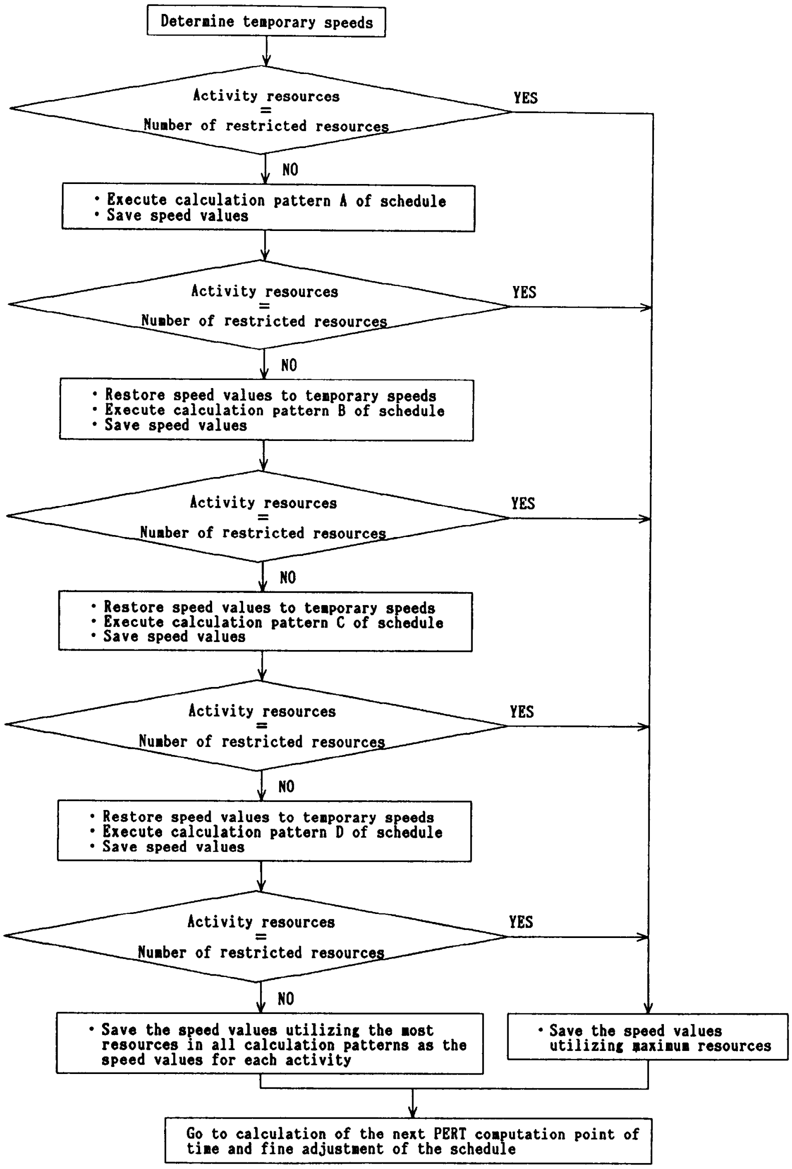

Fig. 6 Algorithm for determining execution pattern of schedule.

Here, the part that increases or decreases the speed value in increasing or decreasing order of priority has two patterns to adjust the resource.

① Resource adjustment pattern 1

Add one to (subtract one from) the speed value in ascending (descending) order of priority, then calculate the activity resource.

② Resource adjustment pattern 2

In ascending (descending) priority order, if it is necessary and possible, increase (decrease) the speed value as much as possible within the limit of the restricted resource amount.

(3) An application example of the routine for determining the execution pattern of a schedule

The above procedure was applied to the example in Table 8. In this instance, the number obtained as a result of calculating the activity resource in the temporary speeds shown in Table 9 was 45 [persons]. Assuming the restricted resource to be 25 [persons], the procedure first enters the resource shortage routine of resource adjustment in the schedule's calculation pattern A.

As a result of applying the resource shortage routine in the schedule's calculation pattern A to the example,

Table 9 Determination of schedule's execution pattern.

State of scheduling activities					Apply the idle resource routine					
AN	Score	3	2	1	Temporary	Resources	Speed	Resources	Speed	Resources
12	615	8	4	2	3	8	2	4	2	4
14	595	6	4	2	3	6	2	4	3	6
21	485	9	3	1	3	9	2	3	2	3
20	455	5	2	1	3	5	2	2	2	2
19	435	4	2	1	3	4	2	2	2	2
2	415	6	3	2	3	6	2	3	2	3
6	390	8	4	2	2	4	1	2	1	2
4	210	6	3	2	2	3	1	2	2	3
Total number of resources						45		22		25

Apply the resource shortage routine

the activity resource becomes 22 [persons]. Since this number is below the restricted resource amount of 25 [persons], it is necessary to re-apply the idle resource routine. The value obtained as a result of execution is equal to the restricted resource amount, so this is saved as the speed value that uses the largest resource and the procedure exits the determination routine of the execution pattern of the schedule.

4. The Learning Process for Optimizing Weight

4.1 A Schedule Execution Example

Figure 7 shows the result of applying this study's procedure to the same example in Fig. 5 under the severe restriction that the restricted resource amount should be 15 [persons]. The weights of the control factors are the same as in the example in Section 3.2.

The finishing dates of projects *A*, *B*, and *C* are the 17th, 9th, and 14th, respectively, which indicates that only project *A* is one day behind schedule. Project *B* finishes two days ahead of the schedule without resource constraints.

Although project *A* is one day behind, we estimate that the schedule has been improved quite significantly, because of the severe restriction.

The resource utilization becomes 96.0%, so resource leveling is improved considerably.

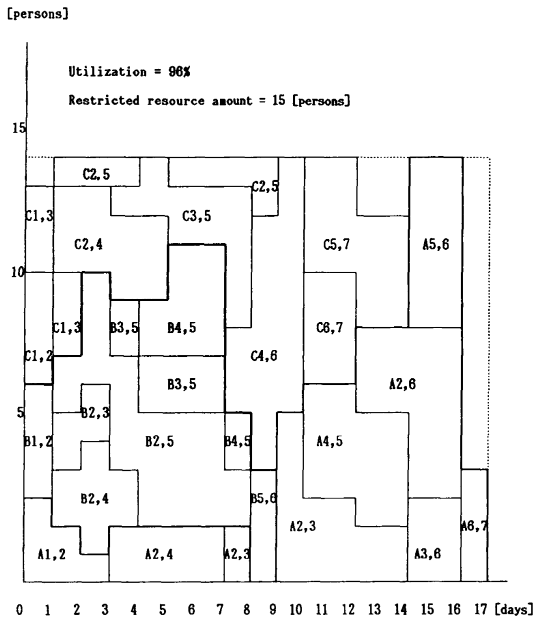

Fig. 7 Load chart of sample schedule.

4.2 Scheduling Evaluation Measures

It was required that evaluation measures should be defined when scheduling was executed. This study defines three evaluation measures and one reference measure. These measures are described below.

① Total Duration: TD

This is the sum of the time (duration) required for all projects:

$$TD=\sum_{i=1}^{h}(\text{Finish}_i-\text{Start}_i)$$

i: the number of projects from 1 to h
Finish_i: the time (date) that project i is finished
Start_i: the starting time (date) of the project's execution

The smaller TD is, the better the schedule becomes.

② Total Penalty Delay: TPD

This is the sum of the delay time of the delivery date required for all projects:

$$TPD=\sum_{i=1}^{h}\text{Penalty Delay}_i$$

Penalty Delay_i:
If $DP_i \geqq \text{Finish}_i$ Then $\text{Penalty Delay}_i=0$
Else $\text{Penalty Delay}_i=\text{Finish}_i-DP_i$
DP_i: the delivery date of project i

The smaller TPD is, the better the schedule becomes.

③ Finishing Time of All Projects: FTAP

This is the time (date) that all projects are finished:

$$FTAP=\underset{i\in h}{\text{Max}}(\text{Finish}_i)$$

The smaller FTAP is, the better the schedule becomes.

④ Resource Utilization: RU

This is the resource utilization in executing all projects.

$$RU=(TWA/(MAR\times FTAP))\times 100$$

TWA: Total Work Amount. i.e. the sum of the actual work amounts for all activities.
MAR: maximum Assigned Resource. i.e. the maximum resource amount assigned during the project execution duration.

The larger this value becomes, the more resource utilization is leveled in the schedule. However, since we will be able to raise it if we intentionally assign a remaining resource to an activity, it is not always better when it becomes larger than other objects, so it is defined as a reference measure.

4.3 Learning Process

The key feature of this study is determining the combination of the optimum weights of control factors by using the following learning process:

(1) The combination of weights in learning

The control factor weights for learning are determined according to the following rules:

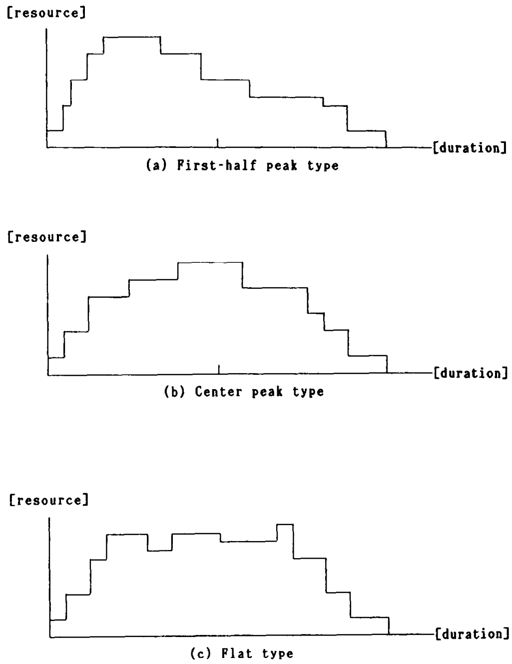

Fig. 8 Load profile.

① The possible weights for each control factor range from 0 tp 100 at intervals of 10.

② The sum of the weights for all control factors is 100.

There are 286 weight combinations. The optimum combination of weights is determined by using all of them. If the combination with the optimum weight is required more precisely, it can be found by searching the adjacency of the weights obtained by using the above learning in practical time.

(2) Learning procedure

In Section 4.2, we defined three evaluation measures that are also objective functions corresponding to control objects, in order to evaluate the scheduling result. In this study, we find the optimum weights for each evaluation measure by using these objective functions and learn the combination of weights by computer simulation.

4.4 Simulation Data

Input data for simulation are classified into the following three types: (a) the first-half peak type, (b) the center peak type, and (c) the flat type. This is done by using a load profile obtained as a result of scheduling at normal speed without resource constraints. The above types are illustrated in Fig. 8.

As shown in Table 10, sixteen projects (A to P) were prepared and the simulation data (EX1 to EX6) were created by combining three projects in each.

4.5 The Simulation Result and its Significance

Table 11 shows the optimum combination of weights for each evaluation measure and the best values of each objective function, where the rows of weights are LSW, FDW, FWW, SAW.

Table 12 shows the schedule result for execution at normal speed without resource constraints, and shows the best and worst values of the simulation result, where TPD becomes O, since we defined the completion date of the schedule executed at normal speed without resource constraints for each project as the projects' delivery date.

By comparing and evaluating the schedule result for execution at normal speed without resource constraints with the best values applied to the procedure of this study, shown in Table 12, we found that the schedule result applied to the procedure of this study was satisfactory, in view of the fact that it is under severe resource constraints. In particular, the first-half peak type (EX1, EX2) shows superior schedule results. It is conjectured that all the projects of this type start at earlier times than those of other types.

Next, by observing the optimum combination of weights for each of the evaluation measures shown in Table 11, although they essentially vary with every change in evaluation measures, data types, and other features, we found that the large weights mainly tend to be FDW and SAW for Total Duration; LSW, FDW, and SAW for Total Penalty Delay; and FDW and FWW for Finishing Time of All Projects. However, the optimum weights vary substantially from case to case. This backs up the necessity of learning the combination of optimum weights of control factors with all the objects and data proposed in this study. Moreover, since there is a large difference between the best and the worst values in Table 12, learning the weights is highly effective.

Table 10 Simulation Data.

Example	Data type	Maximum assigned number of resources [persons]	Restricted number of resources [persons]	Project names contained
EX1	First-half peak type	152 (42.97%) *1	80	ProjectA (0 ~ 183)*2 ProjectB (0 ~ 105) ProjectC (0 ~ 88)
EX2	First-half peak type	85 (55.10%)	50	ProjectD (0 ~ 86) ProjectE (0 ~ 64) ProjectF (0 ~ 32)
EX3	Center peak type	151 (43.14%)	80	ProjectK (0 ~ 183) ProjectL (48 ~ 153) ProjectM (40 ~ 128)
EX4	Center peak type	47 (38.82%)	15	ProjectN (0 ~ 135) ProjectO (40 ~ 148) ProjectP (70 ~ 130)
EX5	Flat type	52 (60.78%)	30	ProjectC (0 ~ 88) ProjectD (55 ~ 141) ProjectG (105 ~ 192)
EX6	Flat type	52 (61.02%)	30	ProjectH (0 ~ 44) ProjectI (10 ~ 47) ProjectJ (30 ~ 74)

(Note)
*1 : Figures are for resource utilization of scheduling at normal speed without resource constraints.
*2 : Figures show the starting time of a project and its delivery date.

5. Procedure for Shortening the Penalty Delay of a Designated Project

Application of the above procedure may require a procedure for shortening the penalty delay for the designated project.

For example, let us suppose that when three projects, *A*, *B*, and *C*, must be executed concurrently, although projects *A* and *C* are permitted to be delayed a few days after their respective delivery dates, project *B* must unconditionally meet its delivery date. A procedure for dealing with such a case, which has a special requirement for only the specific project, is given below.

In this case, two delay adjustment goals are set, in the following order of importance:

① Make the Penalty Delay of the designated project 0.

② Minimize the Total Penalty Delay.

The first implies that the designated project should be made to meet its delivery date. The second aims at minimizing the delays of other projects, because making the designated project meet its delivery date will affect other projects.

To make the designated project meet its delivery date, a higher priority is given to this project. If the scores for each activity that belongs to the designated project are increased, their order of priority will be relatively higher. Thus the scores for each activity belonging to the designated project are made larger than the others by multiplying the scores for each activity by the corrective coefficient of the priority order, which is briefly calculated as follows:

① Multiply the scores of each activity that belongs to the designated project by a coefficient not less than 1.

Table 11 Optimum weights for each evaluation measure.

Example		Total duration	Total penalty delay	Finishing time of all projects
EX1	Best	375 [days]	13 [days]	187 [days]
(First-half peak type)	Combination of optimum weights	(0,30,10,60)	(0,30,10,60)	(0,40,60, 0)
EX2	Best	180 [days]	8 [days]	88 [days]
(First-half peak type)	Combination of optimum weights	(0,30,20,50) (0,30, 0,70)	(30,20, 0,50) (10,30,30,30) (0,40,20,40)	(0,60,40, 0) (0,40,60, 0) (0,30,70, 0)
EX3	Best	389 [days]	25 [days]	179 [days]
(Center peak type)	Combination of optimum weights	(0,30, 0,70)	(0,50, 0,50)	(0,50,50, 0)
EX4	Best	362 [days]	69 [days]	199 [days]
(Center peak type)	Combination of optimum weights	(0,40,10,50)	(0,40,10,50)	(20,70,10, 0) (30,60,10, 0)
EX5	Best	281 [days]	33 [days]	212 [days]
(Flat type)	Combination of optimum weights	(30, 0,30,40)	(20,20,20,40) (10,40,10,40)	(40,20,10,30)
EX6	Best	132 [days]	20 [days]	88 [days]
(Flat type)	Combination of optimum weights	(20,30, 0,50)	(50,10, 0,40)	(50,10, 0,40)

Table 12 Simulation result.

Example		Maximum assigned number of resources [persons]	Total duration [days]	Total penalty delay [days]	Finishing time of all projects [days]
EX1	Normal	152	376	0	183
	Best	80	375	13	187
	Worst	80	466	90	213
EX2	Normal	85	182	0	86
	Best	50	180	8	88
	Worst	50	215	33	104
EX3	Normal	151	376	0	183
	Best	80	389	25	179
	Worst	80	469	60	230
EX4	Normal	47	303	0	130
	Best	15	362	69	199
	Worst	15	477	180	232
EX5	Normal	52	261	0	192
	Best	30	281	33	212
	Worst	30	440	82	232
EX6	Normal	52	125	0	74
	Best	30	132	20	88
	Worst	30	189	39	97

② For other scores, fix the coefficient at 1.

By using this procedure, it is possible to make the designated project meet its delivery date. In this case, a coefficient for the designated project may be selected in such a way that the TPD is the smallest among those for which the designated project's completion date is equal to its delivery date, or one or two days ahead of it.

6. Conclusion

Many factors influence a schedule, such as the data used in a project, the network structure, the amount of work for each activity, and the number of resource units for each speed. Moreover, the schedule may vary greatly depending on the aim of scheduling, such as shortening the required duration or preventing delays in delivery dates.

In this study, control factors were selected in order to give the best scheduling, and it is clear that learning the combination of their weights was effective. An adjustment procedure for shortening the penalty delay of the designated project was also proposed.

We believe that the procedure proposed in this study provides a useful method for multi-project scheduling.

References

1. NABESHIMA, I. Theory of Scheduling, Shikousha (1974), 232-250.
2. MODER, J. J., PHILLIPS, C. R. and DAVIS, E. W. Project Management with CPM, PERT and Precedence Diagramming, Van Nostrand Reinhold Company Inc. (1983), 228-236.
3. NABESIMA, I. A Backtrack Programming Algorithm and Reliable Heuristic Programs for General Scheduling Problems, *Rep. Univ. Elect.-Comm.*, **24**, 1 (1973), 23-36.
4. MOSHMAN, J., JOHNSON, J. and LARSEN, M. Ramps—A Technique for Resource Allocation and Multi-Project Scheduling, *Proceedings-Spring Joint Computer Conference* (1963), 17-27.
5. OHMAE, Y. and KIM, B. K. Multi-Project Scheduling for Information Systems, User-Oriented Information System Symposium, *IPS Japan* (June 1988), 141-150.
6. TONE, K. PERT Course I ~ IV, Touyou Keizai Shinpousha (1967).
7. SEKINE, T. PERT. CPM, *Nikagiren* (1965).
8. KURTULUS, I. S. and DAVIS, E. W. Multi-Project Scheduling: Categorization of Heuristic Rules Performance, *Management Science*, **28**, 2 (Feb. 1982), 161-172.
9. TALBOT, F. B. Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case, *Management Science*, **28**, 10 (Oct. 1985), 1197-1210.
10. KURTULUS, I. S. and NARULA, S. C. Multi-Project Scheduling: Analysis of Project Performance, *IIE Trans*, **17**, 1 (1985), 58-66.