Scheduling Using Shape Processing†

— Application of Coded Boundary Representation (Report III) —

Katsunori INOUE*, Masashi OHKUBO and Yasuhisa YAMAJI*****

Abstract

This paper describes the shape processing approach to the scheduling problem. Although many methods are proposed for the scheduling problem until now, it is very difficult to solve, if the conditions change widely or during the course of processing. The method proposed here is to use the shape processing technique to increase the flexibility in the solution of the scheduling problem. The system utilizes the CBR method which was developed as the geometric model. The CBR method has several features. For example, the model can be represented in the form of lists, and topological features can be easily extracted. The present method is believed to provide a very useful tool for solving the scheduling problem for diversifying products.

KEY WORDS :(Flexible Scheduling)(Shape Processing)(Coded Boundary Representation)

1. Introduction

In scheduling, it is a problem that there is no general technique to analyze the best solution and the ability of the computer is still insufficient for the search into all solutions because of so many existences of the solution which satisfies the constraint conditions in general.

Many researches have been proposed for such problems are divided roughly into the following two kinds. One sticks to the best solution at last, uses the branch-and-bound search method, ATMS [6] and so on [7]. which reduces the search space for the solution, and another technique is to try to request the solution which can be endured to some degree without sticking to the best solution by the approximation algorithm such as PERT [1)-3)].

On the other hand, recently such problems including the scheduling are visualized by shapes, graphics and so on and the research which tries to solve the problem using shape processing is performed a lot. Especially, the scheduling problem is solved using shape processing well [4)-5)].

This research aims at the development of the geometric model which is suitable for scheduling problem and the scheduling system to solve the problems which PERT treats, replacing with the packing problem because these two problems have some resemblance mathematically.

In this research, the system uses the geometric model of the CBR (Coded Boundary Representation) method which developed as a simple CAD model [8].

The features of this shape model are that it can replace various shape processings with the easy list operations because shape in the shape is expressed by some lists and it can extract the topological characteristic of shape easily because one of the lists further expresses the each edge direction of the shape.

2. CBR Method

The CBR method has been developed for simple CAD system. Although the CBR method is basically same as the B-reps which is used as the geometric model in some CAD/CAM systems, these are different in some points, which are CBR method's features as mentioned above. Some application examples for CBR method are reported, which make the best use of its features such as giving the edge direction and expressing the shape with the some lists.

2.1 Basic Model

The shape model used by this research is expressed by the some lists including the direction code list in four directions of the orthogonalization shown in **Fig.1(a).**

Basic shape consists from the edge length list *(l__list)*, the edge name (number) list which identifies the edge *(n__list)*, the vertex lists which show the edge starting point and the terminal point *(s__list* and *e__list)* and the edge direction list *(d__list)*. Then, it is assumed that the substance of the shape is on the edge's left side.

† Received on Oct.31,1992

* Professor
** Research Associate
*** Graduate student of Osaka University (Presently at SHARP Co.)

Transactions of JWRI is published by Welding Research Institute, Osaka University, Ibaraki, Osaka 567, Japan

By defining like this, for example the lists of the shape shown in **Fig.1(b)** can be expressed as follows. In Fig.1(b) the numbers outside of the contour show the edge number and the inside show edge length.

n __list[1,2,3,4].
s __list[a,b,c,d].
e __list[b,c,d,a].
d __list[e,n,w,s].
l __list[4,2,4,2].

Where, the edge direction list *(d__list)* shows topological shape, therefore topological characters in shape are able to be extracted easily from this *d__list*. For instance, whether the ruggedness (convex or concave) exists in the shape is detected by whether the list shown in **Fig.2(a)** or **(b)** is included in the *d__list*. And, it is able to replace shape operations such as the composition, decomposition, the mirrored and the rotated operation with easy list operations by expressing the shape lists like this.

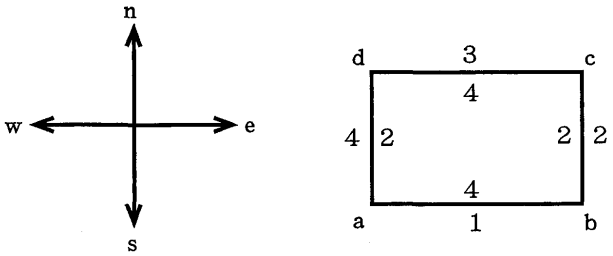
2.2 Example of Shape Processing

As an example of the shape processing, in **Fig.3(a)** the composition processing by connecting the element *'d'* and *'g'* of the vertex of two shapes *A* and *B* is described here. And these operations are well used in the scheduling system as described later.

The shape lists of these two shapes *A* and *B* can be shown as follows.

Shape A:
n __list[1,2,3,4,5,6].
s __list[a,b,c,d,e,f].
e __list[b,c,d,e,f,a].
d __list[e,n,w,n,w,s].
l __list[4,2,2,2,2,4].

Shape B:
n __list[7,8,9,10].
s __list[g,h,i,j].
e __list[h,i,j,g].
d __list[e,n,w,s].
l __list[3,2,3,2].



(a) (b)

Fig.1 CBR method
(a) directional codes
(b) basic shape

Operation 1:

Rotate the all lists of shapes *A* and *B* to become the connecting vertexes (*'d'* and *'g'*) at the head in the *s__* lists.

Shape A:
n __list[4,5,6,1,2,3].
s __list[d,e,f,a,b,c].
e __list[e,f,a,b,c,d].
d __list[n,w,s,e,n,w].
l __list[2,2,4,4,2,2].

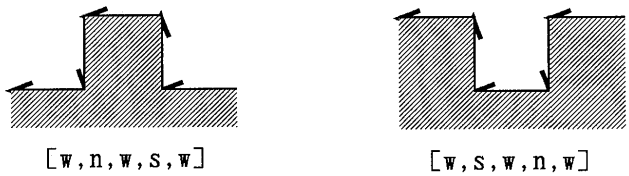
Shape B:
n __list[7,8,9,10].
s __list[g,h,i,j].
e __list[h,i,j,g].
d __list[e,n,w,s].
l __list[3,2,3,2].

Operation 2:

Compare the first element in the *d__*list of shape *A* and the last element in the *d__*list of shape *B* and in case of the combination (element *'e'* and *'w'* or element *'n'* and *'s'*), it means that the first edge of the shape *A* comes into contact with the last edge of the shape *B*.

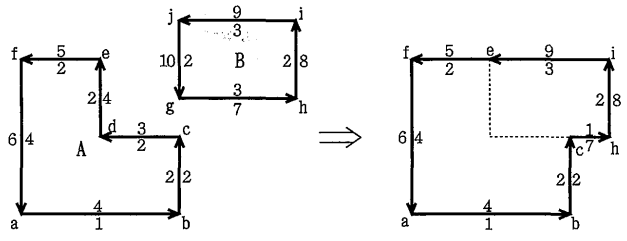
Then compare the each element of the *l__lists*. If the numbers of these elements are same, delete all elements of each list which position is corresponding with the each compared element of the *d__lists*. And in the another case, replace the larger value with difference value of these elements. At this time, the element of the *s__list* in the shape which has larger value is replaced with the element of thes__list in the shape which has smaller value.

On the other hand, all elements of each list which positions correspond to the element are deleted of the shape which has the small value.



(a) (b)

Fig.2 Detection of
(a) convexity
(b) concavity



(a) (b)

Fig.3 Example of shape processing (composition)
(a) before processing
(b) after processing

Shape A:	*Shape B:*
n _list[,5,6,1,2,3].	*n _list[7,8,9,].*
s _list[,e,f,a,b,c].	*s _list[g,h,i,].*
e _list[,f,a,b,c,d].	*e _list[h,i,j,].*
d _list[,w,s,e,n,w].	*d _list[e,n,w,].*
l _list[,2,4,4,2,2].	*l _list[3,2,3,].*

Operation 3:

Do the similar processing to operation 2 for the last element in the *d_list* of shape A and the first element in the *d_list* of shape *B*.

Shape A:	*Shape B:*
n _list[,5,6,1,2,].	*n _list[7,8,9,].*
s _list[,e,f,a,b,].	*s _list[c,h,i,].*
e _list[,f,a,b,c,].	*e _list[h,i,j,].*
d _list[,w,s,e,n,].	*d _list[e,n,w,].*
l _list[,2,4,4,2,].	*l _list[1,2,3,].*

Operation 4:

Append the all lists in shape *B* with shape *A* behind each corresponding list. And after post processing, the lists of composed shape of *A* and *B* is

n _list[1,2,7,8,9,5,6].
s _list[a,b,c,h,i,e,f].
e _list[b,c,h,i,e,f,a].
d _list[e,n,e,n,w,w,s].
l _list[4,2,1,2,3,2,4].

Here, a series of operation is ended **(Fig.3(b)**).

The degeneracy (of the edge) processing, the decomposition of the shape etc. can be defined as each series of easy list operations as well as these composition operations.

3. Application of CBR Method to Scheduling Problem

The scheduling problem treated here is which PERT basically targets. A typical definition of this problem can be written as follows.

(1) The project consists of two or more works.

(2) As for each work, the necessary time to end the work, two or more amounts of the resource and sets of the predecessor work are given.

(3) The constraint conditions are

:All predecessor works of the work (to begin) end,

:Each total of the resource which the works operating at all time must not exceed the amount of the resource,

:Work is not interrupted until ending.

The problem is to optimize the time that is needed to do the entire project under these conditions.

In this system, the geometric models of each work and the frame (axes) to stuff with it will be made to replace this scheduling problem with the packing problem.

3.1 Geometric Model of Work

Each work of the project is expressed in the rectangle in general. Following shape lists will be given to the work which takes 4 days and needs 3 workers, and its shape is shown in **Fig.4(a)** .

n _list[1,2,3,4].
s _list[a,b,c,d].
e _list[b,c,d,a].
d _list[e,n,w,s].
l _list[4,3,4,3].

Even when it is necessary to execute work at once after another work end and certain when the number of work changes during work, it is possible to define the shape similarly. For instance, the work which needs 2 workers from the work beginning to the 2 days, 3 workers next 1 day and 4 workers last 2 days as follows and Fig.4(b) shows the geometric model of this work.

n _list[1,2,3,4,5,6,7,8,9,10].
s _list[a,b,c,d,e,f,g,h,i,j].
e _list[b,c,d,e,f,g,h,i,j,a].
d _list[e,e,e,n,w,s,w,n,w,s].
l _list[2,1,2,3,2,1,1,3,2,6].

3.2 Geometric Model of frame (Coordinate Axes)

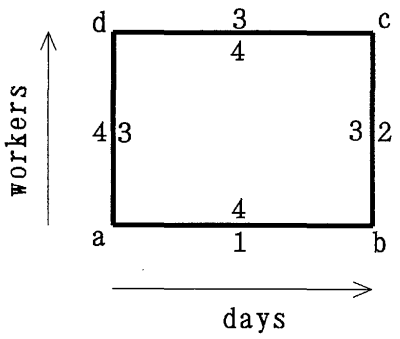
The coordinates axes (frame) to pack the box (work) in is expressed as a box without the width, shown in **Fig.5** and its lists are described as follows.

n _list[1 ,2 ,3 ,4 ,5 ,6].
s _list[x1,x2,x3,x4,x5,x6].
e _list[x2,x3,x4,x5,x6,x1].
d _list[e ,n ,w ,n ,w ,s].
l _list[∞,0,∞,∞,0,∞].

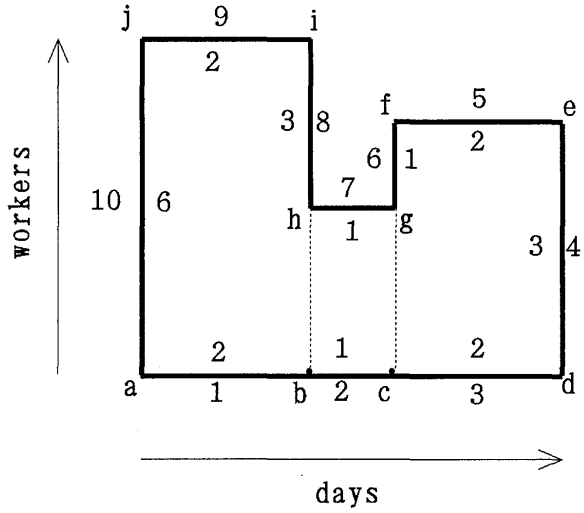
It is possible to process the operation to stuff the work in the frame as the composition of the shape by expressing the coordinate axes like this.

4. The Scheduling System

The system schedules the project by packing the boxes (works) in the flame according to the processing procedure described as follows, and **Fig.6** shows it's flow chart.



(a)



(b)

Fig.4 Geometric models for works
(a) basic shape model
(b) case of changing the number of workers

Processing 1:
Decide the priority levels of all works and advance to Processing 2.
(Method of deciding priority levels:
Request the total of necessary days of each work from itself to the final work end. It is called Earliest Finish Days. When the following work is plural, all the processes will be searched and the maximum value is used as its Earliest Finish Days.
The work which has the large Earliest Finish Days is given to priority. If the Earliest Finish Days are same for several works, the one with a short Necessary Days is given priority, moreover, the necessary days are same, then the one with little Necessary number of workers is given to priority.)

Processing 2:
Make the shape lists of frame and each work and advance to Processing 3.

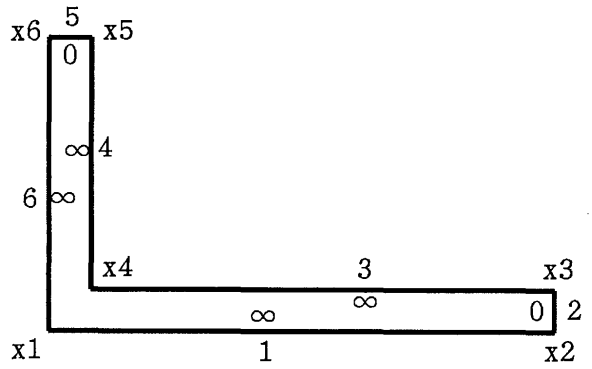


Fig.5 Geometric model for flame (coordinate axes)

Processing 3:
Stuff the work with the highest priority level in the frame.
Preserve the end time of the work.
Update the all lists of the frame, and advance to Processing 4.

Processing 4:
Decide the work to be packed according to the priority levels.
Examine the predecessor work of this work, if existing, advance to processing 6, or advance to Processing 5.

Processing 5:
Request the Vertex Candidates which are the elements of the *e_list* of the frame and can be packed with work.
Select the vertex from the Vertex Candidates which has minimum time, and satisfies the constraint conditions, when the work starts at the time.
Pack the work at the vertex.
Preserve the end time of this work.
Update the all lists of the frame after packing in a frame end.
Delete the making Vertex Candidates are deleted all and advance to Processing 4.

(Method to request the Vertex Candidates:
Take out the element *'w'* except the last from the *d_list* of the frame. Make the Vertex Candidates from the elements in the *e_list* which positions are same as the element *'w'* in the *d_list*. Moreover, take out the element of the *l_list* of the frame and calculate the candidate's time from the value.)

And when the work is stuffed at the vertex, take out the element 'e' from the *d_list* of the work. Then take out

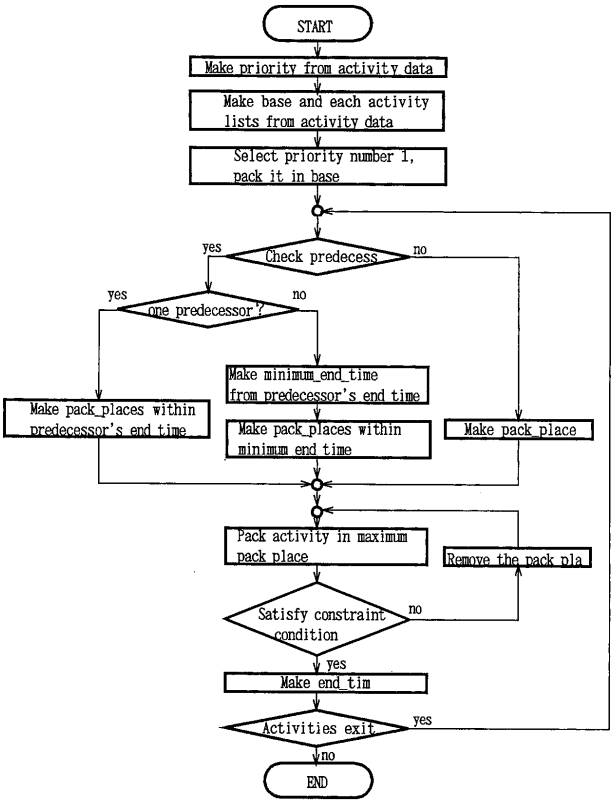


Fig.6 Procedure of the scheduling

the element from the *s__list* which position is same as the element 'e' in the *d__list*. Compose the frame and the work as the vertex and the candidate are suitable.

Processing 6:

If predecessor work of the work is one, advance to Processing 7.

If plural, advance to Processing 8.

Processing 7:

Select the vertex from the Vertex Candidates with the earliest time by which the predecessor work of the selected work was finished and pack the work at the vertex in the frame.

Following procedures are same as Pro.5.

Advance to Processing 9.

Processing 8:

Select the maximum value from the end times of the predecessor works.

Select the vertex from the Vertex Candidates with the earliest time by which the predecessor work with maximum time was finished and pack the work in a frame.

Following procedures are same as Pro.5.

Advance to Processing 9.

Table 1 Work lists

Activity name	Predecessor work	Necessary days	Necessary workers
A	——	3	4
B	——	5	3
C	B	4	1
D	A	4	2
E	A	7	3
F	C,D	5	3
G	E,F	4	5

Table 2 Priority levels obtained from table 1

Activity name	Earliest Finish Day	Priority levels
A	16	2
B	18	1
C	13	3
D	13	4
E	11	5
F	9	6
G	4	7

Processing 9:

If there is a work which is not packed yet, return to Processing 4, or a series of processing will be ended.

5. Example of Executing Computer

The example of executing the computer is shown. The project given by **table 1** is scheduled. The restriction of the number of work is assumed five persons. First of all, the priority level of each work is obtained (**Table 2**). Work B with the highest priority level is stuffed at x4 of frame (**Fig.7(a)**). Next, the Vertex Candidates are requested so that next *Work A* of the priority level may be stuffed in the frame.

The vertex candidates in this case become *[b2,b4]*. Because when *Work A* is stuffed at *b2*, the restriction condition will not be satisfied, therefore it will be stuffed at *b4* (Fig.(b)). Moreover, to stuff with Work C, the vertex candidate is requested. Then vertex candidates are *[a2,a4,b4]*, comparing with the end time of the predecessor *Work B* of *Work C*, *Work C* is stuffed at the vertex *a4* (Fig.(c)).The system keeps same procedure and the scheduled result is Fig.7(d).

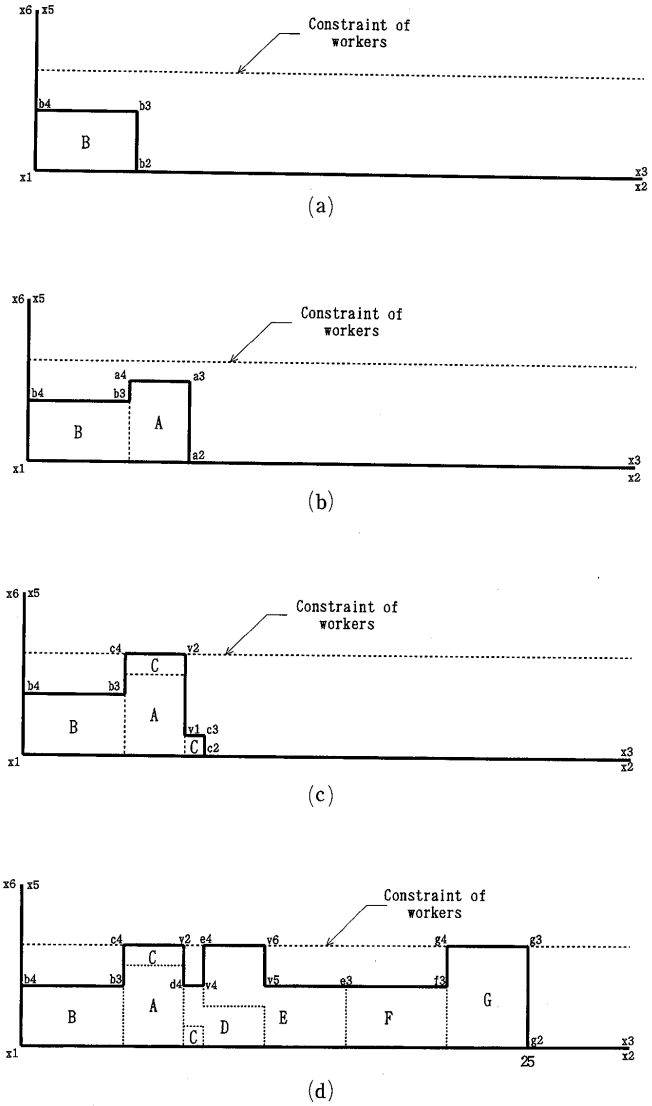


Fig.7 Scheduling procedure and result using shape processing
(a) after packed with Work B
(b) after packed with Work A
(c) after packed with Work C
(d) the result of scheduling

5.1 Comparison with PERT

The scheduling result using PERT of the project given by table 1 is shown in **Fig.8** . Its result is same as the result of Fig.7.

And another result of example which work lists are obtained from **Table 3** is shown in **Fig.9** . Figure (a) is the result using new method and (b) is the result using PERT.

Some examples including the example described above are examined, and these results show that the performances of them are not so difference.

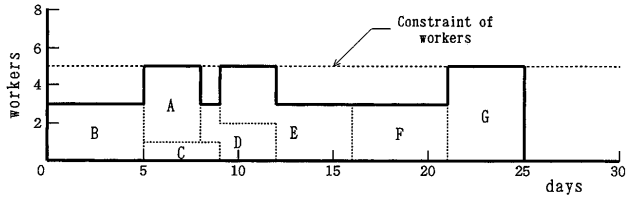


Fig.8 Scheduling result using PERT

Table 3 Work lists and priority levels

Activity name	Predecessor work	Necessary days	Necessary workers	Priority levels
A	—	5	2	2
B	—	3	3	1
C	A	4	1	4
D	A	3	2	6
E	B	4	1	3
F	C	3	5	7
G	D	2	3	8
H	E	5	3	5
I	F,G	3	2	10
J	H	4	2	9
K	I,J	4	3	11

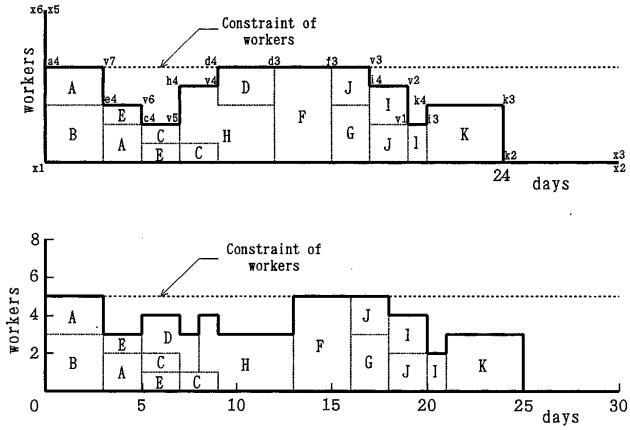


Fig.9 Scheduling results
(a) using shape processing
(b) using PERT

5.2 The Number f Worker Changing Problem

Figure 10 shows the result of scheduling the examples assumed that the work lists defined as **Table 4** under the constraint of six workers condition. The priorities of the Table are calculated according to the method of chapter 4. The geometric models of Fig.(a) show the each work of table, and Fig.(b) shows the scheduling result.

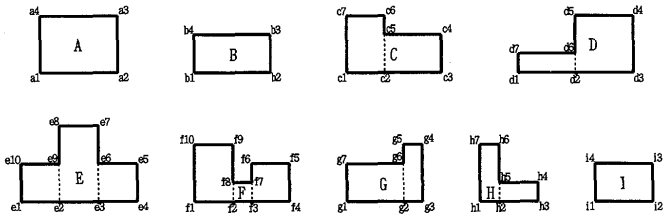
It was confirmed to be able to request the solution by replacing the scheduling problem that the number of worker changes in the work, with the packing problem. However, the result is not guaranteed to be the best one because this technique to depend on the method of

deciding the priority level fundamentally. It is considered that the priority level is not used for such problems absolutely but used as a standard, though the decision of the priority level is important in the scheduling problem.

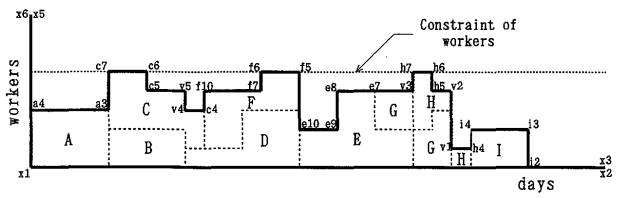
On the other hand, in this technique, it is considered that the solution method such as selecting the work packed next is devised by making the best use of the characteristic of the shape processing, for instance comparing the topology of the work with the it of the frame in this technique.

Table 4 Work lists and priority levels

Activity name	Predecessor works	Necessary days	Necessary workers	Priority levels
A	——	4	3	1
B	A	4	2	2
C	A	2	3	4
		3	2	
D	B	3	1	3
		3	3	
E	C,D	2	2	5
		2	4	
		2	2	
F	B	2	3	6
		1	1	
		2	2	
G	F	3	2	7
		1	3	
H	E	1	3	8
		2	1	
I	G,H	3	2	9



(a)



(b)

Fig.10 Scheduling result for the number of workers changing problem
(a) geometric models for works
(b) scheduling result

6. Conclusion

Up to now, various techniques have been proposed as a method of the scheduling problem. However, it is impossible in a current problem to request the best solution in limited time because there are so many solutions which satisfy the constraint conditions.

In this research, the technique of replacing PERT's scheduling problem with the packing problem, was proposed and the effectiveness was examined.

The effectiveness confirmed by using this technique is described to the following:

(1) Because the shape model is expressed by the some list, various shape processing can be done by easy list's operation. Therefore, the solution can be requested by easy lists operation by replacing the scheduling problem with the packing problem;

(2) For the number of worker changing problem which is difficult to request the solution in PERT and so on, the solution can be requested easily using shape processing;

(3) This method has possibility that the work which can interrupt on the way, of no permission in a lot of scheduling techniques, can be handled.

As mentioned above, the technique proposed in this research is confirmed to be very effective to the scheduling problem.

References

(1) Tomoaki Sekine: " PERT/CPM Introduction" Nikkagiren (1989).

(2) Toshirou Terano: " The System Engineering Introduction" Kyoritsu publication (1985) pp.230-pp.248.

(3) The Ministry of International Trade and Industry, The Council of industrial structure: " The Production Control" Nikkan industrial newspaper pp.190-pp.215.

(4) Michiharu Kudo:" An Interactive Scheduling Mechanism based on Fixed area" Proc.- The 41st IPSJ (1989) pp.243-pp.244.

(5) Michiharu Kudo: " A Constraint Oriented Interface for the Load-Balancing" Proc.- The 4th JSAI (1990) pp.445-pp.447.

(6) Johen de Kleer:"An Assumption-based TMS" Artificial Intelligence vol.28(1986) pp.127-pp.162.

(7)Nobuhiro Yukami, Hirotaka Hara: " A Method of Solving Scheduling Problem" association nationwide rally (the fourth times) (1990) pp.527-pp.530.

(8)Shuichi Fukuda"The Reliability Design Expert System" Maruzen (1990).