

# Maximizing Net Present Value for Generalized Resource Constrained Project Scheduling Problem

*Eiji Goto*<sup>1</sup>, *Takeshi Joko*<sup>2</sup>, *Katsuki Fujisawa*<sup>2</sup>, *Naoki Katoh*<sup>2</sup> and *Shuzo Furusaka*<sup>2</sup>

Nomura Research Institute, Japan<sup>1</sup>

Department of Architecture and Architectural Systems, Kyoto University, Japan<sup>2</sup>

## Abstract

Aiming at potential applicability to real project management, we consider in this paper a generalized class of multi-mode resource-constrained project scheduling problems that maximize the net present value of cash flows under the constraint that the completion time of the project does not exceed a given threshold. The problems we deal with in this paper generalize those studied in the literature so that the time duration of each activity varies depending on not only the mode chosen but the amount of resources used and the way of payment for each resource such as the bill duration and the cash ratio can be treated as a parameter. We propose a multi-pass metaheuristic algorithm for the problems that consists of two-stage tabu search. In order to see the effectiveness of the algorithm, we carry out computational experiments for more than 10,000 problem instances of relatively small size and for a real construction project with 58 activities and 34 resource types.

## 1 Introduction

### 1.1 Background and Motivation

In project scheduling literatures (e.g, see Willy et al. (1997), Rainer et al. (1997), Brucker et al (1999), Herroelen et al. (1999) and Weglarz (1999) for its overview), various models have been proposed depending on the way of representing activity relationships, activity characteristics, resource types and cost for determining an optimal schedule of a project. These models result in a number of project scheduling problems, which commonly focus on one of three types of objectives: time, cost, and net present value (NPV). For projects with a long time span such as construction projects, NPV seems the most appropriate as a performance criterion.

A numerous papers already have studied the models that focused on the objective maximizing NPV, and exact and heuristic algorithms have been proposed (e.g., Russell (1970), Grinold (1972), Yang et al. (1992), Icmeli and Erengüç (1996), Padman et al. (1997), Neumann and Zimmermann (1999a,1999b)). An extensive overview of the existing methods can be found in Brucker et al. (1999). However, the existing project scheduling models are not sufficient to model the real project management. In particular, since we are dealing with the objective of maximizing NPV, we need to determine the cost incurred for each time period, and thus we need to refine the conventional classification of resource types of renewable and non-renewable ones. As will be shown later, the way of determining such cost varies depending on the resource types such as labor, material and

equipment. This is one of the motivations of this paper.

In addition, no one has introduced general method which can comprehensively treat several types of project scheduling problems, although various project scheduling problems have been studied and a number of methods have been developed to solve these problems.

## 1.2 Research Purposes and Main Results

The major purposes of this paper are

- 1) to propose a new model for project scheduling that generalizes the existing method so that real construction projects can be dealt with, and
- 2) to produce a metaheuristic algorithm that effectively solves the proposed problem.

In the early stage of the research for resource-constrained project scheduling, only the single-mode case, (i.e., each activity has to be performed in exactly one predefined way) has been treated. In this case each activity may be characterized by a unique duration and a single collection of resource requirements that have to be met during the time period the activity is being processed. Recent efforts have been made to formulate and to solve the more general resource-constrained project scheduling problem with multiple modes (RCPSMM), where activity duration is a function of the mode selected. The problems we deal with in this paper generalize RCPSMM so that (1) the time duration of each activity varies depending on the amount of resources used even if the mode is fixed, and (2) the way of payments for each resource is taken into consideration. In this model, we define two types of ways for payments according to the type of the consumed resource, i.e., one by cash only, and the other by mixture of cash and bill<sup>1</sup>. Regarding the latter, cash ratio (i.e., the ratio of cash payment to the total one) and bill duration are treated as problem parameters. We consider in this paper such a generalized class of RCPSMM whose objective function is to maximizes the NPV of cash flows under the constraint that the completion time of the project does not exceed a given threshold, aiming at potential applicability to the real construction project management. Notice that we consider in this model the maximization of NPV from the standpoint of the project contractor, that is, cash inflows and outflows indicate the client's investments and the payments for consumed resources, respectively.

The new model which we present not only treats the condition of payments arising in the construction industry (typically of Japan) but also expands conventional concept of modes. The concept of conventional modes is regarded as the absolute element that determines the duration and the amount of consumed resources for an activity. But we regard it as the one of merely determining the way to complete an activity. In conventional project scheduling models activity duration is fixed for each mode. We take yet a different approach. In our model the way to complete an activity is treated by mode selection while the time duration of each activity for a fixed mode varies depending on the amount of labor resource used.

In addition, a traditional renewable resource is further classified into three categories, i.e., material, labor and equipment. Such classification is necessary because there are some differences among these resources in the way of determining the per period cost. The total amount of material resource is fixed for each activity regardless of the mode selection. For labor resources, the amount consumed

---

<sup>1</sup>The actual payments in cash are delayed to some extent by using the settlement by bill, which is traditionally adopted in Japan.

per period can be determined by fixing both its mode and eventual amount to be allocated, and then the duration of an activity is calculated according to the total amount of labor resources. Once the duration of an activity is fixed, we can determine per period amount of consumed material resources. On the other hand, per period amount of equipment resources is given by merely mode selection.

Furthermore the equipment resource is classified into time dependent cost (TDC) items and non-time dependent cost (non-TDC) in terms of cost allocation. TDC is a concept introduced by Gong (1997) which is defined as a part of the project costs depending on the activity duration in the project network (the details will be explained in Section 2). We refer to the resulting problem as the resource-constrained project scheduling problem with “variable duration modes” (RCPSPVDM). When activity duration is fixed for each mode, RCPSPVDM reduces to RCPSP with multiple modes, namely RCPSPMM.

We then propose a metaheuristic algorithm for the problem, which we call MPTS-TS (Multi-Pass Two Stage Tabu Search) method that consists of two-stage tabu search. This metaheuristic algorithm is very flexible so that problems with various objective functions and complex constraints can be treated in a unified manner. In the first stage, we generate a feasible schedule by what we call *initial tabu search*. The initial tabu search is developed by generalizing the heuristic proposed by Lee and Kim (1996). In the second stage, in order to further improve the solution obtained by the initial tabu search, we try to search not only feasible space but also infeasible space starting from the initial solution generated in the first stage. The second stage is repeated until no more improvement is possible. Then, the procedure returns to the first stage, generates a new feasible schedule and subsequently moves to the second stage. Each time the procedure returns to the first stage, a new “pass” is started. The metaheuristic procedure continues until no more improvement is possible. We have verified the excellent performance of MPTS-TS method by solving more than 10,000 problem instances belonging to the class RCPSPMM.

Finally, we apply the proposed model to a real construction project and optimize the project NPV by MPTS-TS method under several conditions in order to see the influence of such project conditions on the project cash flows.

The structure of this paper is as follows. In Section 2 we formulate the new project scheduling problem class, RCPSPVDM. In Section 3 we propose the new heuristic approach, MPTS-TS method. In Section 4 we briefly summarize the results of computational experiments applied to more than 10,000 problem instances of class RCPSPMM. We also report the computational result applied to the real construction project and investigate the influence of project conditions on cash flows of the project. Section 5 concludes this paper and mentions possible future work.

## 2 Problem Description

1) **Basic assumptions:** We first state the assumptions and define variables and terminology necessary to formulate the problem to be discussed in this paper. A project consists of  $J$  activities indexed from 1 to  $J$ . Activities 1 and  $J$  are dummy ones that represent the start and the completion of the project, respectively. The dummy activities do not require any time or resource for processing. There are precedence relations between certain activities due to technological requirement. No preemption is allowed, i.e., once an activity is started, it cannot be interrupted. The notation follows the activity-on-node format, i.e., nodes represent activities and arcs represent precedence relations among activities. It is assumed that a node receives a higher index than all of its preceding nodes.  $\mathcal{S}_j$ ,  $1 \leq j \leq J$ , denotes the set of immediate successors of activity  $j$ . Each activity  $j \in \{1, \dots, J\}$  can be executed in one mode chosen from the mode set  $\mathcal{M}_j = \{1, 2, \dots, M_j\}$ . In particular,  $M_1 = M_J = \{1\}$  is assumed by convention. The difference of the mode applied to an activity implies the difference of the method to complete it. In this model, we assume that the time duration for the project is measured in days and time period  $t$  denotes the  $t$ -th day from the beginning of the project.

2) **Definition of resource types:** Let  $\mathcal{R} = \{1, 2, \dots, K\}$  denote the set of renewable resources required to complete the project.  $\mathcal{R}$  is classified into three subclasses (subsets), that is, material ( $\mathcal{MR}$ ), labor ( $\mathcal{LR}$ ), and equipment ( $\mathcal{ER}$ ) resources. Furthermore we introduce a subclass  $\mathcal{ER}^T$  of  $\mathcal{ER}$ , *time dependent cost* (TDC) resources. The concept of TDC was introduced by Gong (1997). TDC is defined to be the product of “unit time cost” and “service time”. Here the service time of TDC resource is the time duration starting from the first use and ending at the last. In the construction project, for example, the rental equipment such as tower crane can be viewed as a TDC resource whose cost only depends on the service (rental) time but not on the actual days for which the equipment is used. The reduction of the waiting times of TDC resources, therefore, naturally reduces the time dependent cost. On the other hand, the equipment resources which do not depend on its service time are denoted by non-TDC resources ( $\mathcal{ER} \setminus \mathcal{ER}^{TDC}$ ). We introduce a virtual activity for each TDC resource as shown in Fig.1, that is, some activities which use TDC resource are regarded as one activity (virtual activity) whose processing time is equal to the actual execution time of activities involved in a virtual activity plus waiting times among them. We assume that each activity involved in a virtual activity consumes only one unit of TDC resource. Therefore an activity which uses multiple units of TDC resource is decomposed into several ones which have equivalent processing time so that each consumes only one unit of TDC resource. If several activities which use TDC

resources are executed in parallel, sufficient number of virtual activities are supposed to be provided so that activities belonging to the same virtual activity are not be executed simultaneously.

We use  $va_{k,\nu}$  to denote the virtual activity corresponding to TDC resource  $k$ , where  $\nu$  denotes the number of virtual activities which use TDC resource  $k$ .

3) **Resource vectors:** The amount of work of activity  $j$  and the material vector required to complete activity  $j$  are denoted by  $W_j$  and  $\mathbf{MR}_j$ , respectively. The  $k$ -th element of vector  $\mathbf{MR}_j$  represents the total amount of the material  $k \in \mathcal{MR}$  allocated to activity  $j$ . It is assumed that  $\mathbf{MR}_j$  is independent of the choice of the mode. Once the mode of activity  $j$  is fixed to  $m_j$ , two types of resource vectors  $\mathbf{L}_{jm_j}$  and  $\mathbf{E}_{jm_j}$  are determined. Labor vector  $\mathbf{L}_{jm_j}$  is called *basic crew*

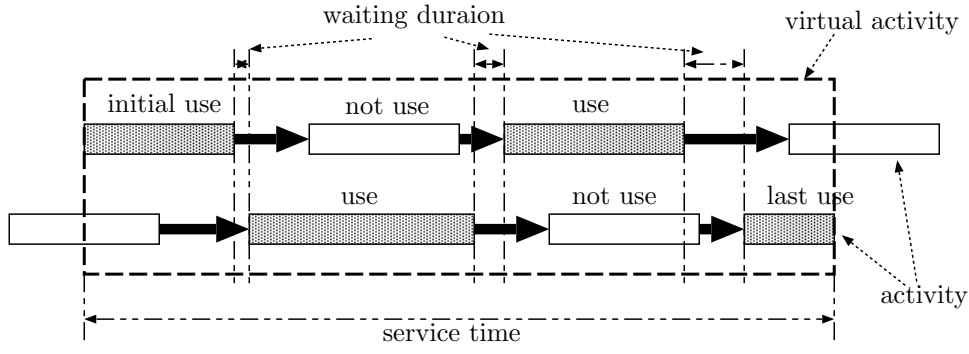


Fig. 1: The concept of TDC resources

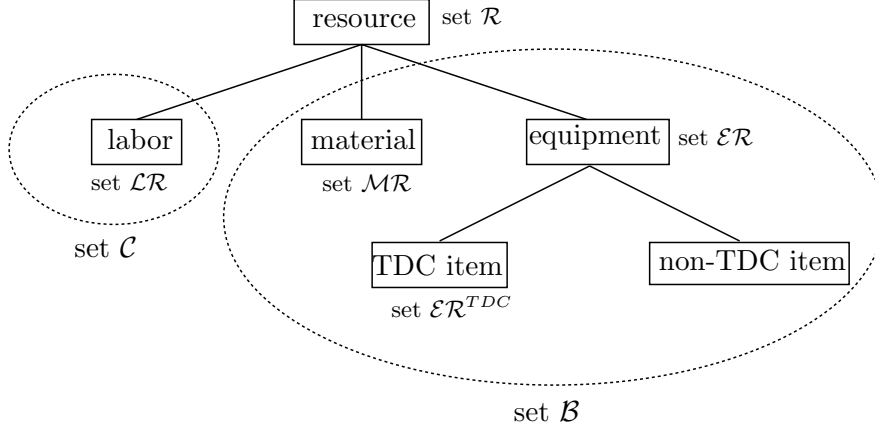


Fig. 2: The concept of resource types

*formation vector*, which represents the combination of the minimum amount of per period labor resources necessary to execute activity  $j$  in mode  $m_j$ .  $\mathbf{E}_{jm_j}$  is equipment vector which represents the amount of per period equipments allocated to activity  $j$  in mode  $m_j$ . The basic productivity  $bp_{jm_j}$  is associated with the basic crew formation vector  $\mathbf{L}_{jm_j}$  which is used to determine the time duration of activity  $j$ . The actual amount of labor resources used for activity  $j$  is a multiple of  $\mathbf{L}_{jm_j}$ , and such multiple is denoted by  $y_{jm_j}$ .

4) **Activity duration:** We assume that  $y_{jm_j}$  satisfies the following condition.

$$y_{jm_j}^{\min} \leq y_{jm_j} \leq y_{jm_j}^{\max}, \quad (1)$$

where  $y_{jm_j}^{\min}$  and  $y_{jm_j}^{\max}$  are given positive integers. Multiplying  $y_{jm_j}$  by the basic productivity  $bp_{jm_j}$  results in the real productivity  $P_{jm_j}(y_{jm_j})$  of activity  $j$  in mode  $m_j$ .

$$P_{jm_j}(y_{jm_j}) = y_{jm_j} \cdot bp_{jm_j}. \quad (2)$$

The duration  $d_j(y_{jm_j})$  of activity  $j$  becomes shortened when  $y_{jm_j}$  increases. The activity duration  $d_j(y_{jm_j})$  of activity  $j$  is then determined below.

$$d_j(y_{jm_j}) = \left\lceil \frac{W_j}{P_{jm_j}(y_{jm_j})} \right\rceil. \quad (3)$$

Here  $\lceil a \rceil$  for real  $a$  stands for the smallest integer greater than or equal to  $a$ . Recalling that activity duration is measured in the number of days, taking the integer part in (3) is necessary.

From (1) and (2),  $d_j(y_{jm_j})$  satisfies

$$d_j(y_{jm_j}^{\max}) \leq d_j(y_{jm_j}) \leq d_j(y_{jm_j}^{\min}). \quad (4)$$

5) **Amount of consumed resources:** For each resource  $k \in \mathcal{R}$ , we assume that  $R_k$  units are available in each period. The per period resource usage vector  $\mathbf{r}_{jm_j}(y_{jm_j})$  is defined as follows. The  $k$ -th element of  $\mathbf{r}_{jm_j}(y_{jm_j})$  denotes the per period usage of resource  $k$  allocated to execute activity  $j$  in mode  $m_j$ .

$$\mathbf{r}_{jm_j}(y_{jm_j})[k] = \begin{cases} \frac{1}{d_j(y_{jm_j})} \mathbf{M}\mathbf{R}_j[k] & k \in \mathcal{MR}, \\ y_{jm_j} \mathbf{L}_{jm_j}[k] & k \in \mathcal{LR}, \\ \mathbf{E}_{jm_j}[k] & k \in \mathcal{ER}. \end{cases} \quad (5)$$

We define the following 0-1 variable  $x_{jm_j}$  to formulate our problem.

$$x_{jm_j} = \begin{cases} 1 & \text{when activity } j \text{ is executed in mode } m_j, \\ 0 & \text{otherwise.} \end{cases}$$

The total per period usage of TDC resource  $k$  for  $k \in \mathcal{ER}^{TDC}$  in period  $t$  is defined by

$$q_{kt}^{TDC} = \begin{cases} \mathbf{E}_{jm_j}[k], & k \in \mathcal{ER}^{TDC}, va_{k\nu} \in \mathcal{SA}_t, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where  $\mathcal{SA}_t$  is the set of activities (including the virtual activities) that are in progress in period  $t$ .

The total per period usage of resource  $k$ , in period  $t$ , is denoted by  $q_{kt}$  and is determined by

$$q_{kt} = \begin{cases} \sum_{j \in \mathcal{SA}_t} \sum_{m_j=1}^{M_j} \mathbf{r}_{jm_j}(y_{jm_j})[k] \cdot x_{jm_j} & k \in \mathcal{R} \setminus \mathcal{ER}^{TDC}, \\ q_{kt}^{TDC} & k \in \mathcal{ER}^{TDC}. \end{cases} \quad (7)$$

6) **Definition of due date and tardiness penalty:** The project is assumed to have a predetermined due date,  $\bar{T} > 0$ . If the completion time of the project exceeds  $\bar{T}$ , penalty cost  $\bar{P} (> 0)$  is incurred for each unit time of delay. If the project due date cannot be violated, then  $\bar{P}$  is set to a sufficiently large value.

7) **Definition of cash flow:**  $c_k$  denotes the cost incurred by using one unit of resource  $k$  per period. We assume that  $L$  months are sufficient to complete the project, i.e., the length of  $30L$  days is an

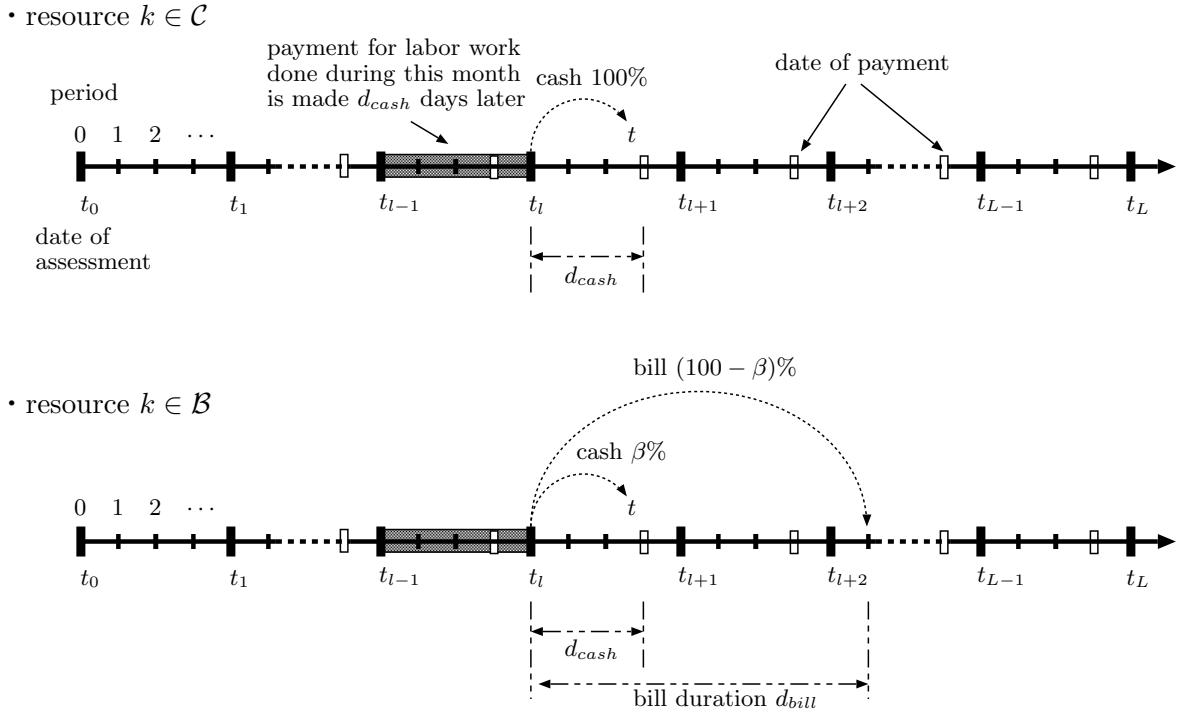


Fig. 3: The concept of the condition of payments

upper bound for the length of any feasible (or meaningful) schedule of the project. Let  $t_i$  denote the last day of the  $i$ -th month. For notational convenience, let  $t_0$  denote the 0-th day. We assume in this paper that any schedule of the project starts from the date  $t_0 + 1$ . Let  $c_{kl}^m$  denote the total cost of resource  $k$  during the  $l$ -th month  $[t_{l-1} + 1, t_l]$ . Payment for all kinds of resources is assumed to be made on monthly basis. From the viewpoint of payment conditions, the set of resources is classified into two subsets  $\mathcal{C}$  and  $\mathcal{B}$ ; in our model, we assume that the payment for resources in subset  $\mathcal{C}$  is done by cash while that for resources in subset  $\mathcal{B}$  is done by the mixture of cash and bill. It is assumed that we are allowed to use the settlement by bill as the way of payment for the cost incurred by the use of resources in  $\mathcal{B}$ . The former corresponds to the labor resources and the latter the other types of resources. This is typical in construction industries of Japan.

All payments for labor are made in cash  $d_{cash}$  days later from the end of the corresponding month, (commonly 15 days in Japanese construction industries) while for other resources (materials and equipments)  $\beta\%$  of the whole payment is done  $d_{cash}$  days later in cash while  $(100 - \beta)\%$  is done  $d_{bill}$  days later (commonly multiple of 30 days in Japanese construction industry) (see Fig. 3), where  $\beta$  is the cash ratio. Therefore the payment for the consumed resources is made even after all the activities are completed. Let  $T$  denote the completion date of the last payment, which is described as follows.

$$T = \begin{cases} t_L + d_{cash} & \text{if the last payment is made by cash only,} \\ t_L + \max\{d_{bill}, d_{cash}\} & \text{otherwise.} \end{cases} \quad (8)$$

Given  $q_{kt}$  of (7), per period cost  $c_{kt}$  of resource  $k \in \mathcal{R}$  for period  $t$  is determined by

$$c_{kt} = c_k q_{kt}. \quad (9)$$

The monthly cost  $c_{kl}^m$  is then

$$c_{kl}^m = \sum_{t=t_{l-1}+1}^{t_l} c_{kt} \quad (k \in \mathcal{R}, l = 1, \dots, L). \quad (10)$$

The cash position  $cp_t$  at period  $t$  is determined as follows. We assume that the client pays in  $p$  phases at time  $\tau_1, \tau_2, \dots, \tau_p$  with the ratios  $\rho_1, \rho_2, \dots, \rho_p$  satisfying  $\rho_i \geq 0$  and  $\sum_{i=1}^p \rho_i = 1$ . In the construction industry of Japan, the client typically pays equally in three phases, i.e., the beginning, the middle, and the completion time of the project. We define the total investment of the client as  $\bar{c}$ .

(i) If period  $t$  is the date of payment in cash,

$$cp_t = cp_{t-1} - \sum_{k \in C} c_{kl}^m - \sum_{k \in B} c_{kl}^m \cdot \frac{\beta}{100} \quad (t = t_l + d_{cash}; l = 1, \dots, L), \quad (11)$$

(ii) If period  $t$  is the date of payment in bill,

$$cp_t = cp_{t-1} - \sum_{k \in B} c_{kl}^m \cdot \left(1 - \frac{\beta}{100}\right) \quad (t = t_l + d_{bill}; l = 1, \dots, L), \quad (12)$$

(iii) If period  $t$  is the date of investment,

$$cp_t = cp_{t-1} + \rho_i \bar{c} \quad \left(\sum_{i=1}^p \rho_i = 1, \rho_i \geq 0, t = \tau_i; i = 1, \dots, p\right), \quad (13)$$

(iv) otherwise

$$cp_t = cp_{t-1}. \quad (14)$$

8) **Definition of the project NPV:** Based on the previous terminology, the project net present value  $NPV$  is described as follows.

$$NPV = \sum_{t=1}^T (cp_t - cp_{t-1}) w_t + cp_0. \quad (15)$$

Here  $w_t$  denotes the discounted cash rate defined by  $w_t = \{1/(1+r)\}^{t-1}$ , where  $r$  is the interest rate, and  $cp_0$  is the advance from the client at  $t_0$ .

9) **Problem formulation:** The objective is to find a feasible schedule that maximizes the sum of net present values of cash inflows, outflows and tardiness penalty. For the formulation of the problem, we introduce additional decision variables  $f_j$  with  $j = 1, \dots, J$  to denote the finish time of



activity  $j$ . Accordingly,  $f_J$  denotes the completion time of the project. A schedule for the problem class of RCPSPVDM consists of three components; i.e., the execute mode, duration and finish time for each activity. Therefore the solution for this problem is expressed as a vector consisting of these variables  $x_{jm_j}$ ,  $y_{jm_j}$ ,  $f_j$ , and is denoted by  $\mathbf{x}$ . The term  $z$  denotes the tardiness of the project, and is defined as  $\max\{0, f_J - \bar{T}\}$ .

In this problem, we are trying to find  $\mathbf{x}$  that maximizes the objective value. The problem discussed in this paper is then stated as follows:

$$\mathbf{RCPSPVDM:} \quad \text{maximize} \quad c(\mathbf{x}) = NPV - \bar{P} \cdot z \quad (16)$$

$$\text{subject to} \quad x_{jm_j} \in \{0, 1\} \quad \text{for all } j, m_j, \quad (17)$$

$$\sum_{m_j=1}^{M_j} x_{jm_j} = 1 \quad \text{for all } j, \quad (18)$$

$$y_{jm_j}^{\min} \leq y_{jm_j} \leq y_{jm_j}^{\max} \quad \text{for all } j \text{ and } m_j, \quad (19)$$

$$f_j \leq f_i - d_i(y_{im_i}) \quad \text{for all } i \in \mathcal{S}_j, j = 1, \dots, J-1, \quad (20)$$

$$q_{kt} \leq R_k \quad \text{for all } k \text{ and } t, \quad (21)$$

$$z = \max\{0, f_J - \bar{T}\}, \quad (22)$$

$$f_1 = 0, \quad (23)$$

$$d_1(y_{1m_1}) = 0, \quad \text{for any } m_1 \in \mathcal{M}_1, \quad (24)$$

$$d_J(y_{Jm_J}) = 0, \quad \text{for any } m_J \in \mathcal{M}_J. \quad (25)$$

We first determine the mode for each activity. Since each activity is executed in exactly one mode, constraint (18) is imposed. Secondly, we determine the actual amount of labor resource for the selected mode, which is done by choosing  $y_{jm_j}$  satisfying (19).

After this, the amount of consumed resources and the duration for each activity are fixed. We then determine the finish time of each activity so as to satisfy (20) and (21). Notations we use in this paper are summarized in Table 1.

Table 1: Definition of Notations

| Legend                           | Definition   |
|----------------------------------|--|
| $j$                              | activity index, $j = 1, \dots, J$  |
| $M_j$                            | number of modes available for activity $j$   |
| $m_j$                            | $m_j$ -th mode of activity $j$   |
| $x_{jm_j}$                       | 0-1 integer variable; 1 if activity $j$ is executed by mode $m_j$ , 0 otherwise  |
| $f_j$                            | finish time of activity $j$  |
| $\mathcal{S}_j$                  | set of immediate successors of activity $j$  |
| $K$                              | number of resource types required for the project  |
| $\mathcal{MR}$                   | set of material resources  |
| $\mathcal{LR}$                   | set of labor resources   |
| $\mathcal{ER}$                   | set of equipment resources   |
| $\mathcal{ER}^{TDC}$             | set of TDC equipment resources   |
| $va_{k_i}$                       | $i$ -th virtual activity for TDC equipment resource $k$  |
| $W_j$                            | amount of work to complete activity $j$  |
| $\mathbf{MR}_j$                  | material vector of activity whose $k$ -th component stands for the total amount of the material $k$ allocated to activity $j$                                      |
| $\mathbf{L}_{jm_j}$              | basic crew formation vector which represents the combination of the minimum allowable per period amount of labor resources allocated to activity $j$ in mode $m_j$ |
| $\mathbf{E}_{jm_j}$              | equipment resource vector which represents the per period amounts allocated to activity $j$ in mode $m_j$  |
| $y_{jm_j}$                       | units of the basic crew formation allocated to activity $j$  |
| $y_{jm_j}^{\min}$                | lower bound of $y_{jm_j}$  |
| $y_{jm_j}^{\max}$                | upper bound of $y_{jm_j}$  |
| $bp_{jm_j}$                      | basic productivity for the basic crew formation vector $\mathbf{L}_{jm_j}$   |
| $P_{jm_j}(y_{jm_j})$             | real productivity of activity $j$ in mode $m_j$  |
| $d_j(y_{jm_j})$                  | duration of activity $j$ in mode $m_j$ when $y_{jm_j}$ units of basic crew formations are allocated  |
| $R_k$                            | upper bound of amount of resource $k$ available per period   |
| $\mathbf{r}_{jm_j}(y_{jm_j})[k]$ | per period usage of resource $k$ of activity $j$ in mode $m_j$ , when $y_{jm_j}$ units of basic crew formation are allocated                                       |
| period $t$                       | $t$ -th time interval ( $t$ -th day)   |
| $L$                              | number of months sufficient to complete the project  |
| $t_l$                            | last date of $l$ -th month $l = 1, \dots, L$   |
| $q_{kt}$                         | total per period usage of resource $k$ for period $t$  |
| $q_{kt}^{TDC}$                   | per period usage of TDC resource $k \in \mathcal{ER}^{TDC}$ for period $t$   |
| $\mathcal{SA}_t$                 | set of activities that are in progress in period $t$ ,   |
| $c_k$                            | per period cost for unit usage of resource $k$   |
| $c_{kl}^m$                       | total cost of resource $k$ for the $l$ -th month   |

|               |   |
|---------------|---|
| $\mathcal{C}$ | set of resources for which payment should be made in cash only          |
| $\mathcal{B}$ | set of resources for which payment should be made in both cash and bill |
| $d_{cash}$    | duration from the date of assessment to the date of payment in cash     |
| $d_{bill}$    | bill duration   |
| $\beta$       | cash ratio of payments  |
| $cp_t$        | cash position at period $t$   |
| $\bar{c}$     | total investment of owner   |
| $NPV$         | project net present value   |
| $w_t$         | discounted cash rate  |
| $r$           | interest rate   |
| $T$           | date of the last payment  |
| $\bar{T}$     | project due date  |
| $\bar{P}$     | penalty incurred by the unit time delay for the due date $\bar{T}$      |
| $z$           | tardiness of the project  |

---

## 3 Search Algorithm

### 3.1 General framework of procedure of MPTS-TS method

We propose a metaheuristic procedure for the problem defined in the previous section. Our heuristic procedure consists of two stages; the initial tabu search and the main tabu search. The initial tabu search generates a feasible and acceptably good solution. The main tabu search tries to search the solution space to improve the schedule obtained by the first stage. The fundamental framework of our two-stage algorithm is similar to the one by Taeho et al. (1998), although the heuristics method they propose is not based on tabu search algorithm. In order to improve the schedule in the main tabu search we define five types of neighborhoods and introduce appropriate operations concerning each neighborhood. The main tabu search relaxes the resource constraints (21) by introducing the penalty term concerning the excess usage of resources. The purpose of such treatment is to enable us to search more diverse solution space. In the main tabu search we introduce two other ingredients; the free penalty search and the oscillation of tabu lengths. The main purpose of them is to further enhance the ability to explore diverse solution space.

The free penalty search is an expansion of the penalty function approach, and is implemented by disregarding the penalty term concerning resource usage. The search is invoked when no improvement has been made for a certain number of main search iterations. The incumbent solution is not updated during this search.

Oscillation of tabu lengths is a technique to implement the diversification by controlling the tabu length: When a certain parameter of activity  $j^*$  is changed,  $tabulength[j^*]$  is increased by one and  $tabulength[j]$  for other activities  $j$  are decreased by a predetermined value  $\bar{\beta}$ . Here, we set  $\bar{\beta} = 1/(2 \times J)$ . The effectiveness of these two ingredients has been verified in Goto (1999).

The main tabu search is repeated until no improvement is possible for a certain number of iterations. Then, the procedure returns to the first stage in which the initial tabu search is repeated in order to generate a new feasible solution. Each time the procedure goes back to the first stage, a new “pass” is started. When the procedure returns to the first stage, the current solution is slightly modified by using the information stored in a two-dimensional array, M\_INDEX introduced by Taeho et al. (1998) which is a device to generate a diverse initial solution for each pass. The entire procedure terminates when no improvement has been made for a certain number of iterations. The overall structure of MPTS-TS is illustrated in Fig 4.

We define the following subroutines to describe the top-level procedure of MPTS-TS method.

|                            |   |
|----------------------------|---|
| INITIALIZE() :             | Initialize the parameters of each activity to construct an initial solution for the first pass,                                       |
| NPV( $\mathbf{x}$ ) :      | Return the project NPV of the solution $\mathbf{x}$ ,   |
| I-TS( $\mathbf{x}$ ) :     | Execute initial tabu search starting with a given initial solution $\mathbf{x}$ ,<br>( $x$ is assumed to be empty in the first pass), |
| M-TS( $\mathbf{x}$ ) :     | Execute main tabu search starting with the solution $\mathbf{x}$<br>which is obtained from initial tabu search,                       |
| SET_MODE( $\mathbf{x}$ ) : | Change modes for a predetermined number of<br>activities using M_INDEX.   |

Using this terminology, the high-level description of MPTS-TS method is given in Fig.5.

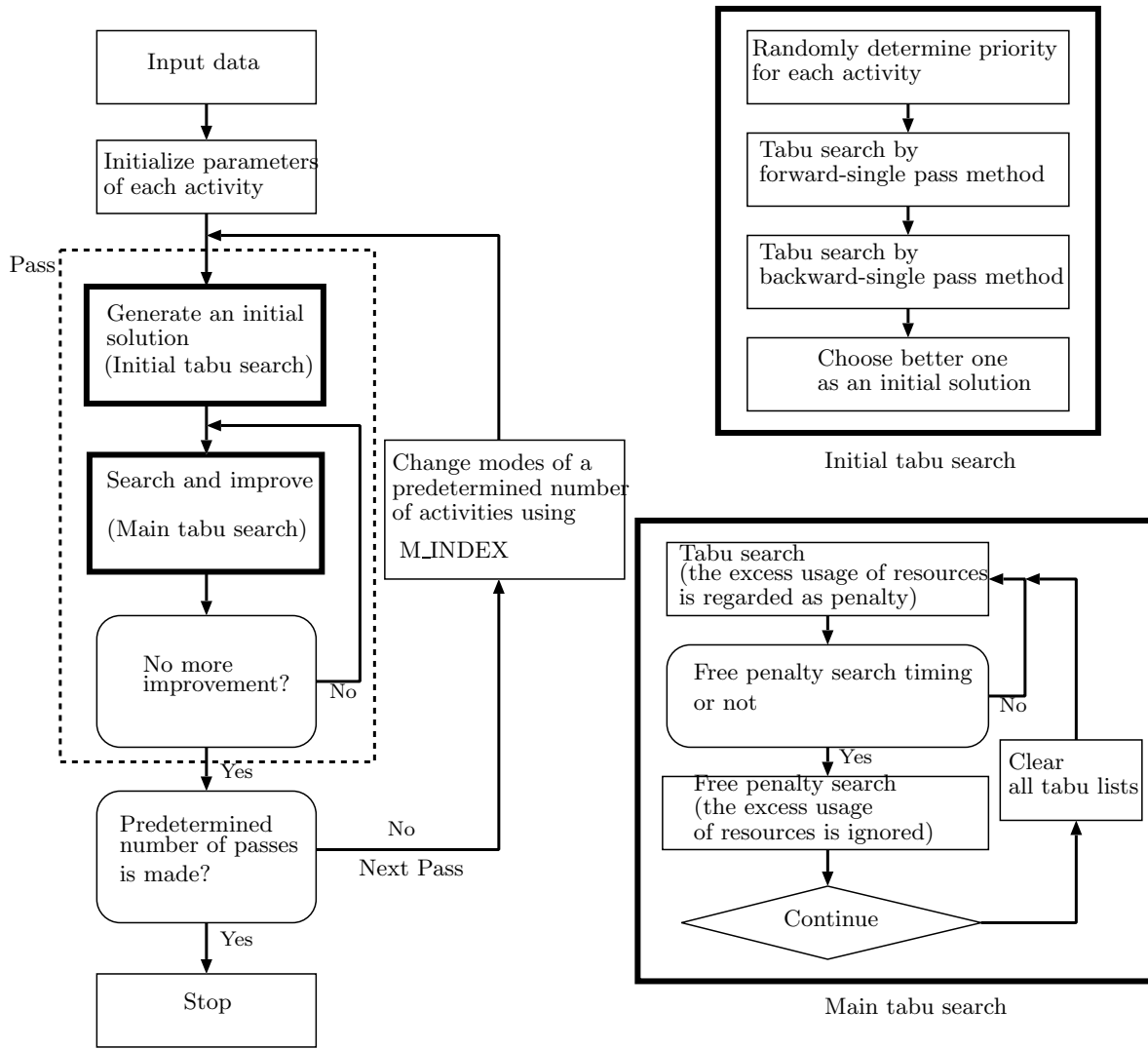


Fig. 4: The overall structure of MPTS-TS

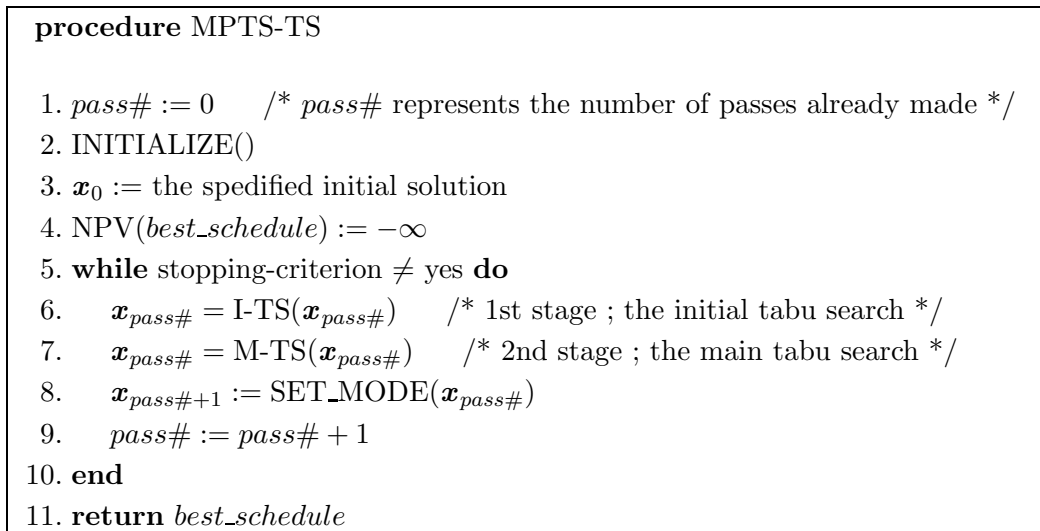


Fig. 5: MPTS-TS for RCPSPVDM

### 3.2 Initial tabu search

In order to construct an initial feasible schedule for the first pass, we first select (initialize) a mode and a duration for each activity in the following way: For each mode of each activity, the amounts of resources used are set as large as possible so that the activity duration of each mode becomes shortest, and then the mode with the least duration is selected. Once the mode and the duration of each activity is fixed, the resulting problem reduces to a single-mode RCPSP with NPV maximization as its objective.

For this problem, the initial tabu search is used, which is based on the heuristic developed by Lee and Kim (1996). It then uses the priority scheduling to resolve conflicts among activities competing for common resources. The initial tabu search consists of two phases.

Priority scheduling assigns a priority to each activity, which is randomly selected between 0 and 1. Then, activities are considered one by one in the order of their priorities to determine their start and finish times subject to precedence and resource constraints by forward or backward single-pass method. The forward (resp. backward) single-pass method chooses the one with highest priority among activities whose successors (resp. predecessors) have been all scheduled.

In the priority scheduling, neighborhood solutions are obtained by perturbing the priorities among some activities. In the method of Lee and Kim (1996), a pair of activities are randomly selected and their priorities are exchanged. On the other hand, our procedure chooses a pair of activities such that one is on the critical path and the other is not, and exchanges their priorities. It then performs the forward or the backward single-pass method to generate a neighbourhood solution. The neighborhood  $\mathcal{N}_{ex}(\mathbf{x})$  for a given solution  $\mathbf{x}$  is defined as the set of such solutions. A procedure for forward single-pass method is given below (the description of backward single-pass method is omitted since it is similar to the forward single-pass method).

**Procedure:** priority scheduling

**Step 1.** Determine priorities of activities. Set the current time  $t_{current}$  to be 0. Initialize the resource availability vector  $\mathcal{RA}$ , whose  $i$ -th component denotes the remaining amount of resource  $i$  available at  $t_{current}$ .

**Step 2.** If all the activities are scheduled, stop. Otherwise, let  $\mathcal{E}$  be the set of eligible activities, i.e., activities whose predecessors have all been completed by  $t_{current}$  and which require no more than  $\mathcal{RA}_i$  units of type- $i$  resource for all  $i$ . If  $\mathcal{E}$  is empty, go to Step 3. Otherwise, go to Step 4.

**Step 3.** Let  $t_{current}$  be the earliest time at which one of activities that are currently being processed becomes completed. After updating the resource availability vector, return to Step 2.

**Step 4.** Select an activity with the highest priority among those in  $\mathcal{E}$ . Determine its start and finish times and update the resource availability vector. Return to Step 2.

The initial tabu search generates two candidates of an initial solution, each by forward and backward single-pass methods respectively, and selects the better one as an initial solution.

Letting

$f\_exchange(\mathbf{x}, i, j) =$  schedule  $\mathbf{x}'$  obtained from  $\mathbf{x}$  by forward single-pass method after the priorities of activities  $i$  and  $j$  are exchanged in solution  $\mathbf{x}$ , and  
 $b\_exchange(\mathbf{x}, i, j) =$  schedule  $\mathbf{x}'$  obtained from  $\mathbf{x}$  by backward single-pass method after the priorities of activities  $i$  and  $j$  are exchanged in solution  $\mathbf{x}$ ,

forward and backward single-pass methods use the following neighborhoods  $\mathcal{N}_{ex}^{for}(\mathbf{x})$  and  $\mathcal{N}_{ex}^{back}(\mathbf{x})$ , respectively.

$$\mathcal{N}_{ex}^{for}(\mathbf{x}) = \{f\_exchange(\mathbf{x}, i, j) : j \in S_{critical}(\mathbf{x}), \text{ and } |j - i| \leq \bar{\alpha}n_i\}, \quad (26)$$

$$\mathcal{N}_{ex}^{back}(\mathbf{x}) = \{b\_exchange(\mathbf{x}, i, j) : j \in S_{critical}(\mathbf{x}), \text{ and } |j - i| \leq \bar{\alpha}n_i\}, \quad (27)$$

where  $S_{critical}(\mathbf{x})$  denotes the set of activities on the critical path of the solution  $\mathbf{x}$ ,  $\bar{\alpha}$  denotes a parameter with  $0 \leq \bar{\alpha} \leq 1$  that controls the range of search space, and  $n_i$  denotes  $\max\{|Predecess_i|, |Success_i|\}$ , where  $Predecess_i$  (resp.  $Success_i$ ) represents the set of all predecessors (resp. successors) of activity  $i$ .

Forward and backward single-pass methods compute the best solution in  $\mathcal{N}_{ex}^{for}(\mathbf{x}) \setminus ITL$  and  $\mathcal{N}_{ex}^{back}(\mathbf{x}) \setminus ITL$ , respectively, where  $ITL$  represents the tabu list for the initial tabu search. The tabu list  $ITL$  consists of pairs of activity numbers whose priorities have been exchanged. The procedure of initial tabu search is illustrated in Fig. 6.



**procedure** I-TS( $\mathbf{x}$ )

```
1.  $\mathbf{x}_0 := \mathbf{x}$  /*  $\mathbf{x}$  represents a specified initial solution */
2.  $it := 0$  /*  $it$  represents the number of iteration */
3.  $NPV(for\_best\_schedule) := -\infty$ 
4.  $ITL := \emptyset$  /*  $ITL$  represents the tabu list for I-TS */
5.  $tabulength :=$  a positive integer
6. while stopping-criterion  $\neq$  yes do /* the forward single-pass method phase */
7.   Let  $\mathbf{x}_{it+1}$  be the best solution in  $\mathcal{N}_{ex}^{for}(\mathbf{x}) \setminus ITL$ 
8.   if  $NPV(\mathbf{x}_{it+1}) > NPV(for\_best\_schedule)$  then
9.      $for\_best\_schedule := \mathbf{x}_{it+1}$ 
10.  Update  $ITL$ 
11.   $it := it + 1$ 
12. end
13.  $\mathbf{x}_0 := \mathbf{x}$  /*  $\mathbf{x}$  represents a specified initial solution */
14.  $it := 0$  /*  $t$  represents the number of iterations */
15.  $NPV(back\_best\_schedule) := -\infty$ 
16.  $ITL := \emptyset$  /*  $ITL$  represents the tabu list for I-TS */
17. while stopping-criterion  $\neq$  yes do /* the backward single-pass method phase */
18.  Let  $\mathbf{x}_{it+1}$  be the best solution in  $\mathcal{N}_{ex}^{back}(\mathbf{x}) \setminus ITL$ 
19.  if  $NPV(\mathbf{x}_{t+1}) > NPV(back\_best\_schedule)$  then
20.     $back\_best\_schedule := \mathbf{x}_{it+1}$ 
21.  Update  $ITL$ 
22.   $it := it + 1$ 
23. end
24.  $initial\_best := \arg \max\{NPV(for\_best\_schedule), NPV(back\_best\_schedule)\}$ 
25. return  $initial\_best$ 
```

Fig. 6: The initial tabu search (first stage) of MPTS-TS

### 3.3 Main tabu search

Main tabu search is started based on the solution  $\mathbf{x}$  obtained from the initial tabu search. The algorithm we propose is to allow us to generate solutions that may violate some of the resource constraints. For this purpose we use the penalty function approach that relaxes the resource constraints (21). The penalty is defined in terms of the distance between the set of feasible solutions  $X$  and an infeasible solution  $\bar{\mathbf{x}}$  denoted by distance  $d(\bar{\mathbf{x}}, X)$ . In MPTS-TS, the penalty term is determined as a function of amount of resources that are over-allocated. Such over-allocation  $E_k$  of resource  $k$  is defined as follows:

$$E_k = \sum_{t=0}^{f_j} \max\{0, q_{kt} - R_k\}, \quad k \in \mathcal{R}, \quad (28)$$

and  $d(\bar{\mathbf{x}}, X)$  is defined by

$$d(\bar{\mathbf{x}}, X) = \sum_{k \in \mathcal{R}} E_k. \quad (29)$$

We then define the penalty function  $p(\bar{\mathbf{x}})$  as follows:

$$p(\bar{\mathbf{x}}) = c(\bar{\mathbf{x}}) - \alpha \cdot d(\bar{\mathbf{x}}, X), \quad (30)$$

where  $\alpha$  is a positive constant.

Now, the problem to be solved by our heuristic algorithm is reformulated as below.

#### RCPSPVDM':

$$\begin{array}{ll} \text{maximize} & p(\mathbf{x}) \\ \text{subject to} & (17), (18), (19), (20), (22), (23), \text{ and } (24). \end{array}$$

For the explanation of the main tabu search, we introduce additional notations below.

$$\begin{aligned} \mathcal{P}_j &= \text{set of immediate predecessors of activity } j, \\ \text{EST}_j(\mathbf{x}) &= \max\{f_h : h \in \mathcal{P}_j\}, \quad j = 1, \dots, J, \quad \mathbf{x} \in X, \\ \text{LFT}_j(\mathbf{x}) &= \min\{s_h : h \in \mathcal{S}_j\}, \quad j = 1, \dots, J, \quad \mathbf{x} \in X, \\ \mathcal{M}_j^{\text{sc}}(\mathbf{x}) &= \{m_j \in \mathcal{M}_j \mid \text{the change of the current mode for activity } j \text{ causes the} \\ &\quad \text{violation of precedence constraints when the start time of } j \text{ is fixed} \\ &\quad \text{as in the current schedule}\}, \\ \mathcal{M}_j^{\text{fc}}(\mathbf{x}) &= \{m_j \in \mathcal{M}_j \mid \text{the change of the current mode for activity } j \text{ causes the} \\ &\quad \text{violation of precedence constraints when the finish time of } j \text{ is fixed} \\ &\quad \text{as in the current schedule}\}. \end{aligned}$$

The main tabu search uses the neighborhood  $\mathcal{N}_{\text{move}}(\mathbf{x})$ , where  $\mathcal{N}_{\text{move}}(\mathbf{x})$  consists of  $\mathcal{N}_{\text{move}}(\mathbf{x}, j)$ ,  $j = 2, \dots, J - 1$ .  $\mathcal{N}_{\text{move}}(\mathbf{x}, j)$  is defined as the set of solutions obtained by modifying a particular component of  $\mathbf{x}$  concerning activity  $j$ , i.e.,  $x_{jm_j}, y_{jm_j}$  and  $f_j$ . So we first consider three types of neighborhoods described below.

- $\mathcal{N}_{ft}(\mathbf{x}, j)$  : the neighborhood obtained by changing the finish time for an activity  $j$  so that it finishes between  $EFT_j(\mathbf{x})$  and  $LFT_j(\mathbf{x})$ .
- $\mathcal{N}_{mode}(\mathbf{x}, j)$  : the neighborhood obtained by changing the execute mode  $m_j$  for an activity  $j$ .
- $\mathcal{N}_{labor}(\mathbf{x}, j)$  : the neighborhood obtained by changing the number  $y_{jm_j}$  units of allocated crew formation vector for an activity  $j$ .

The duration of an activity may vary when a new mode or crew formation is selected. Therefore we further classify each of the neighborhoods  $\mathcal{N}_{mode}(\mathbf{x}, j)$  and  $\mathcal{N}_{labor}(\mathbf{x}, j)$  into two types; the set of solutions with the start time of activity  $j$  fixed, and that with finish time fixed. The former is denoted by  $\mathcal{N}_{mode}^s(\mathbf{x}, j)$  and  $\mathcal{N}_{labor}^s(\mathbf{x}, j)$ , and the latter by  $\mathcal{N}_{mode}^f(\mathbf{x}, j)$  and  $\mathcal{N}_{labor}^f(\mathbf{x}, j)$ , respectively. Moreover, the changes of mode or crew formation may cause the violation of precedence constraints for the activities  $\mathcal{P}_j$  or  $\mathcal{S}_j$ , since the duration of activity  $j$  may change. In this case, we assume that start times of all predecessors  $h \in Predecessor_j$  (successors  $h \in Successor_j$ ) of activity  $j$  are simultaneously slid by its excess date so that the precedence constraints are preserved. Thus, the neighborhood  $\mathcal{N}_{move}(\mathbf{x}, j)$  of MPTS-TS for RCPSPVDM is defined by

$$\mathcal{N}_{move}(\mathbf{x}, j) = \mathcal{N}_{ft}(\mathbf{x}, j) \cup \mathcal{N}_{mode}^s(\mathbf{x}, j) \cup \mathcal{N}_{mode}^f(\mathbf{x}, j) \cup \mathcal{N}_{labor}^s(\mathbf{x}, j) \cup \mathcal{N}_{labor}^f(\mathbf{x}, j). \quad (31)$$

In order to implement tabu search, we introduce the following four types of tabu lists.

- the operation tabu list (*OPE\_TL*): the element of this list is a pair of an activity number and an adopted operation.
- the start-time tabu list (*ST\_TL*): the element of this list is a pair of an activity number and an adopted start-time.
- the mode tabu list (*MODE\_TL*): the element of this list is a pair of an activity number and an adopted execute mode.
- the duration tabu list (*DUR\_TL*): the element of this list is a pair of an activity number and an adopted activity duration.

Then, subroutine *MOVE*( $\mathbf{x}, search\_type$ ) finds the best one from among  $\mathcal{N}_{move}$  that are not in tabu lists, where  $search\_type = NORMAL$  or *FREE* depending upon whether free penalty search is being executed or not. When  $search\_type = FREE$ , we use the objective function  $c(\mathbf{x})$ , i.e., the one obtained from (30) by disregarding the penalty term.

We define two additional subroutines to describe the the main tabu search.

- CLEAR\_TL*( $\cdot$ ):           Reset the elements of the four types of tabu lists, *OPE\_TL*, *ST\_TL*, *MODE\_TL* and *DUR\_TL* to null.
- IMPROVE*( $\mathbf{x}$ ):           Choose the best solution from the neighborhood  $\{\mathcal{N}_{move} \setminus (OPE\_TL \cup ST\_TL \cup MODE\_TL \cup DUR\_TL)\}$ , update a tabu list for the adapted operation, oscillate tabu lengths (i.e., increase the tabu size by one for changed activity, and decrease the tabu size by  $\bar{\beta}(= 1/(3 \times J))$  for others).

```

procedure M-TS( $\mathbf{x}$ )

1.  $\mathbf{x}_0 := \mathbf{x}$  /*  $\mathbf{x}$  represents a specified initial solution */
2.  $it := 0$  /*  $it$  represents the number of iterations */
3. CLEAR_TL()
4.  $free :=$  a positive integer
5. while stopping-criterion  $\neq$  yes do
6.    $\mathbf{x}_{it+1} :=$  IMPROVE( $\mathbf{x}_{it}$ , NORMAL)
7.   if NPV( $\mathbf{x}_{it+1}$ ) > NPV( $best\_schedule$ ) then
8.      $best\_schedule := \mathbf{x}_{it+1}$ 
9.      $it := 0$ 
10.   $it := it + 1$ 
11.  if  $it = free$  then /* free penalty search phase */
12.    while stopping-criterion  $\neq$  yes do
13.      IMPROVE( $\mathbf{x}_{it}$ , FREE)
14.    end
15.    CLEAR_TL()
16. end
17. return  $\mathbf{x}_{it}$ 

```

Fig. 7: The main tabu search (second stage) of MPTS-TS

### 3.4 Treatment of TDC resource

As noted in Section 2, the cost incurred for TDC resources is determined only after the the start and finish times of the usage of each of such resources. However, the start and finish times of the usage of TDC resource is not automatically determined even after the start and finish times of all activities are determined. Therefore, we need to determine them so as to minimize the cost of TDC resources. This is done as follows. Suppose that the start and finish times of all activities are already determined, i.e., the schedule of the project is already given. For each of TDC resource, say  $k$ , we first determine the minimum number of units of the TDC resource  $j$  that is sufficient to execute the current schedule. Let  $p$  denote such number and such units are assumed to be indexed by 1 through  $p$ . Notice that each such unit corresponds to virtual activity of TDC resource  $k$ . Let  $\mathcal{A}_k$  denote the set of activities that uses TDC resource  $k$ . For every activity that uses the resource we determine the assignment of every activity in  $\mathcal{A}_k$  to the resource unit index  $u$  with  $1 \leq u \leq p$  so that the net present value concerning the resource is minimized. In our algorithm, such minimization is done again by applying tabu search. In order to describe minimization procedure, let us define

$$s_j = \text{start time of activity } j, \quad f_j = \text{finish time of activity } j.$$

For convenience, let us assume that all  $s_j$  and  $f_j$  are distinct. Let  $s_{j^*} = \min_{j \in \mathcal{A}_k} s_j$  and assign  $j^*$  to the unit index 1. Let  $j^\circ \in \mathcal{A}_k$  be the activity such that (i) during the time interval  $[f_{j^\circ} - 1, f_{j^\circ}]$  all  $p$  units of TDC resource  $k$  are fully used, but (ii) after the time  $f_{j^\circ}$ , at most  $p - 1$  units are used. We then assign  $j^\circ$  to the unit index 1. We also assign the activities in  $\mathcal{A}_k$  whose start and finish times are in the interval  $[s_{j^*}, f_{j^\circ}]$  to the unit index 1 as many as possible as long as execution times of all assigned activities do not overlap each other. This is done in a greedy manner by giving the

activities with earlier start time to the higher priority to be chosen.

After the assignment of activities to the unit index 1 is completed, we delete the assigned activities from  $\mathcal{A}_k$  and the resource unit of index 1. We again apply the above procedure in the same fashion by trying to assign the remaining activities to the resource unit 2. We repeat this process until all the activities in  $\mathcal{A}_k$  are assigned one resource unit.

In this way, we determine the assignment of activities  $\mathcal{A}_k$  to the resource units for all TDC resources  $k$ . After the above assignment is completed, we can compute the net present value of the cost incurred for TDC resources. With this assignment as an initial solution, we then apply tabu search in order to improve the solution. We use in the tabu search the neighbourhood  $\mathcal{N}_{TDC}(\mathbf{x}, k, j, u; j', u')$  defined for each TDC resource  $k$ , where  $\mathcal{N}_{TDC}(\mathbf{x}, k, j, u; j', u')$  is defined as the set of solutions obtained by for every pair of activities  $j$  and  $j'$  that use TDC resource  $k$  exchanging the assignments of activities  $j$  and  $j'$  to the resource units  $u$  and  $u'$  in the current solution  $\mathbf{x}$ , respectively with those of  $j$  and  $j'$  to  $u'$  and  $u$ . Here we allow  $j'$  to be  $\emptyset$ , which means that  $\mathcal{N}_{TDC}(\mathbf{x}, k, j, u; \emptyset, u')$  contains the solution obtained by changing the current assignment of activity  $j$  to the resource unit  $u$  to  $u'$ .

In our implementation, the above tabu search is executed every time we need to evaluate NPV (e.g., when we seek for the best solution in the neighbourhood used in the initial or main tabu search).

### 3.5 Restart

When our heuristic procedure proceeds to the next pass from the main tabu search, the initial tabu search is again invoked to find a feasible solution. In order to ensure that a different and feasible schedule is generated, we follow the method introduced by Lee and Kim (1996), using two-dimensional array M\_INDEX.

M\_INDEX stores the objective value of a schedule at its element M\_INDEX[j][m<sub>j</sub>] when mode  $m_j$  is assigned to activity  $j$  in a given schedule. Whenever the objective value is improved, we update the corresponding elements of M\_INDEX to the new objective value. The value of each element M\_INDEX is initialized to infinity at the beginning of the algorithm.

We generate a feasible solution for the initial tabu search by using this M\_INDEX when the new pass is started. To put it concretely, we first choose a predetermined number of activities, and for each activity  $j$  chosen we choose the mode  $m_j$  with highest value of M\_INDEX[j][m<sub>j</sub>].

In the beginning, most of elements of M\_INDEX remain infinity so that most activities are assigned the mode which has not been selected so far. On the other hand, most modes will have been executed as the search proceeds. Then each value of the elements of M\_INDEX becomes equal to the best objective value obtained so far. Therefore the mode that has produced good objective value will be selected. In this way we can implicitly control the diversification and intensification depending on the progress of the search.

## 4 Computational Results

In order to implement MPTS-TS method explained in the previous section, there are a lot of parameters such as tabu lengths to tune the algorithm which affect the search behavior and the performance. Since it is impossible to investigate all possible combinations of parameter values, focusing our attention on the most important parameter, tabu length, we determined optimal parameter values by performing preliminary experiments. We omit the details here concerning the parameter optimization (see Goto (1999) for the details).

In this section, we shall first show the effectiveness of the proposed MPTS-TS method by applying it to a large number of problem instances generated by the project generator ProGen (Sprecher and Drexl (1998)). We shall then report the computational results obtained by applying the method to the real construction project and investigate the influence of various project conditions on the project cash flow project.

### 4.1 Preliminary experiments

In this subsection, we examine how powerful the proposed MPTS-TS method is by applying it to more than 10,000 problem instances belonging to the class RCPSPPM for which exact optimal solutions are known. All problem instances are generated by ProGen (Sprecher and Drexl (1998)). They are available in the project scheduling problem library PSPLIB from the University of Kiel. For detailed information see Sprecher and Drexl (1998).

The experiments have been performed on Sun Ultra 10; UltraSPARC-III 300MHz with 256MB main memory. Computational results are summarized in Tables 3 through 6 so that we can see the effect of the change of problem parameters (the numbers of activities, modes, renewable and nonrenewable resources) from the viewpoint of solution quality and computation time (details of computational results are reported in Goto (1999) ).

The standard set of problem parameters we used is that the numbers of activities, modes, renewable resources and nonrenewable resources are 16, 3, 2 and 2, respectively. In Tables 3 through 6, one of these parameters is varied while the others are fixed to those in the standard set. Legends used in Tables 3 through 6 are given in Table 2, where “class” in the first column denotes the problem class. For instance,  $j^*$  in Table 3 indicates that the numbers of activities is  $*$  ( $m^*$ ,  $r^*$  and  $n^*$  are similarly defined).

Table 2: Explanation of Legends

| Legend            | Definition   |
|-------------------|--|
| <i>class</i>      | problem class  |
| #                 | the number of problem instances solved   |
| <i>solved</i>     | the numbers of problem instances for which our method finds a feasible schedule  |
| <i>not_solved</i> | the number of problem instances for which a feasible solution has not been found |
| <i>best</i>       | the number of problem instances for which an optimal solution has been found     |
| $\Delta^{opt}$    | the average relative errors from an optimal solution                             |
| $\Delta^{max}$    | the maximum relative error from an optimal solution                              |
| <i>time</i>       | the average CPU time spent by our method   |

It is observed from Tables 3 through 6 that the proposed method computes solutions of high quality in reasonable amount of time.

Table 3: Computaional results for five different numbers of activities

| <i>class</i> | <i>#</i> | <i>solved</i> | <i>not_solved</i> | <i>best</i> | $\Delta^{opt}$ | $\Delta^{\max}$ | <i>time (sec.)</i> |
|--------------|----------|---------------|-------------------|-------------|----------------|-----------------|--------------------|
| j10          | 476      | 476           | 0                 | 428         | 0.0072         | 0.2917          | 15.6610            |
| j12          | 486      | 486           | 0                 | 425         | 0.0066         | 0.3125          | 32.9932            |
| j14          | 491      | 491           | 0                 | 359         | 0.0131         | 0.1579          | 27.9509            |
| j16          | 490      | 490           | 0                 | 334         | 0.0164         | 0.1667          | 35.0303            |
| j18          | 492      | 492           | 0                 | 333         | 0.0154         | 0.1379          | 43.4473            |
| j20          | 494      | 494           | 0                 | 303         | 0.0208         | 0.2700          | 53.5828            |

Table 4: Computaional results for five different numbers of modes

| <i>class</i> | <i>#</i> | <i>solved</i> | <i>not_solved</i> | <i>best</i> | $\Delta^{opt}$ | $\Delta^{\max}$ | <i>time (sec.)</i> |
|--------------|----------|---------------|-------------------|-------------|----------------|-----------------|--------------------|
| m1           | 580      | 580           | 0                 | 569         | 0.0007         | 0.0769          | 24.7867            |
| m2           | 431      | 431           | 0                 | 354         | 0.0065         | 0.1000          | 28.1034            |
| m3           | 490      | 490           | 0                 | 334         | 0.0164         | 0.1667          | 35.0303            |
| m4           | 489      | 488           | 1                 | 286         | 0.0255         | 0.1765          | 40.3142            |
| m5           | 493      | 493           | 0                 | 268         | 0.0322         | 0.2121          | 43.9819            |

Table 5: Computaional results for five different numbers of renewable resources

| <i>class</i> | <i>#</i> | <i>solved</i> | <i>not_solved</i> | <i>best</i> | $\Delta^{opt}$ | $\Delta^{\max}$ | <i>time (sec.)</i> |
|--------------|----------|---------------|-------------------|-------------|----------------|-----------------|--------------------|
| r1           | 494      | 494           | 0                 | 368         | 0.0139         | 0.1600          | 25.0624            |
| r2           | 490      | 490           | 0                 | 334         | 0.0164         | 0.1667          | 35.0303            |
| r3           | 497      | 496           | 0                 | 344         | 0.0165         | 0.2632          | 46.4113            |
| r4           | 493      | 493           | 0                 | 314         | 0.0195         | 0.1429          | 54.9711            |
| r5           | 477      | 477           | 0                 | 315         | 0.0195         | 0.1667          | 65.0092            |

Table 6: Computaional results for four different numbers of nonrenewable resources

| <i>class</i> | <i>#</i> | <i>solved</i> | <i>not_solved</i> | <i>best</i> | $\Delta^{opt}$ | $\Delta^{\max}$ | <i>time (sec.)</i> |
|--------------|----------|---------------|-------------------|-------------|----------------|-----------------|--------------------|
| n0           | 429      | 429           | 0                 | 365         | 0.0076         | 0.1143          | 26.6247            |
| n1           | 577      | 577           | 0                 | 420         | 0.0153         | 0.2857          | 32.9932            |
| n2           | 490      | 490           | 0                 | 334         | 0.0164         | 0.1667          | 35.0303            |
| n3           | 541      | 541           | 0                 | 359         | 0.0179         | 0.2353          | 39.1631            |



### 4.2 Construction project example

In order to examine the influence of various project conditions on NPV, we have solved an example project with 58 activities and 34 resource types, which is obtained from a real construction project (Koudacho-town, Aichi Prefecture, Japan). Basic information of the construction project is given below.

- Construction work : Laboratory and plant of a pharmaceutical company.
- Structure : RC construction (1F, 2F), steel structure (PH 1F),  
Individual footing, PC piles (footing).
- Total floor area : 2004 m<sup>2</sup> (1F·2F: 960 m<sup>2</sup>, PH: 64 m<sup>2</sup>).

Details of project activities and resources are given in Tables 7 through 11. Table 7 explains details of activities of the project. The third column denotes the material type to be allocated. When an activity does not require any material, the material type is given as “none”. The fifth column shows the type of equipment resource. It is assumed that at most one unit of equipment resource is required for each activity. The sixth column shows the type of labor resources required. We indicate the type of labor resource in the first item “type”, whose notation is explained Table 9. The remainder shows maximal and minimal units of crew formation and the basic productivity, respectively.

Tables 8, 9, 10, and 11 show the details of resources used in the project. Fig. 8 shows precedence relation among activities.

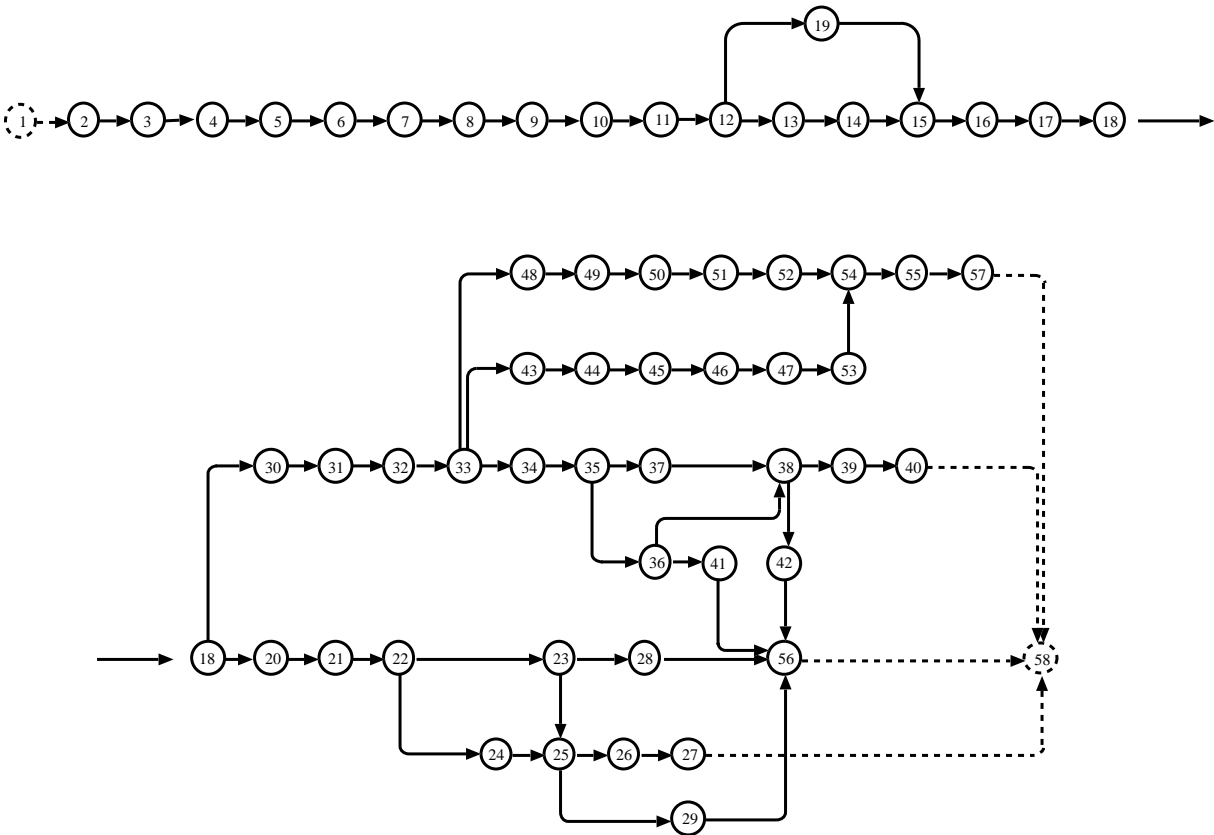


Fig. 8: Precedence relation of an example project

Table 7: Detailed description of activities of an example project

| act.#<br>(j) | activity description           | material<br>type | amount<br>( $W_j$ ) | mode<br>( $m_j$ ) | equipment        |            | type*2   | labor            |                  |
|--------------|--------------------------------|------------------|---------------------|-------------------|------------------|------------|----------|------------------|------------------|
|              |                                |                  |                     |                   | TDC *1           | non-TDC *1 |          | $y_{jm_j}^{max}$ | $y_{jm_j}^{min}$ |
| 1            | dummy job (source)             |                  |                     |                   |                  |            |          |                  |                  |
| 2            | preparatory work               | none             | 1000 ( $m^3$ )      | 1                 |                  |            | nav      |                  |                  |
| 3            | excavation                     | none             | 500 ( $m^3$ )       | 1                 | power shovel (S) |            | nav      | 8/1              | 50               |
|              |                                |                  |                     | 2                 | power shovel (L) |            | cpt      | 8/1              | 80               |
| 4            | broken stone foundation        | none             | 60 ( $m^3$ )        | 1                 |                  |            | nav      | 8/1              | 60               |
| 5            | concrete sub slab              | concrete         | 20 ( $m^3$ )        | 1                 | pumper           |            | con      | 8/1              | 10               |
| 6            | marking                        | none             | 3000 ( $m^2$ )      | 1                 |                  |            | mark     | 12/1             | 1000             |
| 7            | foundation molding             | mold             | 300 ( $m^2$ )       | 1                 |                  |            | cpt      | 12/1             | 20               |
| 8            | reinforcement                  | reinforcement    | 26 ( $t$ )          | 1                 |                  |            | bar      | 12/1             | 2                |
| 9            | concrete placing               | concrete         | 260 ( $m^3$ )       | 1                 | pumper           |            | con      | 8/1              | 60               |
| 10           | curing                         |                  | 5 days*3            | 1                 |                  |            |          |                  |                  |
| 11           | mold demolition                | none             | 300 ( $m^2$ )       | 1                 |                  |            | m-d      | 12/1             | 50               |
| 12           | back filling                   | none             | 180 ( $m^3$ )       | 1                 | power shovel (S) |            | nav      | 8/1              | 80               |
|              |                                |                  |                     | 2                 | power shovel (L) |            | nav      | 8/1              | 100              |
| 13           | mud room construction          | none             | 800 ( $m^2$ )       | 1                 | power shovel (S) |            | cpt      | 8/1              | 100              |
| 14           | 1F large panel readying        | large panel      | 550 ( $m^2$ )       | 1                 | power shovel (S) |            | cpt      | 12/2             | 600              |
| 15           | 1F molding                     | none             | 550 ( $m^2$ )       | 1                 |                  | crane(S)   | cpt      | 12/1             | 50               |
|              |                                |                  |                     | 2                 |                  | crane(L)   | cpt      | 12/1             | 80               |
| 16           | 1F reinforcing                 | none             | 30 ( $m^2$ )        | 1                 |                  |            | bar      | 12/1             | 2                |
| 17           | 1F readying                    | none             | 550 ( $m^2$ )       | 1                 |                  | crane(S)   | cpt, nav | 12/1             | 50               |
|              |                                |                  |                     | 2                 |                  | crane(L)   | cpt, nav | 12/1             | 80               |
| 18           | 1F concrete placing            | none             | 350 ( $m^3$ )       | 1                 |                  | crane(S)   | con, nav | 12/1             | 50               |
|              |                                |                  |                     | 2                 |                  | crane(L)   | con      | 12/1             | 80               |
| 19           | scaffolding                    | none             | 1340 ( $m^2$ )      | 1                 |                  |            | scf      | 8/2              | 100              |
| 20           | 1F concrete curing             |                  | 24 days*3           | 1                 |                  |            |          |                  |                  |
| 21           | 1F mold demolition             | none             | 550 ( $m^2$ )       | 1                 |                  | crane(S)   | m-d      | 12/2             | 80               |
|              |                                |                  |                     | 2                 |                  | crane(L)   | m-d      | 12/2             | 150              |
| 22           | 1F mud room construction       | concrete         | 150 ( $m^3$ )       | 1                 |                  |            | con, pls | 8/1              | 60               |
| 23           | 1F light gauge steel partition | LS               | 320 ( $m^2$ )       | 1                 |                  |            | ls       | 20/2             | 80               |
| 24           | 1F concrete block              | concrete block   | 60 ( $m^2$ )        | 1                 |                  |            | block    | 20/1             | 10               |
| 25           | 1F sash                        | sash, glass      | 10 ( $unit$ )       | 1                 |                  |            | sash     | 20/2             | 5                |
| 26           | 1F steel painting              | none             | 150 ( $m^2$ )       | 1                 |                  |            | paint    | 10/1             | 5                |
| 27           | 1F glazing                     | none             | 50 ( $m^2$ )        | 1                 |                  |            | glz      | 10/2             | 20               |
| 28           | 1F interior finishes           | none             | 20 ( $m^2$ )        | 1                 |                  |            | int      | 20/1             | 1                |
| 29           | 1F plastering                  | none             | 800 ( $m^2$ )       | 1                 |                  |            | pls      | 20/1             | 40               |
| 30           | 2F molding                     | none             | 550 ( $m^2$ )       | 1                 |                  | crane(S)   | cpt      | 12/1             | 50               |
|              |                                |                  |                     | 2                 |                  | crane(L)   | cpt      | 12/1             | 80               |
| 31           | 2F reinforcing                 | none             | 30 ( $m^2$ )        | 1                 |                  |            | bar      | 12/1             | 2                |
| 32           | 2F readying                    | none             | 550 ( $m^2$ )       | 1                 |                  | crane(S)   | cpt      | 12/1             | 50               |
|              |                                |                  |                     | 2                 |                  | crane(L)   | cpt      | 12/1             | 80               |
| 33           | 2F concrete placing            | concrete         | 330 ( $m^3$ )       | 1                 |                  | crane(S)   | con      | 12/1             | 50               |
|              |                                |                  |                     | 2                 |                  | crane(L)   | con      | 12/1             | 80               |
| 34           | 2F curing                      |                  | 24 days*3           | 1                 |                  |            |          |                  |                  |
| 35           | 2F mold demolition             | none             | 550 ( $m^2$ )       | 1                 |                  | crane(S)   | m-d      | 12/2             | 80               |
|              |                                |                  |                     | 2                 |                  | crane(L)   | m-d      | 12/2             | 150              |
| 36           | 2F light gauge steel partition | LS               | 250 ( $m^2$ )       | 1                 |                  |            | ls       | 20/2             | 80               |
| 37           | 2F concrete block              | concrete block   | 50 ( $m^2$ )        | 1                 |                  |            | block    | 20/1             | 10               |
| 38           | 2F sash                        | sash, glass      | 15 ( $unit$ )       | 1                 |                  |            | sash     | 20/2             | 5                |
| 39           | 2F steel painting              | none             | 130 ( $m^2$ )       | 1                 |                  |            | paint    | 10/1             | 5                |
| 40           | 2F glazing                     | none             | 40 ( $m^2$ )        | 1                 |                  |            | glz      | 10/2             | 20               |
| 41           | 2F interior finishes           | none             | 20 ( $m^2$ )        | 1                 |                  |            | int      | 20/1             | 1                |
| 42           | 2F plastering                  | none             | 600 ( $m^2$ )       | 1                 |                  |            | pls      | 20/1             | 40               |
| 43           | PH steel erection, dead bolt   | steel            | 10 ( $t$ )          | 1                 |                  | crane(S)   | scf, stl | 8/1              | 2                |
|              |                                |                  |                     | 2                 |                  | crane(L)   | scf, stl | 8/1              | 3                |
| 44           | PH ALC                         | none             | 130 ( $m^2$ )       | 1                 |                  |            | ALC      | 10/2             | 20               |
| 45           | PH sash                        | sash, glass      | 5 ( $unit$ )        | 1                 |                  |            | sash     | 20/1             | 5                |
| 46           | PH steel painting              | none             | 10 ( $m^2$ )        | 1                 |                  |            | paint    | 10/1             | 10               |
| 47           | PH glazing                     | none             | 4 ( $m^2$ )         | 1                 |                  |            | glz      | 10/1             | 4                |
| 48           | exterior mold demolition       | none             | 550 ( $m^2$ )       | 1                 |                  |            | cpt      | 12/2             | 50               |
| 49           | exterior concrete patching     | none             | 1000 ( $m^2$ )      | 1                 |                  |            | pls-p    | 20/1             | 20               |
| 50           | exterior spray painting        | none             | 1200 ( $m^2$ )      | 1                 |                  |            | paint    | 10/1             | 20               |
| 51           | parapet hardware               | none             | 130 ( $m^2$ )       | 1                 |                  |            | tns      | 20/1             | 10               |
| 52           | parapet painting               | none             | 50 ( $m^2$ )        | 1                 |                  |            | paint    | 10/1             | 20               |
| 53           | roof waterproofing             | none             | 1000 ( $m^2$ )      | 1                 |                  |            | wtp      | 20/2             | 10               |
| 54           | exterior scaffold demolition   | none             | 1340 ( $m^2$ )      | 1                 |                  |            | scf      | 8/1              | 100              |
| 55           | exterior steel stair           | none             | 10 ( $m^2$ )        | 1                 |                  |            | scf, stl | 8/1              | 1                |
| 56           | interior painting              | none             | 3500 ( $m^2$ )      | 1                 |                  |            | paint    | 10/1             | 35               |
| 57           | grading                        | none             | 35 ( $m^2$ )        | 1                 |                  |            | nav      | 8/1              | 7                |
| 58           | dummy job (sink)               |                  |                     |                   |                  |            |          |                  |                  |

\*1 only one unit is required by each activity.

\*2 the notation in this column is explained in Table 9.

\*3 this type of activity does not require any resource but the days indicated are required to cure the concrete.

Table 8: Details of material resources used in an example project

| resource description ( $k \in MR$ ) | cost (1,000 yen/unit) ( $c_k$ ) | $R_k$ (units/day) |
|-------------------------------------|---------------------------------|-------------------|
| concrete                            | 13                              | $\infty$          |
| large panel                         | 10                              | $\infty$          |
| mold                                | 4                               | $\infty$          |
| reinforcement                       | 50                              | $\infty$          |
| steel                               | 120                             | $\infty$          |
| light gauge steel (LS)              | 100                             | $\infty$          |
| glass                               | 10                              | $\infty$          |
| concrete block                      | 10                              | $\infty$          |
| sash                                | 10                              | $\infty$          |

Table 9: Details of labor resources used in an example project

| resource description ( $k \in LR$ ) | cost (1,000 yen/person) ( $c_k$ ) | $R_k$ (persons/day) |
|-------------------------------------|-----------------------------------|---------------------|
| navvy (nav)                         | 20                                | 8                   |
| carpenter (cpt)                     | 30                                | 12                  |
| concrete worker (con)               | 20                                | 8                   |
| marking worker (mark)               | 30                                | 12                  |
| mold demolition worker (m-d)        | 20                                | 12                  |
| reinforcing-bar placer (bar)        | 20                                | 8                   |
| scaffolding man (scf)               | 30                                | 20                  |
| plasterer (pls)                     | 30                                | 20                  |
| light gauge steel (ls)              | 20                                | 20                  |
| bricklayer (block)                  | 20                                | 20                  |
| sash worker (sash)                  | 20                                | 10                  |
| painter (paint)                     | 20                                | 10                  |
| glazier (glz)                       | 20                                | 20                  |
| interior worker (int)               | 20                                | 10                  |
| steel worker (stl)                  | 20                                | 10                  |
| ALC worker (ALC)                    | 20                                | 10                  |
| plaster patching worker (pls-P)     | 20                                | 20                  |
| tinsmith (tns)                      | 20                                | 20                  |
| waterproofing (wtp)                 | 20                                | 20                  |

Table 10: Details of non-TDC equipment resources used in an example project

| resource description ( $k \in \mathcal{ER} \setminus \mathcal{ER}^{TDC}$ ) | cost (1,000 yen/unit) ( $c_k$ ) | $R_k$ (units/day) |
|--|---------------------------------|-------------------|
| crane(S)   | 150                             | 1                 |
| crane(L)   | 200                             | 1                 |

Table 11: Details of TDC equipment resources used in an example project

| resource description ( $k \in \mathcal{ER}^{TDC}$ ) | cost (1,000 yen/unit) ( $c_k$ ) | $R_k$ (units/day) |
|---|---------------------------------|-------------------|
| power shovel (S)                                    | 100                             | 3                 |
| power shovel (L)                                    | 150                             | 3                 |
| pumper  | 100                             | 3                 |

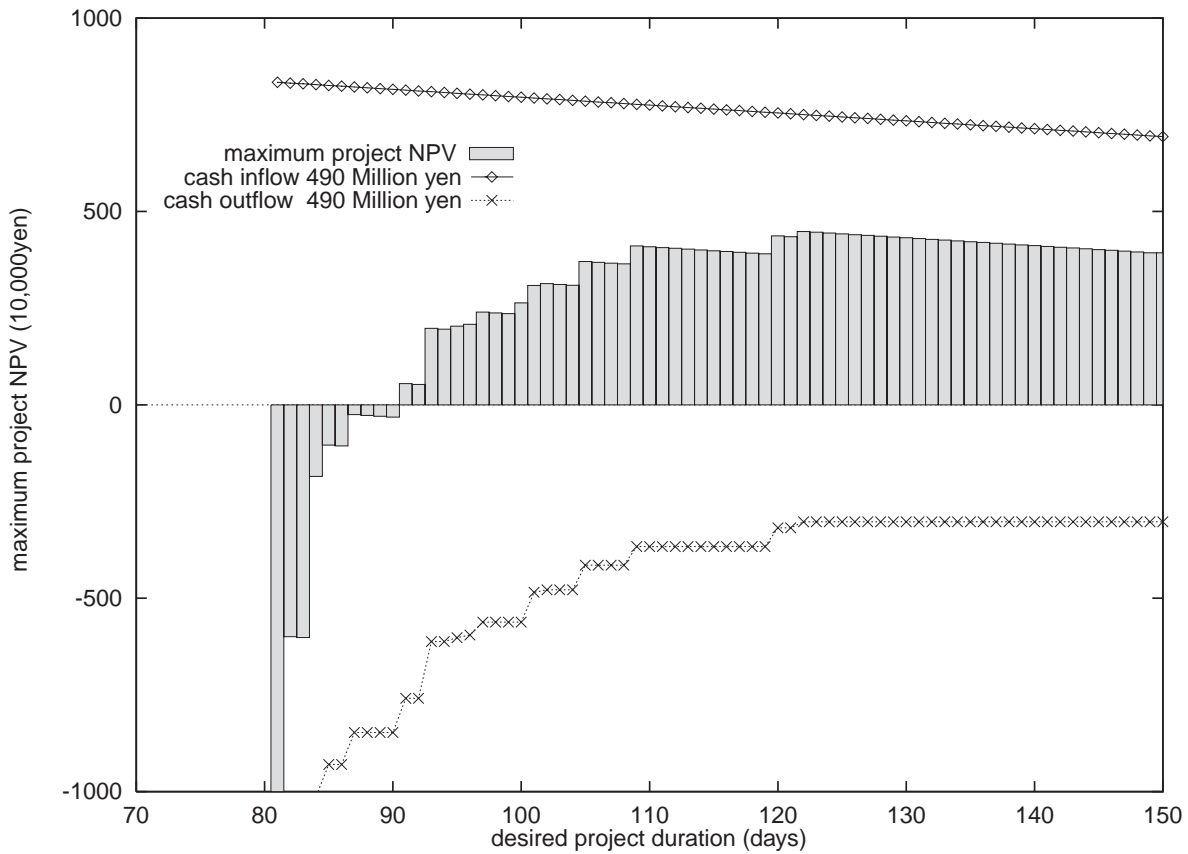


Fig. 9: Illustration of the change of NPV with respect to the increase of the desired project duration at interest rate 3.0%/year, cash ratio 10%, and bill duration 150 days

As explained in Section 1, we are going to maximize the project NPV from the standpoint of the project contractor. We assume that equal amount of cash inflow (client’s investment) occurs in three phases, i.e., at the beginning, in the middle, and upon completion of the project, which is regarded as an unchangeable factor in this project example. On the other hand, the conditions of project cash outflow are considered as changeable parameters. By varying them we observe how they affect the project NPV. This means that we try to maximize the project NPV under the given due date  $\bar{T}$  by changing the conditions of project cash outflow.

Fig.9 illustrates cash inflow, cash outflow, and the project total NPV under the condition that the interest rate is 3.0%/year, the cash ratio is 10%, and the bill duration is fixed to 180 days. MPTS-TS method produces a feasible schedule for this example project when the given project duration is at least 81 days. The project can be scheduled without difficulty when  $\bar{T}$  increases. Since payments are assumed to be made on a piece work basis every month, large decrease of cash outflow is observed when the number of months of the desired project duration increases by one (say, at  $\bar{T}=91$  and 121 in Fig. 9). On the other hand, the change of cash inflow is smooth and monotone decreasing with respect to the project duration as verified from (13) and (15). When  $\bar{T}$  is less than 90 days, NPV is negative, while when it exceeds 91 days NPV becomes positive. The best desired project duration is 122 days.

Fig.10 shows the influence of the change of bill duration and cash ratio on the project NPV at a fixed interest rate. As seen from the figure, the project NPV increases as bill duration increases, or cash ratio decreases. The same observation holds for the interest rate. As seen from the figure, the

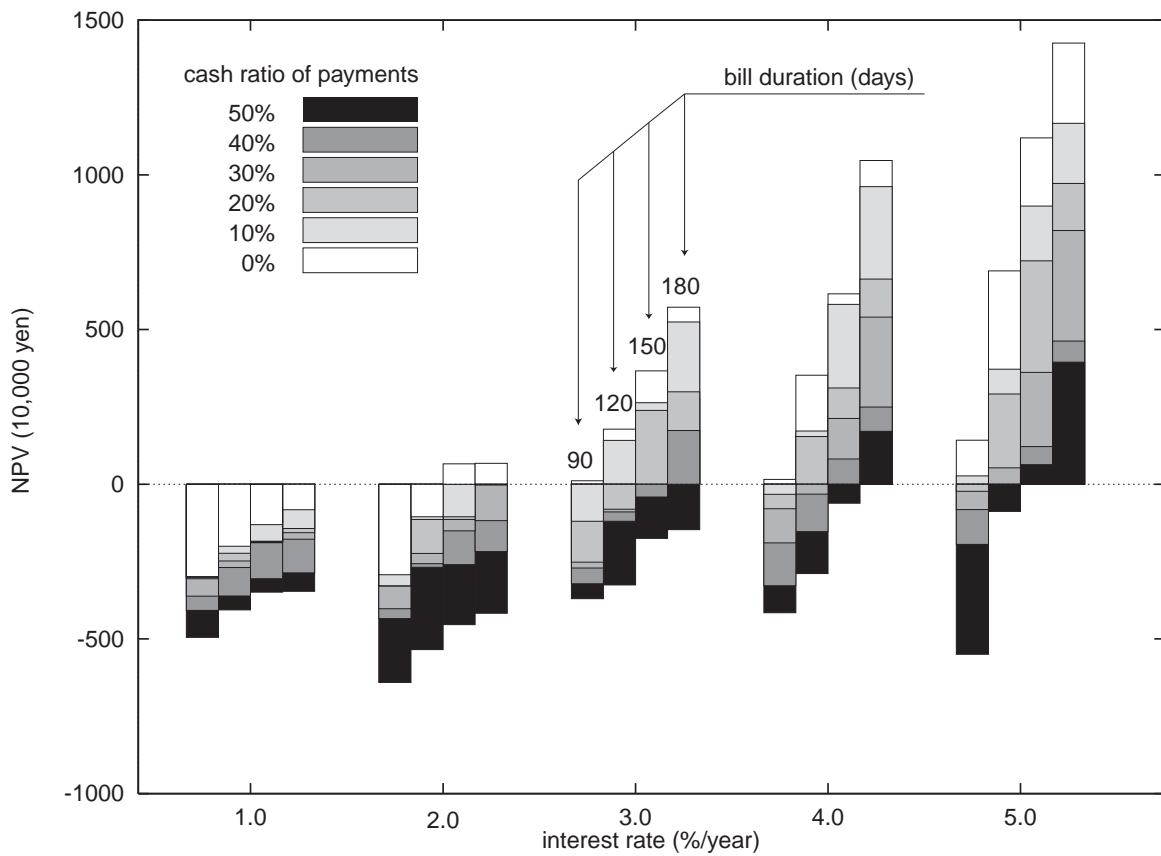


Fig. 10: The relationship between the interest rate and the project NPV with the several project conditions; the cash ratio of payments is fixed to the value between 0.0 and 50.0 %, the bill duration is fixed to the value between 90 and 180 days, and the project due date is set to 100 days.

project cannot be completed with positive profit anyway at interest rate 1.0%/year, and the project can be in the black at interest rate 2.0%/year under appropriate conditions. After then, when the interest rate further increases, the project becomes more likely to be in the black. The reason for such results is that the timing of client's investment is fixed at three phases as explained above, while project payment can be concentrated somehow to the latter half of the project duration. Therefore according to the increase of the interest rate, the rate of discounted cash outflow becomes larger than that of discounted cash inflow, which results in the increase of the total NPV. In addition, more particular attention should be paid to the rapid change of NPV with the increase of the interest rate. It is also seen from the figure that the difference between the highest and the lowest NPV at a fixed interest rate expands as the increase of the interest rate. In particular, at the interest rate 5.0%, such difference reaches approximately 20,000,000 yen, although it is only 5,772,000 yen at 1.0%. This fact indicates that the influence of condition of payments on NPV becomes more sensitive when the interest rate gets higher.

## 5 Conclusion

We first introduced a model of general project scheduling problem which we call RCPSPVDM. The concept of this model generalizes those which have been studied in the field of project scheduling so that we can deal with more complex project characteristics and conditions. Then we proposed a general metaheuristic algorithm for solving the problems which is applicable to most of existing project scheduling problems. The proposed approach, which is called MPTS-TS method in this paper, was based on the so-called tabu search method. Extensive numerical results revealed that our algorithm is effective for the problem class of RCPSPMM.

Finally, we applied the proposed method to the real construction project problem in order to observe the influence of the cash flow planning on NPV of the project from various aspects such as payment conditions and interest rate. In this paper, the project scheduling model RCPSPVDM was originally developed so that it can be applied to the construction industry. But we believe that the proposed model is applicable to many other fields. For future research, we are planning to extend the proposed model so as to simultaneously treat several projects for attaining an overall optimal scheduling of projects.

## References

- Taeho Ahn and S. Selcuk Erenguc, *A Heuristic Procedure: The Resource Constrained Project Scheduling Problem with Multiple Crashable Modes*. European Journal of Operational Research, Vol. 107, 250-259, 1998.
- Peter Brucker, Andreas Drexl, Rolf Möhring, Klauss Neumann and Erwin Pesch, *Resource-constrained project scheduling: Notation, classification, models, and methods*, European Journal of Operational Research, Vol. 112 (1999), 3-41.
- Daji Gong, *Optimization of Float Use in Risk Analysis-Based Network Scheduling*. International Journal of Project Management, Vol. 15, No. 3, 187-192, 1997.
- Eiji Goto, *Maximizing Net Present Value for Generalized Resource Constrained Project Scheduling Problem*. Master's thesis, Department of Architecture and Architectural Systems, Kyoto University, 1999.
- R.C. Grinold, *The Payment Scheduling Problem*. Naval Research Logistics Quarterly, Vol.19, No.1, 123-136, 1972.
- Willy S. Herroelen, Erik L. Demeulemeester, and Bert De Reyck, *A Classification Scheme for Project Scheduling*, in *Project Scheduling: Recent Models, Algorithms and Applications* (Jan Weglarz eds.), 1-26, 1999.
- Willy S. Herroelen, Patrick Van Dommelen and Erik L. Demeulemeester, *Project Network Models with Discounted Cash Flows A Guided Tour through Recent Developments*. European Journal of Operational Research, Vol. 100, 97-121, 1997.
- , O. Icmeli and S.S. Erengüç, A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows, *Management Science*, Vol.42 (1996), 1395-1408.
- Jae. Kwan Lee and Yeong. Dae Kim, *Search Heuristics for Resource Constrained Project Scheduling*. Journal of the Operational Research Society, Vol. 47, 678-689, 1996.
- K. Neumann and J. Zimmermann, Resource leveling for projects with schedule-dependent time windows, *European Journal of Operational Research*, Vol.117 No.3 (1999) 591-605.
- K. Neumann and J. Zimmermann, Methods for resource-constrained project scheduling with regular and nonregular objective functions and schedule-dependent time windows, in Jan Weglarz ed. *Project Scheduling: Recent Models, Algorithms and Applications*, 261-288, 1999.
- R. Padman, D.E. Smith-Daniels, and V.L. Smith-Daniels, Heuristic scheduling of resource-constrained projects with cash flows: An optimization approach, *Naval Research Logistics Quarterly*, Vol.23 (1997), 365-381.
- Rainer Kolisch and Rema Padman, *An Integrated Survey of Project Scheduling*, 1997.
- A.H. Russell, *Cash Flows in Networks*. *Management Science*, Vol.16, No.5, 357-372, 1970.
- Roman Slowinski and Jan Weglarz, *Advances in Project Scheduling*, Elsevier Science, 3-27. 1996.

Arno Sprecher and Andreas Drexl, *Multi-Mode Resource-Constrained Project Scheduling by A Simple General and Powerful Sequencing Algorithm*. European Journal of Operational Research, Vol. 107, 431-450, 1998.

Jan Weglarz, *Project Scheduling: Recent Models, Algorithms and Applications*. Kluwer Academic Publishers, 1999.

Kum Khiong Yang, F.Brian Talbot and James H. Patterson, *Scheduling a Project to Maximize its Net Present Value: An Integer Programming Approach*. European Journal of Operational Research, Vol. 64, 188-198, 1992.