Resource-Dependent Critical Path Method for Identifying the Critical Path and the "Real Floats" in Resource-Constrained Project Scheduling

Salman Ali NISAR[1], Koshi YAMAMOTO[2], Koji SUZUKI[3]

[1]Doctoral Student, Dept. of Civil Eng., Nagoya Institute of Technology
(Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan)
E-mail: ali_nisar02@yahoo.co.jp
[2]Professor Emeritus, Dept. of Civil Eng., Nagoya Institute of Technology
(Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan)
E-mail: yamamoto.koshi@nitech.ac.jp
[3]Associate Professor, Dept. of Civil Eng., Nagoya Institute of Technology
(Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan)
E-mail: suzuki.koji@nitech.ac.jp

In this paper, we develop a method called the resource-dependent critical path method (RDCPM) to identify resource dependencies that exist between activities in order to determine the critical paths and real floats in resource-constrained scheduling (RCS) techniques. The popular critical path method (CPM) and program evaluation and review technique (PERT) network techniques are based on the assumption of unlimited resources. This assumption is not valid in most practical applications, wherein exist definite limits on the amount of available resources. Although RCS techniques can consider resource limitations, they do not provide the correct floats and critical path, as do the CPM and PERT techniques. This is because in addition to technological relationships, a resource constraint schedule contains resource dependencies between activities that are neglected in RCS techniques.

RDCPM provides reasonable resource links between activities so that both total floats and free floats can be computed accurately, and critical activities and critical sequences can be correctly identified. Moreover, to minimize the number of resource links and reduce the complexity of the network, RDCPM establishes resource links between activities while considering their optimization and removes redundant relations. This approach makes a schedule more realistic and provides a stable schedule with progress updates. Therefore, it should be considerably more beneficial and useful for the construction industry.

Key Words: *Construction scheduling, Critical path method, Resource dependent, Forward and backward passes of resource-constrained schedule.*

1. Introduction

The critical path method (CPM) is widely used as a project management tool for improving scheduling and project administration tasks, supporting project managers by ensuring timely project completion, and reducing the budgetary strain[1)]. CPM entails a forward pass to determine total project duration (i.e., critical path) along with the earliest start time (ES) and earliest finish time (EF) for each activity. The forward pass is ensued by the backward pass for calculating the latest finish time (LF), the latest start time (LS), and the float for each activity[2)].

Critical path and float are the most important concepts in a project schedule. Critical path is the longest ordered sequence of activities through the project schedule; note that a schedule may contain more than one critical path. Each of the activities on such a path is said to be a "critical" or a "zero" float. When any of them is delayed, it causes a delay in the project completion date.

There are several types of floats, of which the simplest and most important types are total float (TF) and free float (FF). TF is the maximum amount of time for which the finish date of an activity can be delayed without affecting the completion of the entire project. TF is calculated as the difference between the LS and ES or between the LF and EF of an activity. FF is the amount of time for which the finish date of an activity can be delayed without affecting the start time of any other activities in a project. FF is calculated as the difference between the earliest ES among all the immediate successors of an activity and the EF of that activity. This information is very important for the project manager to plan and control the project more actively and efficiently[3),4)].

However, CPM is generally not realistic because it assumes that project activities have an access to unlimited resources. Moreover, some resources are highly limited in practice, and hence in most construction projects, scheduling without considering resource limitations may result in a non-credible schedule, and activities may be delayed by the unavailability of resources as well as by technological requirements.

To overcome the aforementioned problem, various mathematical, heuristic, and meta-heuristic resource constrained scheduling (RCS) techniques have been developed[1),4),5),6),7),8)]. Mathematical methods such as Linear Programming, Integer Programming, and Dynamic Programming techniques attempt to find the optimum solution in terms of minimum project duration. However, these techniques usually require very long computational times and are impractical for actual construction projects. Heuristic approaches, such as priority rules, provide reasonable solutions in a practical time, but do not guarantee optimality; hence, the user does not actually know how beneficial these approaches may be.

On the other hand, meta-heuristic or evolutionary algorithms such as genetic algorithms, simulated annealing, tabu search, and ant colonies are search techniques used in computing to find optimal or approximate solutions. Although meta-heuristic methods do not necessarily guarantee global optimal solutions, their ability to search the solution space intelligently rather than completely helps in obtaining relatively good solutions for large-sized problems.

The essential objective of the RCS technique is to minimize project duration and set resource availability to the maximum by delaying the activities' ES. Project completion time can be extended if an activity is delayed beyond its total float. Furthermore, RCS can successfully generate schedules that accomplish its own objectives. However, it does not provide the correct floats and the critical path of the schedule[9),10),11),12),13)]. While resource constraints are applied, the activity sequence relies not only on technological relationships, but also on resource dependencies. Because resource dependency is omitted in the CPM backward pass, the LS and LF of the activities that have resource dependencies can be greater than the real values. As a result of increasing the LS of the activities, incorrect floats are generated. Some critical activities will also find incorrect floats that cause hiding of the critical path[3),10),14)]. Furthermore, powerful project management software such as Primavera P6 and MS Project cannot provide the correct float[2),3)]. In this condition, the project manager cannot trust any float in the schedule and has to treat every activity as critical.

This paper proposes a new method, i.e., the resource-dependent critical path method (RDCPM), to identify resource dependent relations and to establish resource links between activities in a resource-constrained schedule. RDCPM aims to determine the correct floats and critical path by considering both technological and resource relations. The concept of RDCPM is illustrated along a simple schedule network taken from literature. To demonstrate its effectiveness, RDCPM is applied on a real project schedule.

2. Previous literatures

Several studies are proposed to solve the RCS limitations. Kim and Garza[3)], Bowers[9)], Lu and Li[11)], Wiest[12)], and Woodworth and Shanahan[13)], studied the methods of scheduling projects for identifying the resource dependency between activities for correct float calculation and critical activities in the RCS. These studies provide useful information for the present problem.

However, each method has significant shortcomings: the one proposed by Wiest et al. does not provide resource dependencies between activities, those by Woodworth et al. and Bowers are not capable of correctly identifying all resource links, and that by Lu et al. does not consider the original technological links of the CPM network when resource links are identified[14)].

To overcome these drawbacks, Kim and Garza[3)] proposed a 5-step resource-constrained critical path method (RCPM) algorithm. Step 1 performs traditional CPM calculations, and Step 2 performs serial RCS, in which if an activity is delayed due to resource constraints, one or more resource links are created from activities that have caused the activity delay and that have completed the release of the delay-caused resources. Step 3 performs a CPM backward pass considering both technological and resource links. In Step 4, every activity of the non-zero TF is checked by delaying the completion time on a daily basis, and resource link(s) are created if the TF is not available due to resource constraints. Finally, in order to make the schedule flexible, Kim and Garza proposed Step 5 to identify alternative schedules for certain activities. In this step, all resource links are temporarily removed, and every activity that has successor(s) by resource link is checked if the alternative schedule is available.

However, Kim's approach[3] also has certain limitations. First, as per Step 4, checking every activity of nonzero TFs can be a tedious task and is sometimes not applicable to real projects, which have a high number of activities. Furthermore, Kim and Garza do not explicitly provide information regarding how the resource link is created if the TF is not available. Second, it may not be required to check every activity that has successor(s) using a resource link in order to find an alternative schedule, whereas certain activities can provide an alternative schedule. Third, Kim's approach may generate a large number of redundant relations because it does not consider original technological links of the CPM network when resource links are identified. Those redundant activity relationships do not cause any time calculation errors, but the complexity of the scheduling network will be significantly increased.

For the above reasons, we propose RDCPM for resource constrained scheduling. A comparison of results shows that in most scenarios, by creating resource links and identifying the correct floats and critical path, RDCPM is superior to other methods. Important advantages of RDCPM compared to other methods are concluded as follows. (1) Different types of resource dependencies can exist between certain activities. RDCPM can identify these dependencies based on the activities' characteristics. Therefore, it is not necessary to check the entire schedule for resource dependencies. (2) Several alternative schedules may be provided by certain activities in an RCS. RDCPM can determine these specific activities so the project manager does not have to check every activity for an alternative schedule. (3) If resource dependency exists between more than one predecessor activity and more than one successor activity within a certain time, there will be several solutions for establishing resource links. Some solutions will significantly complicate the schedule network. Therefore, RDCPM provides a resource link optimization model to minimize the number of resource links and decrease network complexity. (4) Some predecessor relations between activities may become redundant after establishing resource links. RDCPM can identify and remove these redundant relations.

However, a limitation of RDCPM compared to other methods is that it does not consider multiple resource constraints. The other limitation of not only RDCPM but all methods that have been developed up to the present is that the maximum amount of available resources remains constant during execution of the project. However, the maximum amount of resources may vary in different stages of a real project. Therefore, further research is required to enhance RDCPM while considering the availability of a variety of multiple resources during the scheduling.

3. Basic assumption

All studies in this paper are based on the following assumptions:

- The duration of activity i (d_i) is a known deterministic integer.
- The current activity is non-preemptive or cannot be interrupted during scheduling.
- The project is constrained by one renewable resource r, which is constant during scheduling.
- There exist only finish–start technological precedence relations with zero time lags.
- The scheduling unit is on a daily basis, and the start time of the project is considered as day 0.

4. RDCPM description along a sample schedule

To create resource links between activities for the purpose of computing the correct TF and FF of each activity and identifying the critical path(s), RDCPM is composed of 4 main steps. For better demonstration of the algorithm, each step is described along a sample schedule taken from Ahuja et al.[1] and used by Lu and Li[11].

As shown in **Table 1** and **Fig.1**, the schedule network consists of 9 activities (SS and EE are start and finish activities of the project respectively, with zero duration). The maximum resource availability at any time is assumed to be 6 units. To represent the project schedule and the relation between the activities, we adopt the fenced bar chart[15] technique. The RDCPM processes are described as follows and are summarized on the flowchart given in **Fig.2**.

(1) Step 1: Performing RCS forward pass

After the application of the CPM, an RCS tech-

Table 1 Activity data of sample schedule

Activity	Duration	Resource	Precedence
SS	–	–	–
A	2	4	SS
B	3	4	SS
C	5	4	SS
D	4	3	A
E	4	1	A
F	3	2	B
G	6	2	B, C
H	2	2	D
I	3	2	F, G
EE	–	–	E, H, I

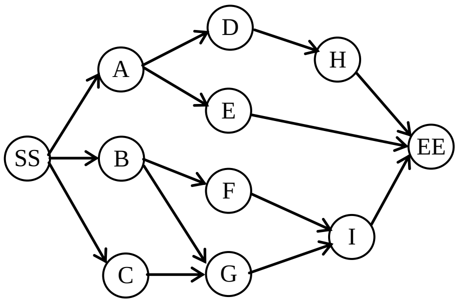


Fig.1 Sample schedule network

nique such as the heuristic method, an evolution algorithm, or a mathematical method is employed. The RCS forward pass attempts to find the ES and EF for each activity within the minimum project completion date such that all sequential and resource constraints are satisfied. In the sample schedule, for the sake of simplicity, the rank position weight method[16], is applied, which is a priority-rule-based heuristic method of RCS. The results of CPM and of the RCS forward pass are shown in **Fig.3** and **4**.

(2) Step 2: Performing RCS backward pass

The purpose of the RCS backward pass is to find

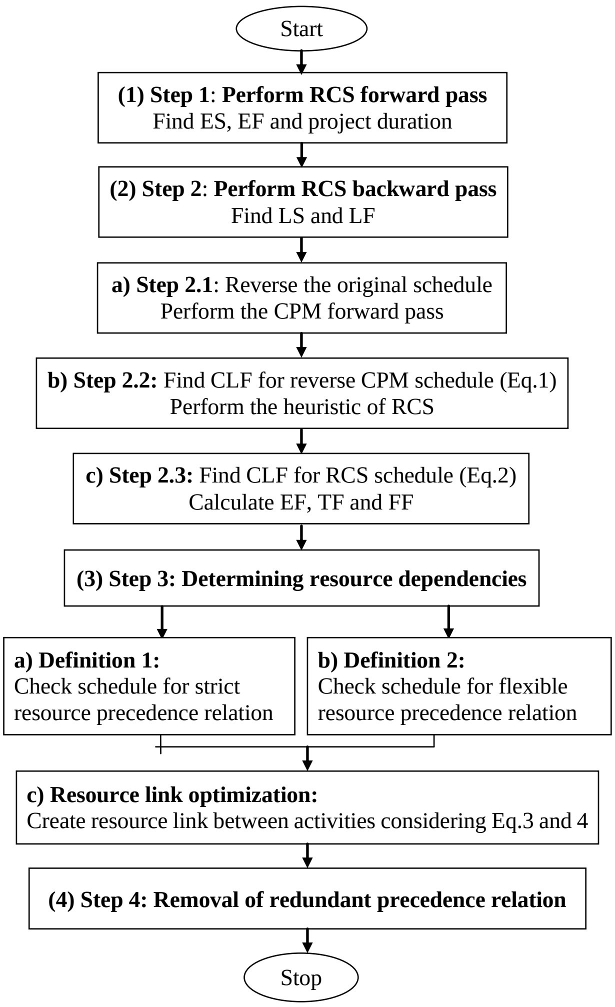


Fig.2 Flowchart for RDCPM process

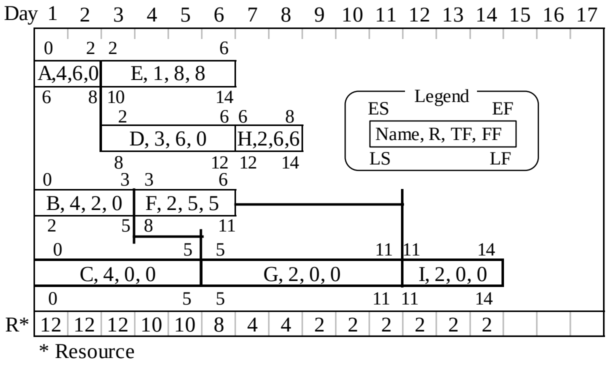


Fig.3 CPM schedule in fence bar chart for sample schedule

the LS and LF for each activity considering the resource constraints.

a) Step 2.1

The original schedule is reversed as all predecessor activities are changed to successor ones, and vice versa. Only the forward pass of the CPM is employed on the reversed schedule to find the ES and EF for each activity.

b) Step 2.2

Based on the ES and project duration of Step 1**,** the constraint's LF (CLF) is calculated from Equation 1 and imposed on each activity in the reverse CPM schedule. In this study, the CLF is defined as a time restriction imposed on each activity that limits the latest time by which it can be completed.

$$CLF_{Bi} = T - ES_{Ri} \quad i = 1, 2 \ldots n, \tag{1}$$

where CLF_{Bi} is the constrained latest finish time for activity i in the reverse CPM schedule, T is the minimum project duration in the RCS schedule in Step 1, ES_{Ri} is the earliest start time of activity i in the RCS schedule in Step 1, and n is the number of activities.

The LS and TF are found as per the CPM calculation while considering CLF. **Fig.5** shows the reverse CPM schedule with CLF. Afterwards, based on the heuristic method of the RCS technique, the maximum available resources are set, and resource overuse is eliminated by delaying the activities within

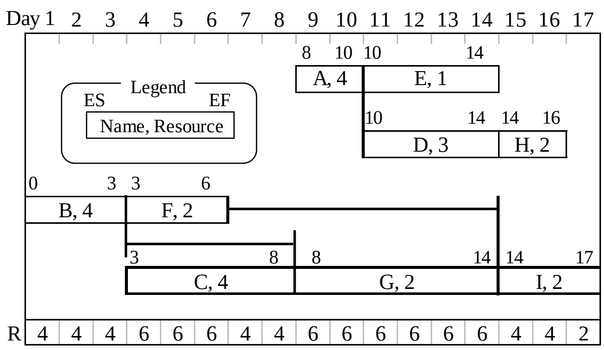


Fig.4 RCS schedule in Step 1

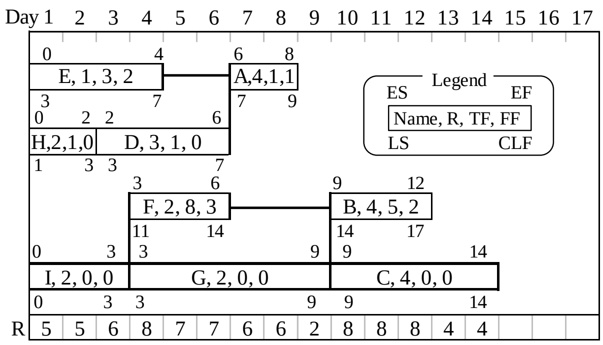


Fig.5 Reverse CPM schedule with CLF in Step 2.1 & 2.2

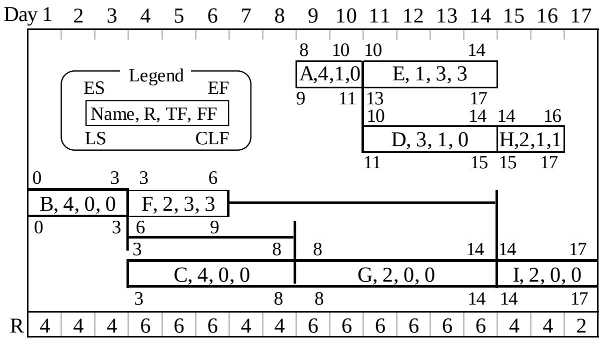


Fig.7 RCS schedule with CLF in Step 2.3

their TF. The activity is scheduled and resources are allocated by the CLF in ascending order. Note that the project duration is fixed and is equal to the result of Step 1. Therefore, an activity cannot be delayed beyond its TF; otherwise, the project duration will be extended. Because the CLF for all activities is calculated based on Step 1, there is a sufficient number of TFs for activities to set the maximum resource availability. The result of Step 2.2 is shown in **Fig.6**.

c) Step 2.3

Based on the results of Step 2.2, the CLF for the RCS schedule is calculated from Equation 2, and is imposed on each activity in the RCS schedule.

$$CLF_{Ri} = T - ES_{Bi} \quad i = 1, 2 \ldots n, \qquad (2)$$

where CLF_{Ri} is the constraint late finish time for activity i in the RCS schedule, and ES_{Bi} is the earliest start time of activity i in the reverse RCS schedule.

Afterwards, based on CPM calculation and considering the CLF, the LS, TF, and FF can be found for each activity. It should be noted that the FF will be calculated based only on the technological relation because the resource link is not yet created. Thus, some FFs may be calculated to be greater than the TF for a certain activity. If so, we assume FF = TF because FF cannot be greater than TF[17)]. As shown in the result of Step 2.3 in **Fig.7**, the critical path (the sequence activities denoted by bold line) and most of the correct TFs are identified. However, some TFs

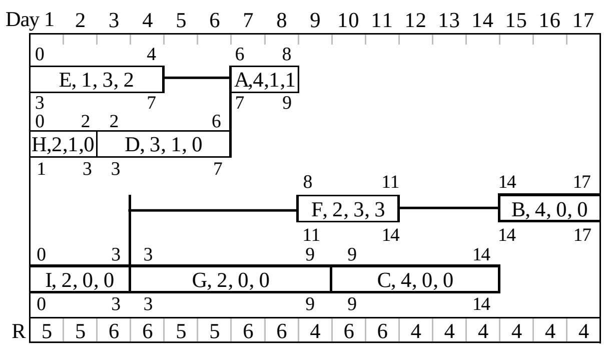


Fig.6 Reverse RCS schedule in Step 2.2

may still be incorrect, a situation described in the following steps.

(3) Step 3: Determining resource dependencies

There are two types of resource dependency relations: strict resource precedence relations and flexible resource precedence relations, depending on the effect of delaying a predecessor activity j on its current activity i if there is no technological relation between them. The activity currently being evaluated in the process at time t is referred to as the current activity.

a) Definition 1

Strict resource precedence relation (SRPR): if the finish time of predecessor activity j is delayed by a unit time, the start time of one or more current activity i is immediately delayed by the same unit time. Then, the relation between the predecessor activity j and the current activity i is called the SRPR, and the predecessor activity is called a strict resource predecessor activity. This relation is provided during the transfer of resources from the predecessor activity to the current activity by the processes of forward pass and backward pass in the RCS techniques. The strict resource precedence relation is determined as in the following cases.

Case 1: If $EF_j = ES_i$ and $TF_j = TF_i$, then the relation between predecessor activity j and current activity i is determined as SRPR. Hence, a resource link is created between them. The process begins from i=1, for which all activities are checked for the above condition. In the sample example shown in **Fig.7**, activities C and B satisfied the condition ($TF_C = TF_B = 0$, $ES_C = EF_B = 3$); thus, as shown in **Fig.8**, a resource link is established between them. Note that activities I and G also satisfy the conditions; however, as these activities have a technological relation, the resource link is not needed. Similarly, activities H, D, and A are not needed for the resource link.

Case 2: If case 1 is not satisfied for the current activity, then this activity should have at least one immediate predecessor activity with a technological relation finished at time t, i.e., the start time of the

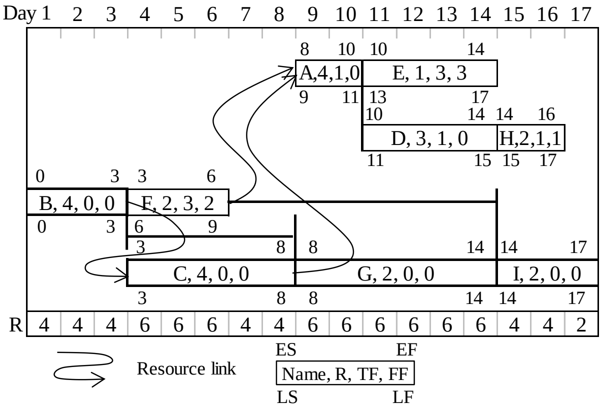


Fig.8 RDCPM schedule result in Step 3

current activity. Otherwise, one predecessor activity with a finish time of t ($EF_j = ES_i = t$) is nominated as its strict resource predecessor activity. In the sample schedule shown in **Fig.7**, activity A does not have any technological immediate predecessor activity at $t = 8$. Thus, activity C ($EF_C = ES_A = 8$) is nominated as its strict resource predecessor, and, as shown in **Fig.8**, a resource link is created between them.

If there exist more than one predecessor activities with $EF_j = t$, then the activity with the minimum TF is selected. In **Fig.9**, for instance, both activities A and C are finished at $t = 4$. Because activity C has the minimum TF, it is nominated as the predecessor activity of activity B.

b) Definition 2

Flexible resource precedence relation (FRPR): If the finish time of predecessor activity j is delayed by a unit time, it will delay the start time of one or more current activities by a` unit time, where $a > a$`. In other words, if the finish time of the predecessor activity is delayed by at least more than one unit time, it affects the start time of the current activity. Then, the relation between the predecessor activity and current activities is called the FRPR, and the predecessor activity is nominated as the flexible resource predecessor activity. In this situation, for the predecessor activity, FF_j and TF_j become positive. We describe Cases 3, 4 and 5 to identify the FRPR.

Case 3: In this case, the resource dependency relation is provided between a certain predecessor activity j_1 and current activity i, similar to SRPR, during the resources transfer in the forward and backward pass process of the RCS technique. However, due to the condition of SRPR, this relation could not be identified. In addition to the specific predecessor activity j_1, there is one or more predecessor activity j_2 that has either a resource relation or technological relation with current activity i, and its finish time is greater than the finish time of activity j_1. Therefore, the ES of current activity i is determined by precedence activity j_2 rather than by precedence activity j_1. For example, the ES of activity A in **Fig.7** is determined by activity C during the resource transfer, so they have SRPR. In addition, activity F cannot be executed concurrently with activity A on day 9 because of resource constraints. Thus, there is a flexible resource precedence relation between activities F and A, so that a resource link is created between them.

Case 4: Unlike the previously mentioned situations, the predecessor activity is scheduled by the processes of forward pass and backward pass in the RCS techniques without the transfer of resources. Then, the ES and LF are calculated based on the technological relation rather than by the resource transfer. Therefore, this kind of activity with a non-zero TF may not have its full float if there is any resource constraint for the TF period. When this situation happens on a schedule, there may be one or more alternative schedule. For instance, as shown in **Fig.10**, the ES and LF for activity C are found to be 0 and 7 by the RCS forward and backward passes, respectively, and the TF is calculated to be 5 days (TF = LF - EF). However, this TF cannot be used on days 4 and 5 because of resource constraints. Therefore, there exists a resource dependency between activity C and B so that a resource link is required between them. Now, the real TF for activity C becomes one day. An alternative schedule is available for activity C on days 6 and 7. In this schedule, the TF of activity C is zero (critical). It is obvious that the schedule with a greater TF is more flexible and will be superior. However, the alternative schedule(s) may be important in some special situations for a particular project.

To identify the flexible resource precedence rela-

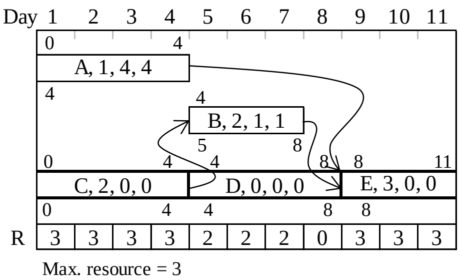


Fig.9 Creating resource link in Step 3

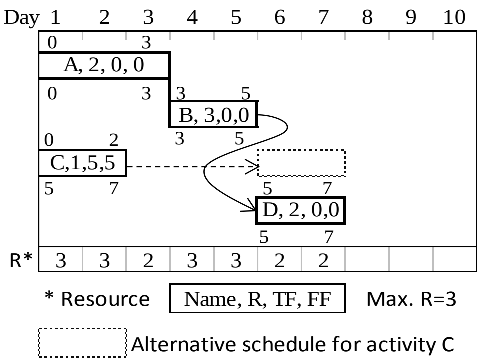


Fig.10 Identifying alternative schedule

tion and alternative schedule (if any), every activity with FF > 0 is checked by the extension of its duration within its full TF, and one or more resource link is created if there is resource dependency. For the sample schedule network in **Fig.7**, when the duration of activity F is extended within its full TF, it cannot be executed concurrently with activity A because of resource constraints. Hence, a resource link is created between them (**Fig.8**). Note that if the resource required for activities with FF > 0 is zero, its immediate predecessor activity(s) with TF > 0 is checked to identify the resource dependency.

Case 5: If the activity *j* can use its full TF without any resource constraints, then there should be at least one current activity *i* with technological relation with activity *j* at time *t*, where $t = LF_j = ES_i$. Otherwise, activity *j* is nominated as a flexible resource precedence activity for one current activity *i* with zero TF and $ES_i=LF_j$. For example, both activities A and B in **Fig.9** can use their full TF, because they do not have any successor activity with a technological relation at their LF ($LF_{A\&B}$ = 8), and there is only activity E with zero TF and ES_E = 8 ($ES_E = LF_{A\&B}$ = 8). Therefore, there exist the flexible resource precedence relations between among activities A, B and E so that resource links are created between them.

c) Resource link optimization

Four types of resource relation will be formed between current activities and their predecessor activities:

(1) one current activity has only one predecessor activity at *t* (One–One),

(2) many current activities have only one predecessor activity at *t* (Many−One),

(3) only one current activity has many predecessor at *t* (One–Many),

(4) many current activities have many predecessor activities (Many−Many).

It is easy to create a resource link between activities if there is a relation of type One−One, One−Many or Many−One. However, in the case of Many−Many relations, there are several solutions for creating resource links between activities. Therefore, a resource link(s) should be created between a certain predecessor activity and a certain current activity for the purpose of minimizing the number of resource links and to reduce the network complexity on the condition that the resource constraint can be satisfied. Therefore, the following equations are provided to find the optimal resource link(s) between activities.

$$\text{Min}\left\{NRL_{ji}^{t} = \textstyle\sum_{ji} \text{sgn}(r_{ji})\right\} \quad (3)$$

$$\text{such that} \quad \textstyle\sum_{j} r_{ji} \geq RR_i \ , \ i = 1, 2, \dots C^t \quad (4)$$

where NRL_{ji}^{t} is the number of resource links from predecessor activities *j* to current activities *i* at *t*; r_{ji} represents the idle resources of predecessor activities *j* which are transferred to current activity *i*; sgn(r_{ji}) is the sign function: 1 for $r_{ji} > 0$, 0 for $r_{ji} = 0$, and −1 for $r_{ji} < 0$; RR_i represents the required resources for current activity *i*; and C^t is the set of current activities at *t*.

The objective function of (3) is to minimize NRL_{ji}. If $r_{ji} > 0$, then one resource link is created between predecessor activity *j* and current activity *i*, and hence sgn(r_{ji}) = 1, whereas if $r_{ji} = 0$, then no resource link will be created between them, so sgn(r_{ji}) = 0. In **Fig.11**, for instance, based on condition of the Case 1, there are strict resource precedence relations between predecessor activities A, B, and C and their current activities D, E, G, and F, at time 3. Thus, some resource links have to be created between these activities. There may be many solutions to satisfy resource relations. However, in the case of **Fig.11**, where two solutions, i.e., (a) and (b), are provided and both satisfy the objective of resource link optimization, solution (a) will dominate solution (b).

As shown in **Fig.8**, RDCPM has so far reasonably identified the resource dependency, and has created the resource link so that a stable schedule is provided. However, there may be some redundant relation between activities. In other words, after creating resource links, some technological relations may become redundant. Therefore, an additional step is required to identify and remove the redundancy.

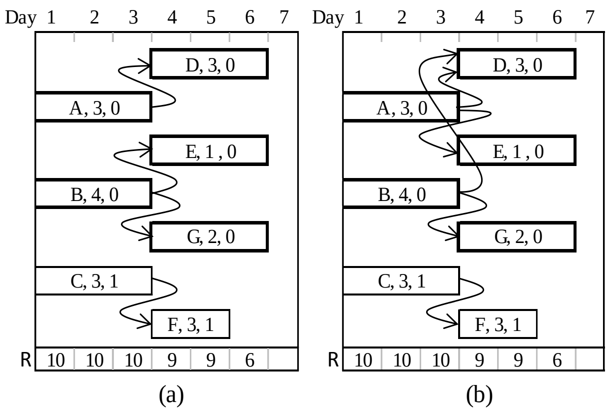


Fig.11 Resource link optimization for Many-Many type of resource relation

(4) Step 4: Removal of redundant precedence relation

The redundancy is defined as follows: if one activity among the immediate predecessor set of current activities is a predecessor for several other activities in the same set, then the relation between that activity and its current activity is redundant[18]. For example, in **Fig.12**, activities B and C are immediate predecessors of activity G; also, activity B preceded

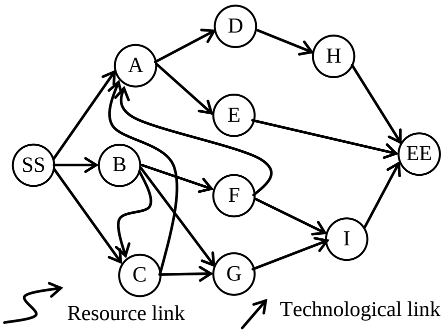


Fig.12 RDCPM network result in Step 3

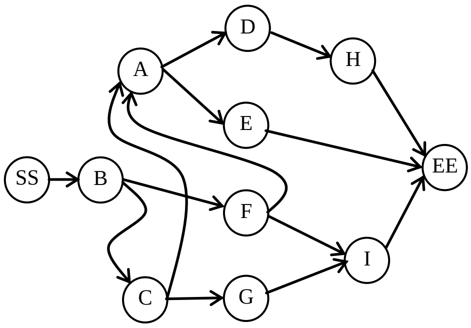


Fig.13 Final result of RDCPM network in Step 4

activity C (activity B is the resource predecessor activity of C). Thus, the relation between activities B and G is redundant and can safely be eliminated.

Although some methods have already been proposed for the identification and removal of redundant relations[18),19)], this study introduces a simple method to identify the redundancy.

Let us apply the method to the sample schedule network shown in **Fig.12**. As shown in **Table 2**, all activities are listed in column 1 and their immediate predecessor activities and more distant predecessor activities, i.e., predecessors of predecessors, are entered in columns 2 and 3, respectively. If a predecessor activity exists in the set of an immediate predecessor as well as in the set of a more distant predecessor for a certain activity, then the activity is redundant in the set of the immediate predecessor. The activities of the redundant immediate predecessor are underlined in **Table 2**.

As shown in **Fig.13**, RDCPM provided a stable schedule network in which the resource link can be considered to be like a technological relation. When the CPM is applied to the schedule network, the floats and the critical path(s) will be correctly computed, and the maximum resource availability will hence be satisfied. Furthermore, the schedule can be updated like the CPM schedule.

Table 2 Removing redundancy in sample schedule

Activity	Immediate pre.	More distant pre.
A	<u>SS</u>, C, F	SS, B
B	SS	
C	<u>SS</u>, B	SS
D	A	SS, B, C, F
E	A	SS, B, C, F
F	B	SS
G	<u>B</u>, C	SS, B
H	D	SS, A, B, C, F
I	F, G	SS, B, C,

5. Case Study

To evaluate the effectiveness of RDCPM, it is applied to a Warehouse project schedule used by Kim and Garza[3)] originally taken in the paper by Fondahl[10)]. The Warehouse project is broken down into 30 activities. The precedence relationship, duration, and three resource requirements, i.e., C (Carpenters), L (Laborers), and I (Iron workers), for each activity are shown in **Table 3**. The initial Warehouse schedule network is shown in **Fig.14**. The maximum

Table 3 Activity data of Warehouse project

Activity	Duation	Resources	Predecedence
1	1		
2	3		1
3	2	4C	2
4	2	2C	2
5	4	4C	2
6	1	4L	2
7	2	2C	2
8	3	4C	2
9	1	4C, 2L, 2I	6
10	1		6
11	1	C, 2L	9
12	1	C, L	11
12a	3		12
13	1	4I	12a
14	2	2L	10, 12a
15	1	4I	5, 13
16	1	2C, 2I	14
17	1	2C, 2I	15
18	1	C, 2L	13, 16
19	1	4I	4, 17
20	6		18
21	1	4I	19
22	2	4C	8, 20
23	2	2C	21
24	1	2C	7, 22
25	1	2L	22
26	1	2L	23
27	2		24, 25
28	2	2C	26
29	1	4C	3, 20
30	1	4L	27, 28, 29

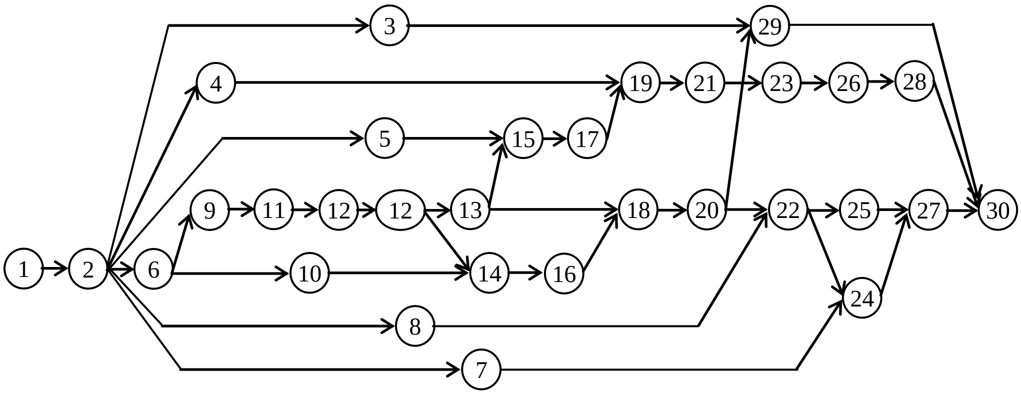
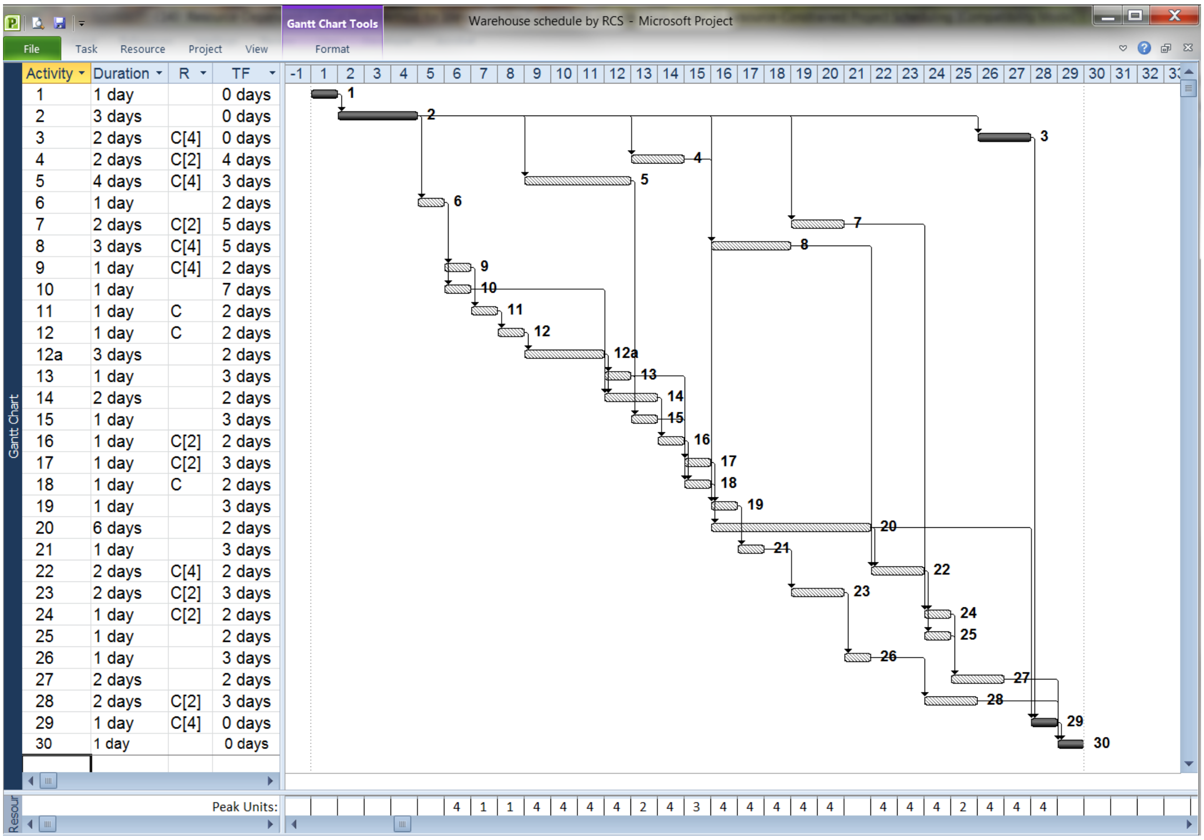


Fig.14 Warehouse scheduling network

availability of resource C is supposed as four units per day. However, for the sake of simplicity, it is assumed that the availability of the remainder of resources is unlimited. The project duration is set as 27 days by CPM considering unlimited resources.

The maximum available resource C was leveled in MS Project based on RCS. As shown in **Fig.15**, the project duration was extended to 29 days and the resource limitation is successfully leveled. However, MS Project generated and incorrect total float (total slack) for almost all activities, and could not provide the critical path. As shown in **Fig. 15**, only 5 activities were set as critical, i.e., activities 1, 2, 3, 29, and 30.

RDCPM was applied on the Warehouse project schedule step-by-step, as previously described. The MS Project result was adopted as the first step (performing RCS forward pass) of RDCPM. **Fig.16** shows the Warehouse schedule network including resource links which were provided by RDCPM. As shown in **Fig.16**, the redundant precedence relations were removed. This schedule network is now stable and can be updated safely like the CPM schedule. In addition, it is applicable to any kind of scheduling method, e.g., CPM, PERT, and scheduling project management software such as Primavera P6 and MS Project. For instance, if the precedence relations are modified in MS Project based on the RDCPM result, the real TF and critical path are generated successfully, as shown in **Fig.17**.



Activity	Duration	R	TF
1	1 day		0 days
2	3 days		0 days
3	2 days	C[4]	0 days
4	2 days	C[2]	4 days
5	4 days	C[4]	3 days
6	1 day		2 days
7	2 days	C[2]	5 days
8	3 days	C[4]	5 days
9	1 day	C[4]	2 days
10	1 day		7 days
11	1 day	C	2 days
12	1 day	C	2 days
12a	3 days		2 days
13	1 day		3 days
14	2 days		2 days
15	1 day		3 days
16	1 day	C[2]	2 days
17	1 day	C[2]	3 days
18	1 day	C	2 days
19	1 day		3 days
20	6 days		2 days
21	1 day		3 days
22	2 days	C[4]	2 days
23	2 days	C[2]	3 days
24	1 day	C[2]	2 days
25	1 day		2 days
26	1 day		3 days
27	2 days		2 days
28	2 days	C[2]	3 days
29	1 day	C[4]	0 days
30	1 day		0 days

Fig.15 Warehouse schedule by traditional RCS in MS Project

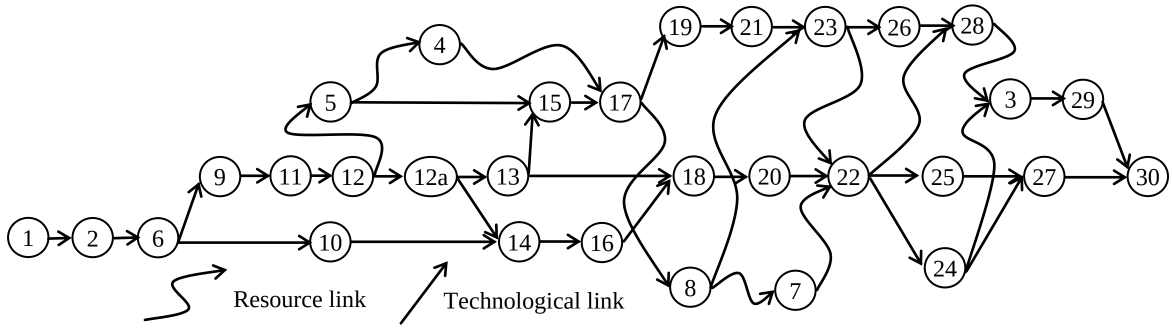
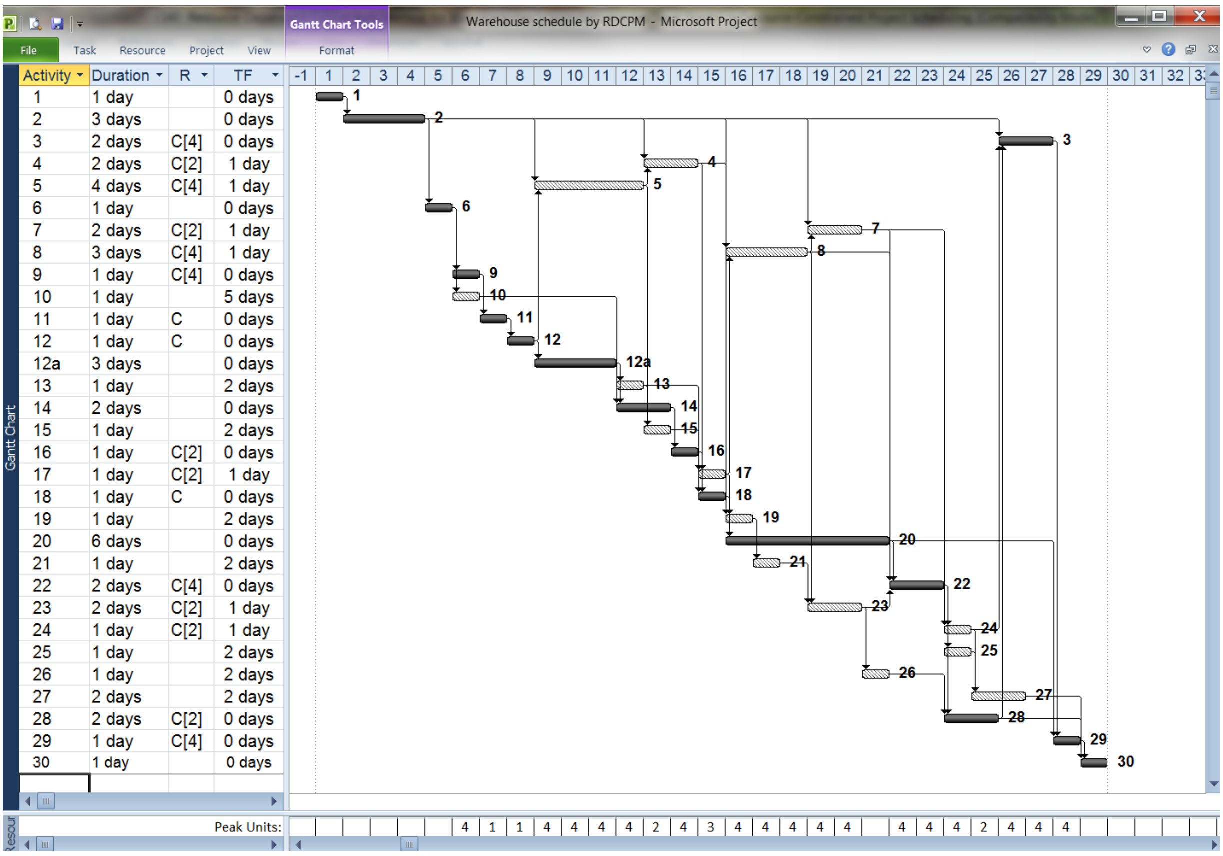


Fig.16 Warehouse schedule network after applying RDCPM

Fig.18 shows the warehouse schedule network provided by Kim and Garza[3]. There are several differences in the project duration and resource links between RDCPM and the results for Kim's method. This is because each method has applied different priority rules, and the maximum resource availability considered by each method is different. However, comparing the result generated by RDCPM with those generated by Kim's method, it observed that Kim's result appears to be much more complicated because of the many existing redundant links between activities. For instance, activities 2 and 9 in **Fig.18** are the immediate predecessors of activity 4. In addition, activity 2 is preceded by activity 9. Therefore, the link between activities 2 and 4 is redundant and is no longer required. Such redundant links between the activities in **Fig.18** are indicated by crosses.

6. Discussion and Conclusion

The critical path and float are the most important concepts in a project network schedule. The critical path is the sequence of critical activities that shows the longest path from the start to finish of a schedule network. On the other hand, a float is the amount of time by which the finish date of an activity can be delayed without affecting the completion of the project. The traditional RCS techniques are considered only for technological precedence relations between activities. However, resource dependencies



Activity	Duration	R	TF
1	1 day		0 days
2	3 days		0 days
3	2 days	C[4]	0 days
4	2 days	C[2]	1 day
5	4 days	C[4]	1 day
6	1 day		0 days
7	2 days	C[2]	1 day
8	3 days	C[4]	1 day
9	1 day	C[4]	0 days
10	1 day		5 days
11	1 day	C	0 days
12	1 day	C	0 days
12a	3 days		0 days
13	1 day		2 days
14	2 days		0 days
15	1 day		2 days
16	1 day	C[2]	0 days
17	1 day	C[2]	1 day
18	1 day	C	0 days
19	1 day		2 days
20	6 days		0 days
21	1 day		2 days
22	2 days	C[4]	0 days
23	2 days	C[2]	1 day
24	1 day	C[2]	1 day
25	1 day		2 days
26	1 day		2 days
27	2 days		2 days
28	2 days	C[2]	0 days
29	1 day	C[4]	0 days
30	1 day		0 days

Fig.17 Warehouse schedule by RDCPM in MS Project

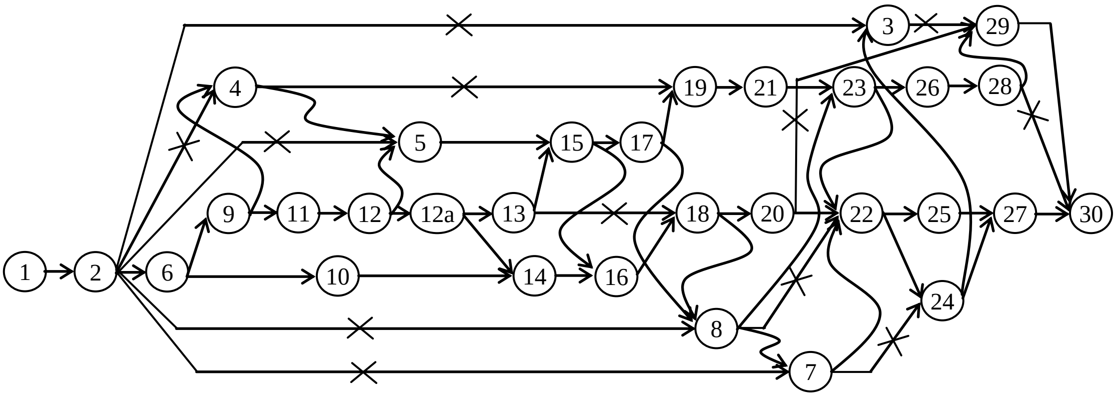


Fig.18 Warehouse schedule network provided by Kim and Garza, the redundant links are indicated by crosses

should also be considered in resource constrained projects. Otherwise, the floats cannot be computed correctly, and the critical path may not be identified accurately.

This study has proposed the concept of RDCPM, in which the resource dependency is identified so that the floats and critical path are found correctly. RDCPM entails a forward pass of RCS techniques such as the heuristic method, mathematical method, or evolution algorithm to determine the ES and EF for each activity, as well as minimum project duration. Based on the forward pass results, a backward pass has been applied to find the EF, LF, and floats. Using these two simple processes, the critical path is exactly identified and almost most of the TFs are computed. RDCPM determines two different resource precedence relations, i.e., strict and flexible precedence relations as mentioned in Step 3 in this paper. Moreover, the resource link optimization minimizes the number of resource links between current activities and their predecessor activities in order to decrease the network complexity of the schedule.

Finally, the redundant relation is identified by employing a method using **Table 2** and then removed from the schedule in order to reduce the network complexity.

REFERENCES

1) Ahuja, H. :*Project management techniques in planning and controlling construction projects*, Wiley, USA, 1984.
2) Lu M. and Lam H. : Critical path scheduling under resource calendar constraints, *J. Constr. Eng. Manage.*, Vol. 134, No. 1, pp. 25-31, 2008.
3) Kim, K. and De la Garza, J. M. : Phantom float, *J. Constr. Eng. Manage.*, Vol. 129, No. 5, pp. 507–517, 2003.
4) Wiest, J. D. : A heuristic model for scheduling large projects with limited resources, *Manage. Sci.*, Vol. 13, No. 6, pp. 359–377, 1967.
5) David, E. W. and Patterson, J. H. : A comparison of heuristic and optimum solutions in resource-constrained project scheduling. *Manage. Sci*., Vol. 21, pp. 944–955, 1975.
6) Hegazy, T. : Optimization of resource allocation and leveling using genetic algorithms, *J. Constr. Eng. Manage.*, Vol. 125, No. 3, pp. 167–175, 1999.
7) Pinson, E., Prins, C. and Rullier, F. : Using tabu search for solving the resource-constrained project scheduling problem, *In Proc. of the 4th Int. Workshop on Project Manage. and Scheduling, Leuven, Belgium,* pp. 102–106, 1994.
8) Boctor, F.F. 1996. Resource-constrained project scheduling by simulated annealing. *Int. J. of Production Res.*, Vol. 34, No. 8, pp. 2335–2351, 1996.
9) Bowers, J. A. : Criticality in resource-constrained networks, *J. Oper. Res. Soc*., Vol. 46, No 1, pp. 80–91, 1995.
10) Fondahl, J. W. : The development of the construction engineer: Past progress and future problems, *J. Constr. Eng. Manage.*, Vol. 117, No. 3, pp. 380–392, 1991.
11) Lu, M. and Li, H. : Resource-activity critical-path method for construction planning, *J. Constr. Eng. Manage.*, Vol. 129, No. 4, pp. 412–420, 2003.
12) Wiest, J. D. : Some properties of schedules for large projects with limited resources, *Oper. Res.*, Vol. 12, pp. 395–418, 1963.
13) Woodworth B M. and Shanahan S. : Identifying the critical sequence in a resource constrained project. *Int. J. of Proj. Manage.*, Vol. 6, No. 2, pp. 89–96, 1988
14) Kim, K. and De la Garza, J. M. : Evaluation of the resource constrained critical path method algorithms, *J. Constr. Eng. Manage.*, Vol. 131, No. 5, pp. 522–532, 2005.
15) Melin, J. W. and Whiteaker, B. : Fencing a bar chart, *J. Constr. Div., Am. Soc. Civ. Eng.*, Vol. 107, No. 3, pp. 497–507, 1981.
16) Tam, P. W. M. and Dissanayake, P. B. G. : Construction project scheduling by ranked positional weight method, *Can. J. Civ. Eng*., Vol. 25, pp. 424–436, 1998.
17) Mubarak, S. : *Construction project Scheduling and Control*. 2nd ed., John Wiley & Sons, Canada, 2008.
18) Wiest, D. J. and Levy, K. F. : A Management Guide to PERT/CPM with GERT/DCPM and other Networks, *Second Edition, USA*, 1977
19) Bashir. H. A. : Removal of redundant relationships in an AON project network for evaluating schedule complexity, *J. Constr. Eng. Manage.*, Vol. 136, pp. 787-793, 2010.

(Received May 20 , 2013)