A Resource-constrained CPM (RCPM) Scheduling and Control Technique with Multiple Calendars

by
Kyunghwan Kim

Dissertation submitted to the Faculty of
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
in
Civil Engineering

June 12, 2003
Blacksburg, Virginia

KEYWORDS: Scheduling, CPM, Resource-constrained Scheduling, RCPM, Phantom Float, Multiple Calendars, Project Management Software

A Resource-constrained CPM (RCPM) Scheduling and Control Technique with Multiple Calendars

by

Kyunghwan Kim

(ABSTRACT)

This research presents a Resource-constrained Critical Path Method (RCPM) technique that capitalizes on and improves the existing Critical Path Method (CPM) and Resource-Constrained Scheduling (RCS) techniques. A traditional CPM schedule is not realistic since it assumes unlimited resources, some of which are highly limited in practice. Although traditional RCS techniques can consider resource limitations, they do not provide correct floats and the critical path as the CPM does. The difference between the theoretical remaining total float and the real remaining total float is referred to as "Phantom Float" in this study. Another disadvantage of the traditional RCS techniques is that work sequence in the schedule could be considerably changed with a schedule update resulting in high costs to reorganize it. These problems are caused by the fact that, in addition to technological relationships, a resource-constrained schedule contains resource dependencies between activities that are neglected in traditional RCS techniques.

This study proposes a step-by-step RCPM algorithm to consider those resource-constrained relationships. Hence, the method can identify real floats and correct critical paths, considering both technological and resource-dependent relationships. RCPM also provides a certain level of stability with a schedule update due to the newly identified resource relationships. Based on the RCPM algorithm, a prototype RCPM system has been developed using Visual C++, Visual Basic, and Ra (Primavera Project Planner API). The system is integrated with P3, so that it reads project information directly from a P3 project, performs necessary RCPM procedures, and updates the P3 project to contain identified resource relationships. To make the system more practical, functions to handle multiple project calendars and progressed schedules have been included as well.

This dissertation is dedicated to the memory of
my grandparents and grandparents-in-law.

Acknowledgements

I would first like to thank my advisor, Dr. Jesús M. de la Garza, for his guidance and patience over the last several years. His knowledge, support, and advice were invaluable to this dissertation. I am also indebted to him for providing financial support through the Virginia Tech Construction Affiliates' Center for Construction Improvement and many opportunities that he has brought to me.

I would also like to thank the other members of my committee, Dr. Vorster, Dr. Martinez, Dr. Songer, and Dr. Ellis for their precious feedbacks, comments, and suggestions. Special thanks go to my previous advisor, Dr. Jae-Jun Kim at Hanyang University, for his continuous encouragement.

I would like to express my sincere gratitude to all friends in Patton 321 and all Korean colleagues in the Department of Civil and Environment Engineering for their priceless, warmhearted friendship. Many friends in other departments and in Korea must also be included for this acknowledgement.

I could not thank enough my parents, Youngho Kim and Jungseun Youn, and parents-in-law, Youndon Choi and Yupja Youn, for their unconditional love, support, and encouragement.

Finally, my deepest gratitude goes to my beloved wife, Moonjung, for her devotion and sacrifice on this long journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Project completion within a time limit is essential for successful project performance, regardless of the size and complexity of the project. Each day of delay in the completion time constitutes a loss in revenue that can hardly be recovered later. Developed in the 1950s, the Critical Path Method (CPM) has been widely used as a project management tool to improve scheduling and project administration tasks, supporting project managers to ensure project completion on time and on budget.

The CPM generates useful information about the project, such as the shortest project duration, the critical path(s), and the Total Float (TF) and the Free Float (FF) of each activity. This information is substantially important for the project manager to plan and control the project more actively and efficiently. Especially, critical and near-critical activities should receive the management attention that might be unnecessary on other activities. This management by exception is an important advantage of the CPM, especially on large, complex projects [Ahuja 1976; Bowers 1995].

In addition to planning and control tools, the CPM is useful to analyze project delay impacts. By simulating the CPM network with delays, the impact of the delays on the project completion date can be shown, and the amount of each participant's responsibility for the delays can be identified [Callahan et al. 1992; Bartholomew 1998].

The basic assumption of the CPM, however, is that resources required by activities are unlimited, while some resources are highly limited in practice. In most real construction projects, scheduling without considering resource limitations may result in a non-credible schedule, since the startability of activities is affected by resource availability. Various analytical and heuristic Resource-Constrained Scheduling (RCS) techniques have been developed to apply resource availability into the scheduling process [Ahuja 1976; David 1975; Hegazy 1999; Wiest 1967; Kelley 1963]. Analytical methods attempt to find the optimum solution in terms of the minimum project duration, but usually require a very long computational time, making them impractical for real construction projects. On the other hand, heuristic approaches provide reasonable solutions in practical time [Boctor 1990; Hegazy 1999]. For this reason, most project management systems employ heuristic methods.

1.2 Shortages in RCS

Once resources are considered in the scheduling process, however, correct float information is not available, and, as a result, one or more odd paths become critical [Bowers 1995; Fondahl 1991; Just and Murphy 1994; Wiest 1964; Woodworth and Shanahan 1988]. The difference between the "theoretical remaining" TF and the "actual remaining" TF discussed by Fondahl during the Fifth Annual Peurifoy Construction Research Award address [Fondahl 1991] is referred to as **Phantom Float** (PF) in this study.

Correct floats and the correct critical path may be identified by juggling with all activities moving back one by one until the project completion date is affected. However, this method, even with modern project management software, is not applicable to all activities in a real project that may have several hundreds of activities. Project managers therefore may have to assume that all activities are on the critical path since they cannot trust float data in the resource-constrained schedule and the correct calculation of them are practically impossible. This is a significant burden for project managers because they have to pay attention to all activities without taking the advantages of management by exception of the CPM. Fondahl (1991) pointed out that incorrect floats and broken critical paths in RCS have been widely ignored by construction engineers.

In addition, work sequence in a resource-constrained schedule may not be stable with a schedule update as the project proceeds, while the sequence is consistent in a CPM schedule [Willis 1985]. Since the schedule needs to be updated periodically based on current project performance, it is required to rebuild a resource-constrained schedule when a certain amount of the schedule has changed (e.g. activity duration extension, activity start/finish time change, etc.). This rescheduling normally generates a different schedule in terms of activity work sequences. For instance, an activity scheduled to start one month later from the current date could be scheduled to start the day after the current date. The changed activity sequence occurs since current practical RCS techniques employ certain priority rules (LST, TF, FF, etc.), and the updated activity data could affect those activity priority orders. The changed activity sequence could cause the project manager to reorganize the project at high cost and this reorganization may be required every time the schedule is updated.

The other drawback of a resource-constrained schedule is its inability to provide flexibility in activity start and finish times. Traditional RCS techniques normally generate one fixed schedule. However, depending on the resource usage and the network conditions, there could be alternative start and finish times for certain activities in the same schedule without delaying the project completion time [Bowers 2000]. If these alternative schedules are known, the schedule will be more flexible to handle unanticipated events, such as equipment failure, delays in material delivery, etc.

1.3 Current Scheduling State

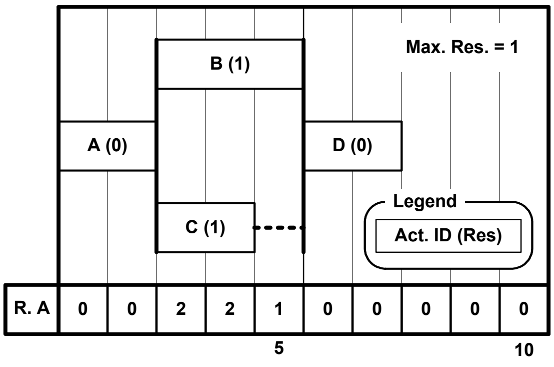
Regardless of the aforementioned disadvantages, construction professionals are heavy users of the CPM and RCS techniques assisted by project management software. Primavera Project Planner™ (P3) is one of the most widely-used systems in project planning and control, while Microsoft Project™ is the second most popular in the market [Liberatore et. al. 2001]. Those software tools have assisted project managers in the planning and administrating of complex projects, performing large, repetitive calculation [Feibus 1993; Hernandez 1999], while the shortages in the CPM and RCS techniques are still existing.

Many CPM construction schedules, in addition to genuine technological activity relationships and preferential activity sequences, often include resource links based on the manual observation of important resources' usage [Bowers 1995; Clough et. al 2000]. The resource links can be eliminated if sufficient resources become available, but the technological and preferential links may not be removed even with unlimited resources due to means and methods of construction that are independent of resource supplies. It is not efficient or effective to find and remove resource links manually due to the complexity of the real scheduling network and the multiplicity of resource requirements and availabilities. This method may require considerable work to handle those resource dependencies, probably covering only a small number of resources. In addition, it does not guarantee an efficient use of those considered resources compared to an RCS technique. Furthermore, significant rework is necessary to change those relationships as the schedule is updated. After a schedule update, the initial resource dependency condition can be changed. As a result, some relationships may not be necessary and some additional relationships may be required depending on the changed resource condition.

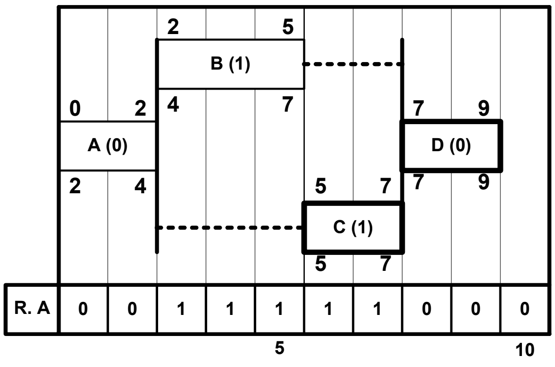
1.4 Problem Diagnosis and Approach to Solution

Phantom floats are generated by the CPM-like backward pass based on only technological relationships after RCS [Bowers 1995; Wiest 1967; Woodworth and Shanahan 1988]. Once resource constraints are applied, the sequence of activities relies not only on technological relationships but also on resource dependencies. Since the resource dependency is omitted in the CPM-like backward pass in RCS, the late time of an activity that has resource dependency becomes greater than the real values. If the late time of an activity is greater than the real value, the total float will be increased by the difference. Furthermore, the increased total float or the late time could affect predecessors of the activity based on the scheduling network condition, and then, most critical activities would become non-critical. The same applies in P3 and MS Project™; the critical path is broken (not continuous) and most TF values have phantom floats in a resource-constrained schedule.

A scheduling example as shown in Figure 1.1 illustrates how phantom floats are generated in a traditional RCS technique. With a resource constraint of one per day, the CPM schedule in Figure 1.1 (a) is not realistic since both *Activity B* and *C* require one *Resource A* for day 3 and 4. In order to eliminate the resource-overused period, RCS is applied and Figure 1.1 (b) schedule is generated as a result. Corresponding P3 schedules of these schedules are shown in Figure 1.2.

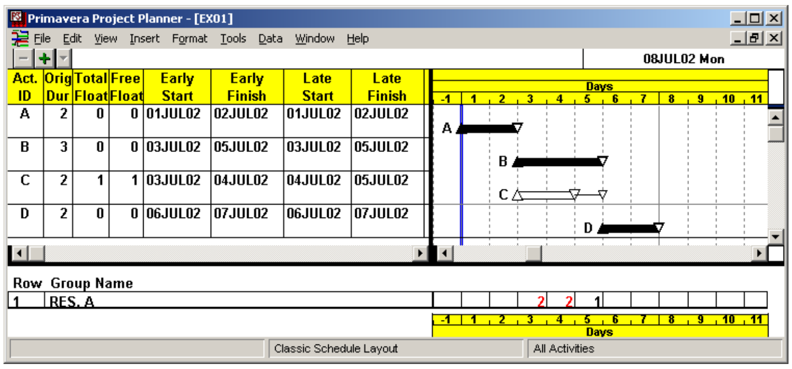


(a) CPM Schedule



(b) RCS Schedule

Figure 1.1 Phantom Floats in RCS (Bar Chart)

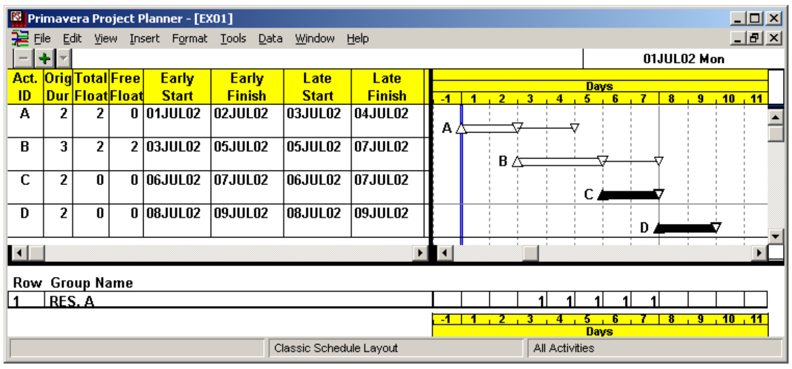


Primavera Project Planner - [EX01]
File Edit View Insert Format Tools Data Window Help
08JUL02 Mon
Days
-1 1 2 3 4 5 6 7 8 9 10 11
A
B
C
D
Classic Schedule Layout
All Activities

Act. ID	Orig Dur	Total Float	Free Float	Early Start	Early Finish	Late Start	Late Finish
A	2	0	0	01JUL02	02JUL02	01JUL02	02JUL02
B	3	0	0	03JUL02	05JUL02	03JUL02	05JUL02
C	2	1	1	03JUL02	04JUL02	04JUL02	05JUL02
D	2	0	0	06JUL02	07JUL02	06JUL02	07JUL02

Row	Group Name												
1	RES. A				2	2	1						

(a) CPM Schedule



Primavera Project Planner - [EX01]
File Edit View Insert Format Tools Data Window Help
01JUL02 Mon
Days
-1 1 2 3 4 5 6 7 8 9 10 11
A
B
C
D
Classic Schedule Layout
All Activities

Act. ID	Orig Dur	Total Float	Free Float	Early Start	Early Finish	Late Start	Late Finish
A	2	2	0	01JUL02	02JUL02	03JUL02	04JUL02
B	3	2	2	03JUL02	05JUL02	05JUL02	07JUL02
C	2	0	0	06JUL02	07JUL02	06JUL02	07JUL02
D	2	0	0	08JUL02	09JUL02	08JUL02	09JUL02

Row	Group Name												
1	RES. A				1	1	1	1	1				

(b) RCS Schedule

Figure 1.2 Phantom Floats in RCS (P3)

In this RCS schedule, the critical path is broken as only *Activities C* and *D* are critical. Based on the technological relationships in the CPM-like backward pass, the LFT of *Activity B* came from the LST of *Activity D*, creating 2-day total float for *Activity B*, and then creating 2-day total float for *Activity A*. However, if the finish time of *Activity B* or *A* is delayed by 1 day, the start time of *Activity C* and then the overall schedule completion time should also be delayed by 1 day to hold the maximum daily resource availability. Hence, the 2-day total floats of *Activity B* and *A* are phantom floats since they do not really exist.

Considering these circumstances, the following question is contemplated: Is there a systematic way to identify the resource dependency and to keep track of activity late finish time considering both technological relationships and resource dependencies to resolve the given problem?

1.5 Study Objective

The objective of this study is to develop a Resource-constrained CPM (RCPM) algorithm that takes advantage of both the CPM and the RCS techniques. The RCPM algorithm 1) considers resource availability that makes a schedule more realistic, 2) correctly identifies floats and the critical path, 3) recognizes alternative activity schedules in the given schedule, and 4) provides a stable schedule with progress updates.

A computer system is developed to implement the RCPM algorithm and facilitate its effective use. In order to save time and effort for users to exchange project data between systems, the system is integrated with P3, which is the most popular project management software in the construction industry [Liberatore et. al. 2001]. In order to make the system practical for real construction projects, functions to handle multiple project calendars and progressed schedules have also been included.

Chapter 2

Literature Review

2.1 CPM

Generally two types of floats, total and free floats, are considered in the CPM[1]. The total float of an activity is the difference between its earliest and latest start times (EST and LST) or between its earliest and latest finish times (EFT and LFT). The total float indicates the amount of time by which the start or the finish of an activity can be delayed without affecting the project's scheduled completion date.

Activities share the total float with other activities when they are on the same path. Hence, a delay within the total float range of an activity also consumes its succeeding activities' total floats. Because of its sharing property, project participants are highly concerned about the total float. This sharing property is a very important concept to understand especially in change orders or time-related claims where the float ownership issues are involved between the owner and the contractor or between activities [Bartholomew 1998; de la Garza and Vorster 1991; Callahan et al. 1992].

If the total float of an activity is zero, the activity becomes a critical activity. One or more continuous chains of critical activities constitute a critical path, which is the longest path in the network. The critical path is an important by-product of the CPM because it indicates the shortest project duration and shows which activities need to be

[1] This study employs the Precedence Diagramming Method (PDM) or the Activity On Node (AON) method with four activity relationships, which are finish-to-start, finish-to-finish, start-to-start, and start-to-finish. The lag time available in the PDM is negative, zero, or positive. Activity splitting is not considered in this study.

managed more carefully than the other non-critical activities [Ahuja 1976; Willis 1986]. If most activities of the schedule were on the critical path, it would not be easy to efficiently manage all critical activities simultaneously during the course of the project.

On the other hand, the free float of an activity indicates the amount of time for which the activity can be delayed without affecting the early times (EST or EFT) of its immediate successors. Because of this property, in contrast to the total float, the free float is not shared with other activities. The free float can be useful to select activities in resource leveling (smoothing) techniques or to organize tasks at multiple sites for subcontractors [Callahan et al. 1992].

2.2 Resource-constrained Scheduling

The resource-constrained scheduling (RCS) technique attempts to reschedule the project in order to satisfy a number of resource limits. Generally, there are two approaches in RCS: analytical and heuristic methods. Analytical methods such as integer linear programming, dynamic programming, and branch and bound attempt to find the optimum solution in terms of the minimum project duration [David 1975]. However, these methods require a very long computational time making them impractical for real projects. To overcome such a problem, various heuristic methods have been developed after the first published work by Kelley (1963). Although heuristic approaches may not find the optimal duration for resource-constrained conditions, they provide reasonable solutions in practical time [Boctor 1990; Hegazy 1999; Lew et al. 1999].

Typically two types of heuristic methods are used: serial and parallel methods [Kelly 1963; Moder et. al. 1983]. In the serial heuristic method, all activities are sorted by the network sequence and by certain other priorities (if some are parallel), and then each activity is scheduled one at a time (serially) in the order of the sorted list. When an activity cannot be scheduled at its early start time (depending on its precedent activities) due to lack of resources, the activity start time is delayed until it meets its resources requirement. On the other hand, the parallel method schedules several activities at the same time (in parallel). At every given time interval, eligible activities are searched and sorted by certain priorities, and then they are scheduled in the order of the sorted list. When an activity cannot be scheduled due to lack of resources, the activity is reconsidered in the next time interval. To develop the RCPM algorithm, the serial heuristic scheduling method has been adopted because it is P3's algorithm for RCS [Primavera 1999].

2.3 Criticality in Resource-constrained Scheduling

The following sections present previous research to find the resource dependency between activities for a correct float calculation in resource-constrained schedules. Each section reviews and evaluates the proposed method.

2.3.1 Wiest's Study (1964)

Review

This study proposes a systematic procedure to identify the total float of each activity after RCS. The total float of an activity is the difference between the scheduled start time of the left-justified schedule (EST) and the scheduled start time of the right-justified schedule (LST). In the left-justified schedule as shown in Figure 2.1, no activity can start earlier by left shifting of that activity alone without violating technological relationships and resource constraints. A heuristic RCS result can be a left-justified schedule because it finds the earliest possible time for each activity to start, considering technological relationships and resource availability [Moder et. al. 1983; Ritchie 1985].

Given the left-justified schedule and resource constraints, the right-justified schedule can be made according to following priority rules:

1) Right shift activities with the latest EFT first.
2) In the case of a tie in the EFT, right shift activities with the latest possible LST first
3) In the case of a tie in the LST, right shift activities with the smallest resource requirement first
4) In the case of a tie in the resource requirement, right shift activities with the smallest ID first.

Figure 2.2 is the right-justified schedule of Figure 2.1 based on the above rules. In this right-justified schedule, only *Activity 1* and *2* could have a different start time. After comparison with its left-justified schedule, *Activity 1* has a two-day TF and *Activity 2* has a one-day TF, and *Activity 3* and *4* are critical activities having a zero-day TF. Hence, the critical sequence[2] of this schedule is *Activity 3* and *4*. Figure 2.3 shows TF with dotted lines and the critical sequence with bold lines in the left-justified schedule.

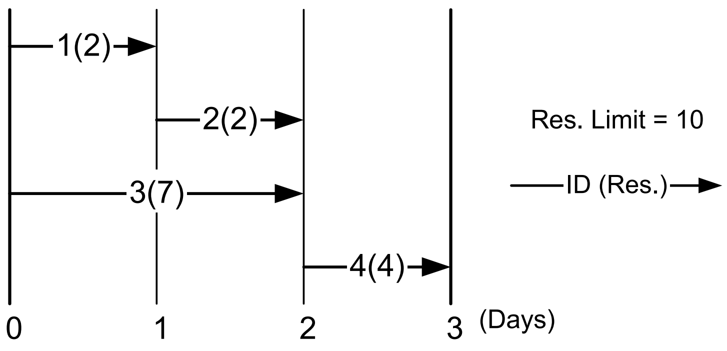


Figure 2.1 Left-justified Schedule in Wiest's

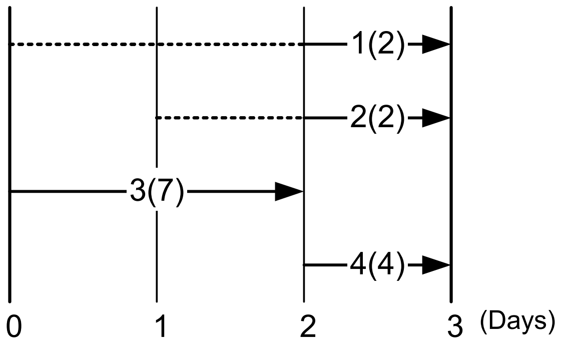


Figure 2.2 Right-justified Schedule in Wiest's

[2] TF calculated in Wiest (1964) reflects both technological relationship and resource constraints. Wiest (1964) named the chain of critical activities as "critical sequence" in that the "critical path" in the CPM normally reflects technological relationships only.

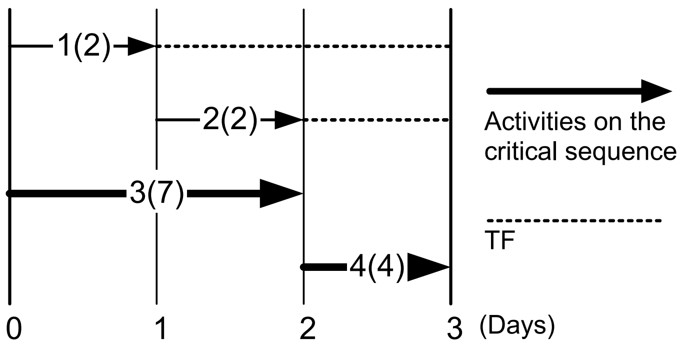


Figure 2.3 TF and Critical Sequence in Wiest's

<u>Evaluation</u>

The critical activities identified in Figure 2.3 are apparent because any time extension of them will affect the overall completion time. However, there are some ambiguities in TF usage. As shown in Figure 2.4, *Activity 1* and *2* cannot be concurrently performed during day 2 due to resource limit (10 units), even though they do not have any dependency as shown in Figure 2.3. In a similar way, the time extension of *Activity 1* within its TF periods is not allowed as shown in Figure 2.5. *Activity 1* can be scheduled on day 3, but the two-day TF of it is not correct unless *Activity 2* is also delayed. This dependent TF between independent activities should be handled in another way.

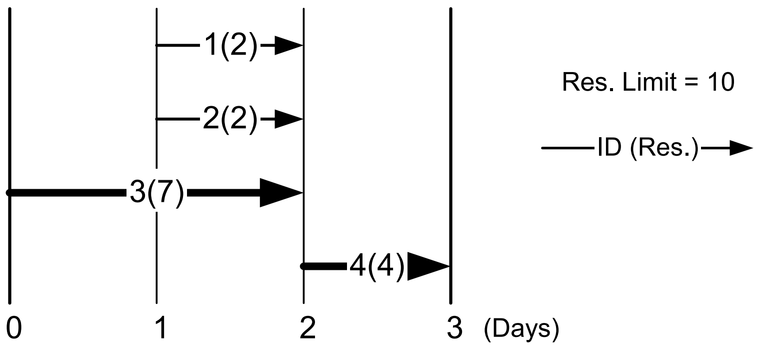


Figure 2.4 Start Time Delay within TF Period in Wiest's

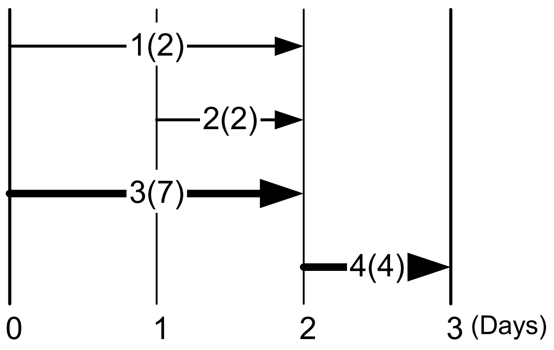


Figure 2.5 Time Extension within TF Period in Wiest's

In addition to the ambiguity in TF usage, this method does not provide resource dependencies, even if there is no such ambiguity between activities. Hence, it will be not easy to predict consequences of any changes such as activity start time delay, activity duration extension, etc., in a large schedule. Rescheduling is required in these changes, but it does not guarantee the work sequence stability and as a result, reorganization may be required with an extra time and cost.

2.3.2 Woodworth's Study (1988)

Review

This study proposes a set of procedures to find correct floats and the critical path in a resource-constrained schedule. The main features of the algorithms are a resource label for each activity and resource-constrained linkages between activities. Basic procedures with a scheduling example demonstrated in the article are as follows.

1) CPM forward and backward passes; Table 2.1 shows the example activity data and Figure 2.6 shows a time scaled network.

Table 2.1 Woodworth's Example Schedule Data[3]

Activity	Duration	Resource	Quantity
2-3	6	A	1
2-4	3	A	1
3-11	4	B	1
4-5	6	A	1
4-6	3	B	1
5-10	8	A	1
6-7	4	B	1
6-8	3	B	1
7-9	6	A	1
8-9	2	B	1
9-10	8	A	1
10-12	5	B	1
11-12	2	A	1
12-13	2	A	1

2) Forward resource constrained scheduling
   - Prior to scheduling, establish a resource pool for each resource type; resource limits are one for each in the example.
   - During scheduling (parallel method), an activity will be delayed if there is not enough resource in the resource pool at the time when it is attempted to schedule; Figure 2.7 shows the result of the forward scheduling using the activity on arrow (AOA) format that the authors have applied.
   - During scheduling, a resource sequence label is given to each activity for an individual resource requirement. Each label consists of a resource ID and usage order. For example, the label *B3* indicates that the activity is the third activity to use resource *B*. The resource label is also kept in each resource category so that the order of resource use can be easily identified. Figure 2.8 shows the activity schedule with resource indexes.

[3] Reprinted from International Journal of Project Management, Vol. 6 (2), Woodworth, B. M., and Shanahan, S., "Identifying the critical sequence in a resource constrained project", 89-96, 1988, with permission from Elsevier

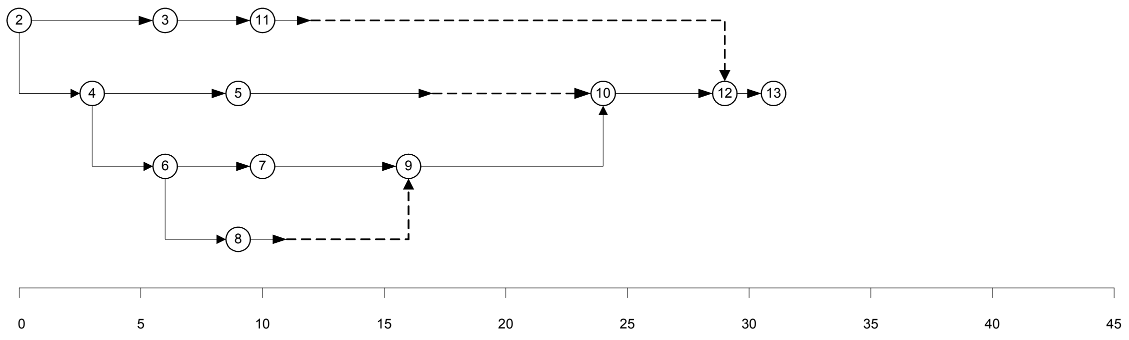


Figure 2.6 Woodworth's Schedule in a Time-scaled CPM Network

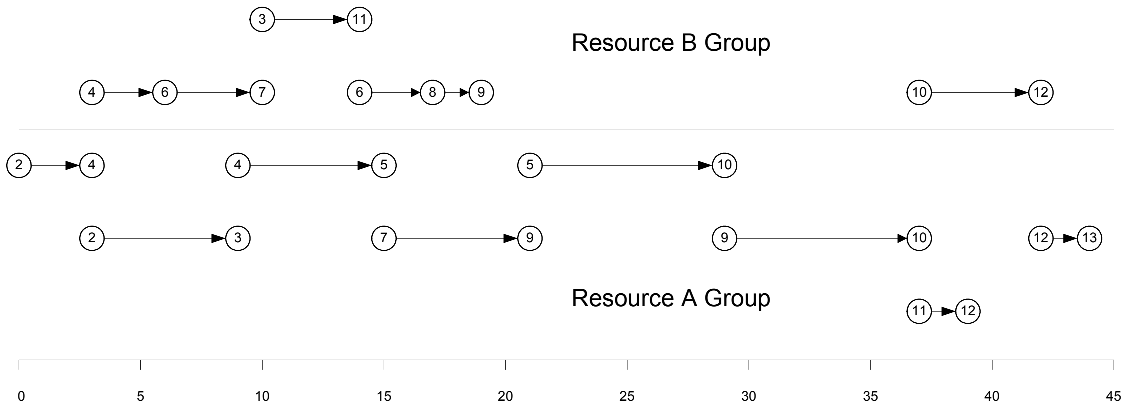


Figure 2.7 Woodworth's Schedule after Forward Resource Scheduling Pass

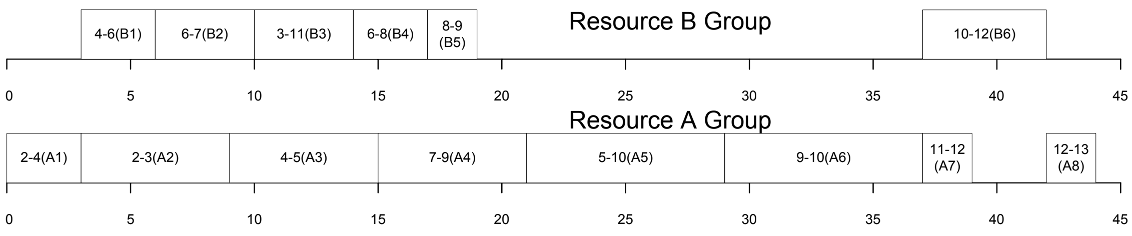


Figure 2.8 Resource Indexes with Schedule in Woodworth's

3) Backward resource constrained scheduling
   - This procedure starts from the terminal activity and visits predecessors like the CPM backward pass except several procedures to make the resource linkages and to find LFT and LST as follows.

- Search backward from the current activity (in-progress activity) considering technological dependencies and/or resource usage history, and find the immediate predecessor activities of the current activity. The search by resource usage history is available since each activity has a resource index and each resource category has the list of activities as shown in Figure 2.8. An identified previous activity could be either a technological predecessor or a resource-constrained predecessor, or both. The LFT of each identified activity is set equal to the LST of the current activity. A resource-constrained link is created if the identified activity is not a technological predecessor of the current activity. For example as shown in Figure 2.9, the backward pass starts from the terminal activity (*12-13*). A search can find *Activity 10-12* and *11-12* as predecessor activities, and the LST of *Activity 12-13* is assigned to their LFT. A search from *Activity 11-12* finds *Activity 9-10* as a resource constrained predecessor (based on the resource index) and a resource link is created between *Activity 9-10* and *11-12*. Figure 2.10, a scanned copy from the article, shows the result of backward resource constrained scheduling.

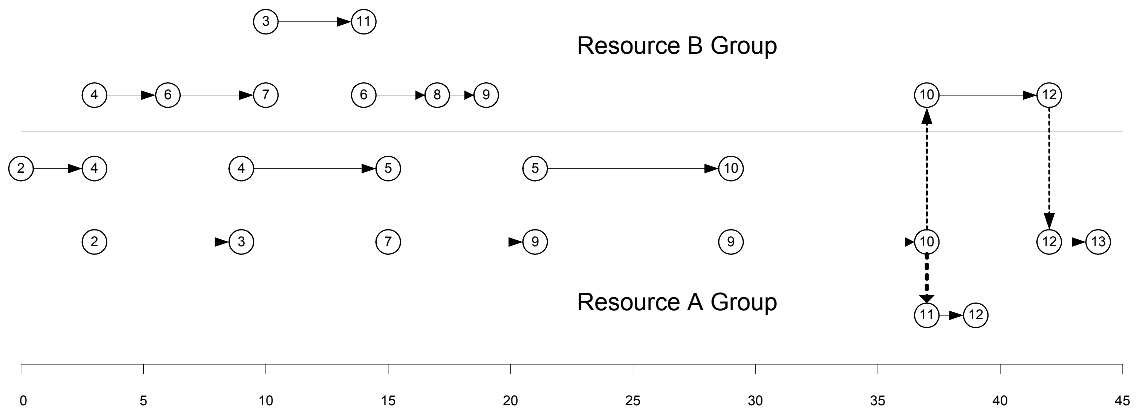


Figure 2.9 Backward Pass in Woodworth's

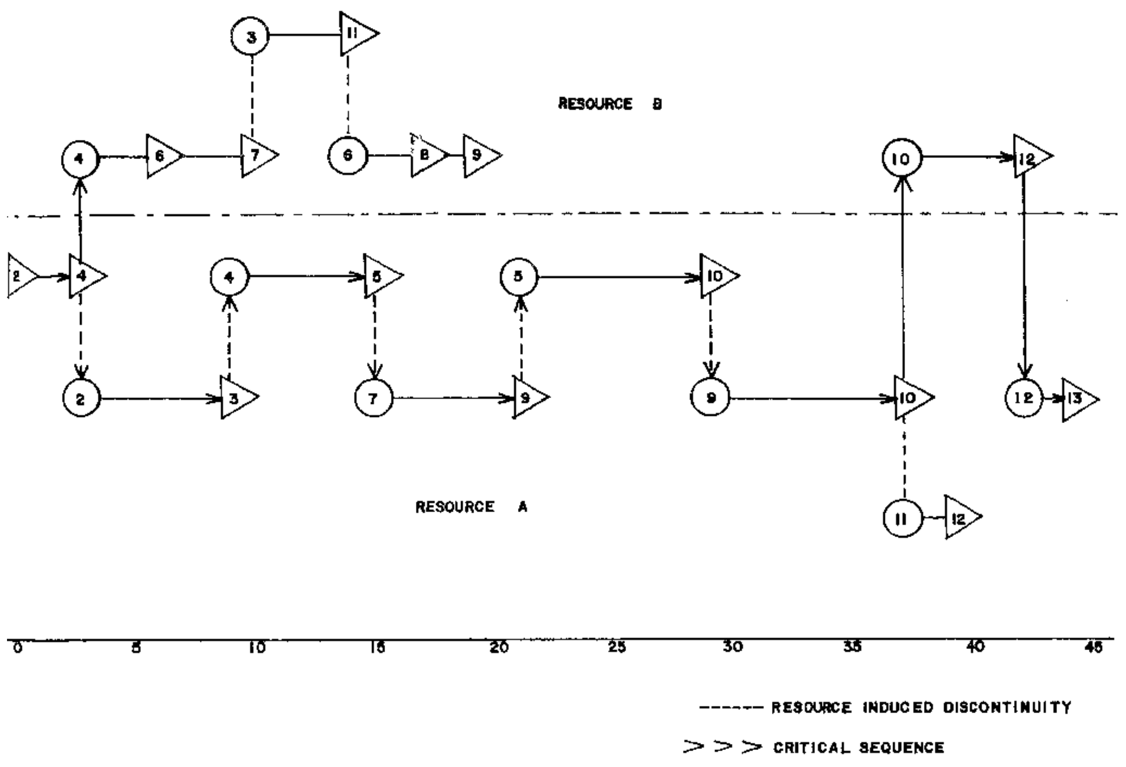


Figure 2.10 Final Schedule in Woodworth's[3]

Evaluation

The proposed algorithm suggests a **resource-constrained link** in order to show resource usage flows and to identify correct floats in RCS. With the given example of very restricted resource condition such as a limited number of resource requirements and availability and the uniformly distributed daily resource requirement condition (*Resource A*), the algorithm performs well. However, it does not provide a clear method as to how any multiple resources and not-preferable distribution of daily requirements can be handled. Any activity with any number of any resource types should be resolved in a real schedule.

In addition, the algorithm could generate a wrong TF. Figure 2.11 is the initial schedule without resource limit. Once RCS is applied, all activities could become critical based on the algorithm. As shown in Figure 2.12, the search from *Activity 4* could find two activities, *Activity 1* and *3*, for the immediate predecessors. Then day 2, the LST of *Activity 4*, will be assigned to the LFT of *Activity 1* and *3*. However, *Activity 3* can be concurrently executed with *Activity 4*, so the LFT is day 3, having a one-day total float.

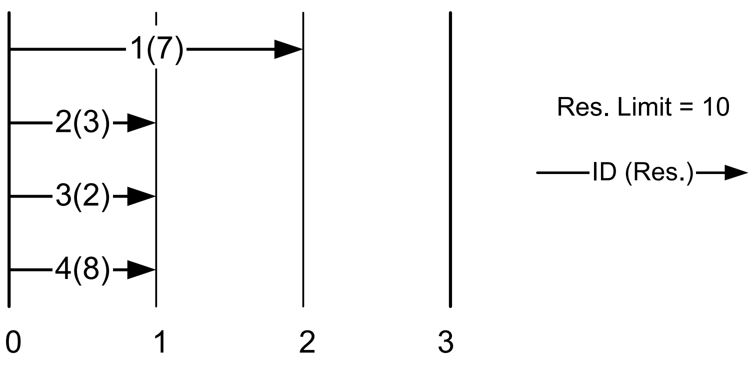


Figure 2.11 Initial Schedule in Woodworth's

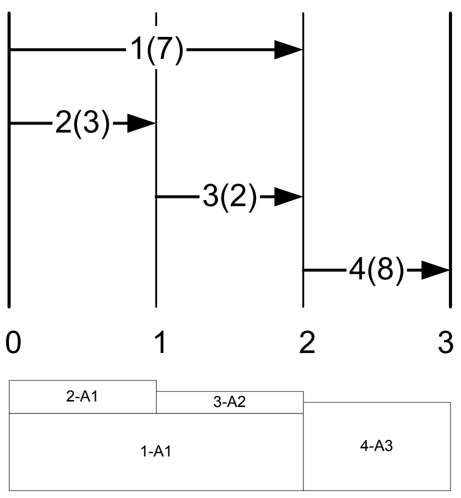


Figure 2.12 Resource-constrained Schedule with Resource Indexes in Woodworth's

2.3.3 Badiru's Study (1993)

This study proposes a critical resource diagram for a better resource management. The diagram can be translated from a normal scheduling network, and the format of the diagram is same as the AON network. In the diagram, a node represents a resource unit and has its own duration originated from the resource owner (activity). A node also has links to other nodes based on the technical relationship between activities. Although this diagram is not directly related to RCS, it proposes making links based on resource dependency between nodes, if a certain resource cannot be used at the same time. In the view of RCS, this additional resource link seems fine if both the maximum availability and the requirement of each resource are identical, because any of the two nodes with an identical resource cannot be scheduled at the same time. However, if the maximum availability and the requirement are different, the proposed method does not provide answers as to how the resource links are created.

2.3.4 Bowers' Study (1995)

Review

This study also suggests creating resource links for a solution to find correct floats and the critical path in a resource-constrained schedule. The AON network is adopted to represent resource links more efficiently than the AOA network, which was employed in Woodworth and Shanahan (1988). The AOA is awkward, creating many dummy activities to represent the resource links as shown in Figure 2.10. The procedure is similar to that of Woodworth and Shanahan (1988), but the resource link is created before the backward resource pass as shown on Table 2.2.

Table 2.2 Bowers' Algorithm

Step	Procedure	Output
1	Forward Pass, ignoring resources	EST, EFT
2	Backward Pass, ignoring resources	LST, LFT
3	Forward Pass, with resources	EST, EFT
4	Note resource utilization history	Resource links
5	Backward Pass, with resources	LST, LFT

[p. 81, Bowers 1995]

Evaluation

This study does not provide enough information how the resource link can be created. Although the author comments that the method is applicable to activities with multiple units and types of resources, it may not be available to handle complex concurrent activities based on information provided in the article. The only descriptions about creating resource links found in the article are as follows.

During the construction of the resource-constrained schedule (forward pass), the sources of each activity's resources are noted and this information is then used to create additional links explicitly representing the resource dependencies in the network.

The algorithm provides results consistent with the examples examined by Woodworth and Shanahan (1988), using a method involving a repeated examination of the history of the resource availabilities at each allocation decision of the reverse pass, rather than explicitly creating the resource links.

[p. 81, Bowers 1995]

In addition to the lack of description, the algorithm does not consider technological activity relationships, making unnecessary resource links that could lead a wrong float calculation. The article provides two example schedules: one simple and the other relatively big (still simple in terms of resource usage). The resource links found in the simple schedule are feasible, but not feasible in the big schedule. There exist two unnecessary resource links in the big schedule, one of which lead to a wrong float time calculation, while the other does not affect any time data.

Figure 2.13 is a part of the big example schedule in CPM (the rest of the activities are omitted). The maximum availability of *Resource B* is two, and *Activity 11, 12* and *13* are the only activities that require *Resource B* in the whole network of 35 activities. Identified resource link for this partial network according to the resource-constrained schedule data provided in the article is shown in Figure 2.14. It is obvious that any time extension of *Activity 11* does not affect the start time of *Activity 13* because resource availability is two. If there is no technological relationship between *Activity 12* and *13*, the resource link may be required in case both activity 11 and 12 are delayed.

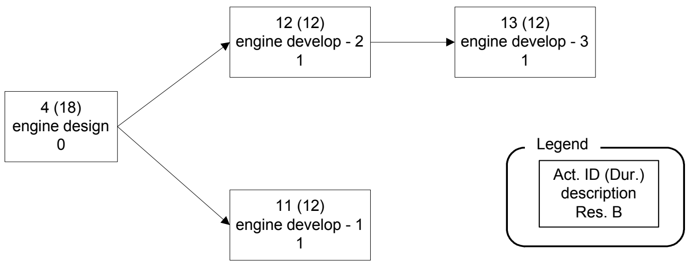


Figure 2.13 Bowers' Example Schedule

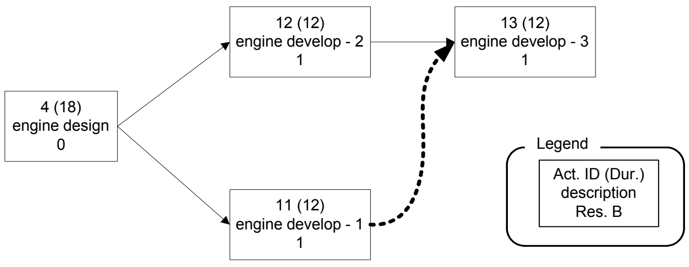


Figure 2.14 Bowers' Example Schedule with a Resource Link

Another unnecessary resource link, as well as the resource link between *Activity 11* and *13*, was found from the network obtained directly from the author as shown in Figure 2.15. On the top right side of the network, the activity "avionics flight trials (1)" and the "avionics flight trials (2)" have a resource link as well as a technological link. This resource link could not be found from the schedule data provided in the article because the technological link overlaps the resource link so that the LST and LFT of the activity "avionics flight trials (1)" are the same whether the resource link exists or not. In addition, the unnecessary resource link depicted in Figure 2.14 is also shown on the left bottom of the network.

These two unnecessary resource links lead to conclude that the proposed algorithm does not consider the technological link, and then may generate wrong float data.

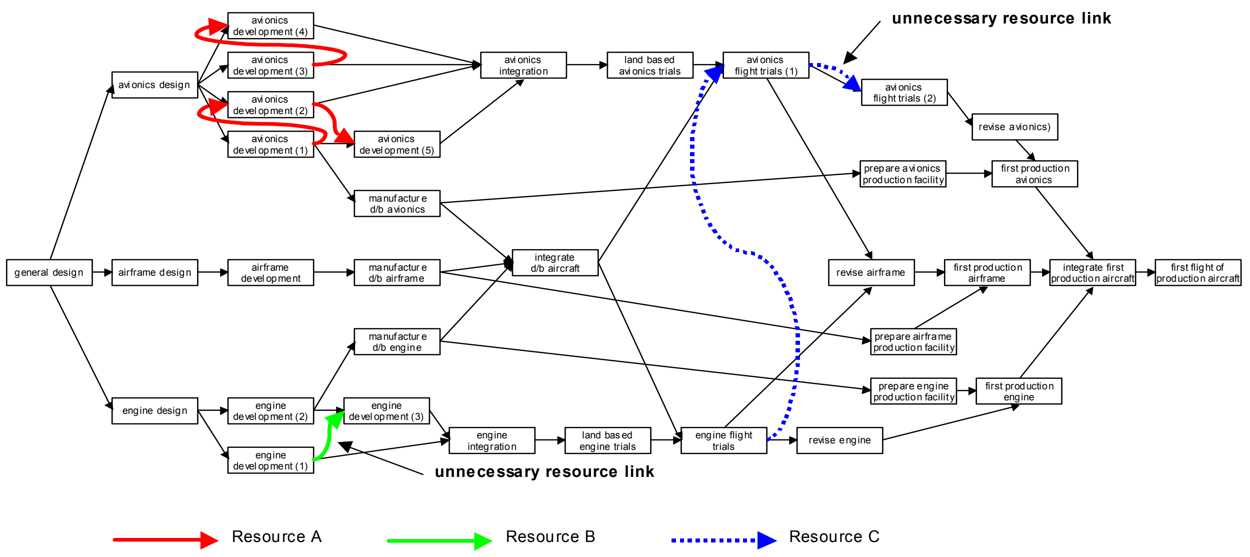


Figure 2.15 Bowers' Example Schedule (complete network)
[network provided by J. A. Bowers, personal communication, 2001]

2.4 Alternative Activity Schedules

This section reviews and evaluates Bowers' study (2000) to find alternative schedules in a resource-constrained schedule.

Review

This study proposes a set of procedures to generate alternative schedules and to measure the start time flexibility of each activity in resource-constrained schedules. The method proposed in Bowers (1995), which identifies resource links in RCS, is applied to calculate late start and finish times. In order to find alternative schedules, the algorithm generates $4n + 2$ schedules per project, including one original schedule, where n equals the number of activities in the project. Among the $4n + 2$ schedules, two schedules are used for basis and four schedules are created for each activity. Among the schedules, only equivalent schedules, which have the same duration as the original schedule, are eligible for consideration. Six stages to generate those schedules are as follows.

0) RCS that generates the original schedule $j = 0$ with a project duration of D_j.
1) RCS repeated with each activity i

   subject to

$LS(i, j) \geq LS(i, 0)$ and

$D_j \leq D_0$

where, $j = i$

$LS(i, j)$ = latest start for activity i in schedule j

The late start constraint is implemented by a dummy activity whose duration = $LS(i, 0)$, predecessor = the project start, and successor = Activity i.

2) RCS repeated with each activity i

subject to

$LS(i, j) \geq LS(i, 0) + \Delta$ and

$D_j \leq D_0$

where, $j = n + i$,

$\Delta > 0$, but as small as possible allowed in the network,

$LS(i, j)$ = latest start for activity i in schedule j

3) Resource constrained scheduling as in stage 0), but $j = 2n + 1$ and but with a reversed network such that all relationships are reversed making an activity's predecessors become its successors

4) The same process as in stage 1), but $j = 2n + 1 + i$

5) The same process as in stage 2), but $j = 3n + 1 + i$

The early and late start times (EF′ and LF′) of each activity of each schedule j in Stage 4) and 5) can be transferred to the original (not reversed) network (ES and LS) as follows.

$ES = D - LF'$

$LS = D - EF'$

where, D = project duration of each resource-constrained schedule

Each stage of 1), 2), 4), and 5) generates n schedules based on the two schedules from stage 0) and 3). The earliest ES (EES), the latest LS (LLS), and the maximum float (F_2) among the schedules and the maximum float range with different schedules (F_3) can be calculated as follows:

$$EES(i) = \min_{j=0}^{4n+2}\{ES(i,j)\}$$

$$LLS(i) = \max_{j=0}^{4n+2}\{LS(i,j)\}$$

$$F_2(i) = \max_{j=0}^{4n+2}\{F_1(i,j)\}$$

$$F_3(i) = LLS(i) - EES(i)$$

where, F_1 (i, j) = float of *Activity i* in *Schedule j*

subject to

$$D_j = D_0$$

Based on comparing $F_1(i, 0)$ with $F_2(i)$ and $F_3(i)$, the existence of alternative schedules of *Activity i* could be identified. If $F_1(i, 0) = F_2(i) = F_3(i)$, then there is no alternative schedule for *Activity i*. But, if $F_2(i)$ or $F_3(i)$ is greater than $F_1(i, 0)$, there is at least one alternative schedule that can be found from acceptable ($D_j = D_0$) schedules among 4*n* schedules. In addition, the difference between $F_1(i, 0)$ and $F_2(i)$ or $F_3(i)$ can be the amount of flexibility during which *Activity i* can be scheduled on a different time period rather than the original schedule without delaying the project completion.

Evaluation

Although the proposed method generates alternative schedules of a resource-constrained schedule, showing flexibility of the original schedule, there are several shortages as follows.

First, the burden of computation cost to generate *4n + 1* schedules is considerable in a relatively big schedule. For instance, a project with 1,000 activities will require 4,001 RCS processes. A CPM process is also required prior to every RCS because of the inserted dummy activity for each schedule. Every RCS process can be terminated at anytime if $D_j > D_0$, but this condition will not be known until the end of the process.

Another shortcoming is that each alternative schedule is independent. In other words, if an alternative *Schedule j* for *Activity i* were selected, then all other alternative schedules would not be valid unless they are identical to *Schedule j*. Then, in order to find alternative schedules for another activity of the selected alternative *Schedule j*, it would be required to execute the whole processes again, generating additional *4n* + *1* schedules.

Finally, an alternative *Schedule j* for *Activity i* is not only for *Activity i* but also for other activities. As mentioned in the work sequence stability issues in RCS, the changed EST of *Activity i* by a dummy activity could alter time data such as EST, LST, TF, FF, etc. of other activities on the same path in the network. If the priority factors of RCS include one of those time data, then unexpected consequence will come out resulting different work sequence. Hence, in order to employ alternative *Schedule j*, reorganization would be necessary with additional cost and time.

2.5 Resources in P3 [Primavera 1999a; Primavera 1999b]

P3 provides many, complex options for resource handling in order to accommodate various needs by users. Relatively important options related to the RCPM development could be the resource option, the priority option, the leveling option, and the smoothing option. This section will briefly explain these options.

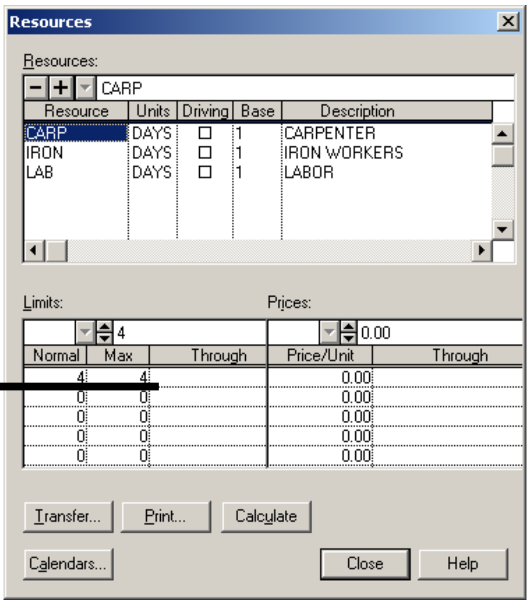
2.5.1 Resource option

P3 allows its users to select resources that are limited in use. Since not all resource types are necessarily limited, users can select only the limited resources from the resource dictionary. There are two types of resource limits as shown in Figure 2.16: *normal* and *maximum*. The normal resource limit is **typical** availability per planning unit such as a day, a week, etc., and the maximum limit is the **highest** availability per planning unit. Those limits of a resource can be changed; six sets of *normal* and *maximum* limits are available in P3 throughout the project duration.

2.5.2 Priority Option

As shown in Figure 2.17, P3 allows users to set a priority list that can be used to break a tie when more than one activity compete to be scheduled at the same time in RCS.

The activity with higher priority is always scheduled prior to the activity with lower priority. Since P3 normally generates a unique scheduling output per priority set, users can simulate a project to achieve a better schedule. More than 60 items are available for the priority entry.


Resources

Resources: CARP

Resource	Units	Driving	Base	Description
CARP	DAYS	☐	1	CARPENTER
IRON	DAYS	☐	1	IRON WORKERS
LAB	DAYS	☐	1	LABOR

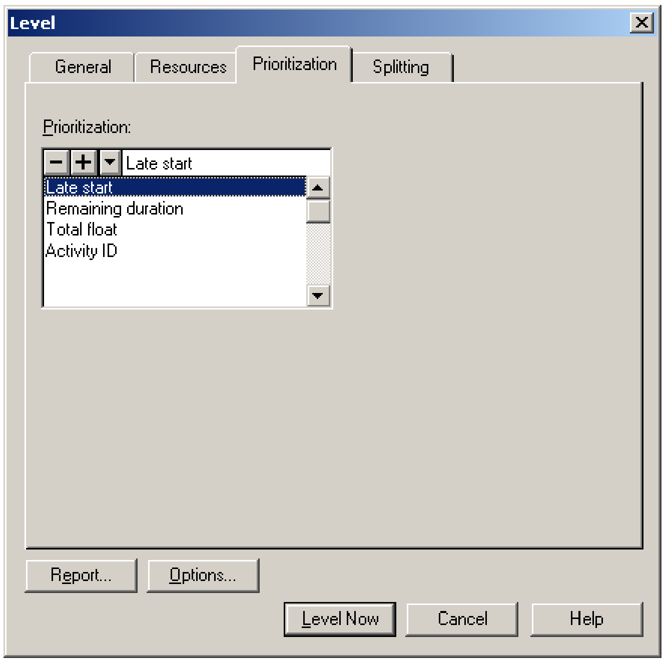
Limits: 4

Normal	Max	Through
4	4	
0	0	
0	0	
0	0	
0	0	

Prices: 0.00

Price/Unit	Through
0.00	
0.00	
0.00	
0.00	
0.00	

Figure 2.16 Resource Creation Dialog Box in P3


Level

Prioritization: Late start

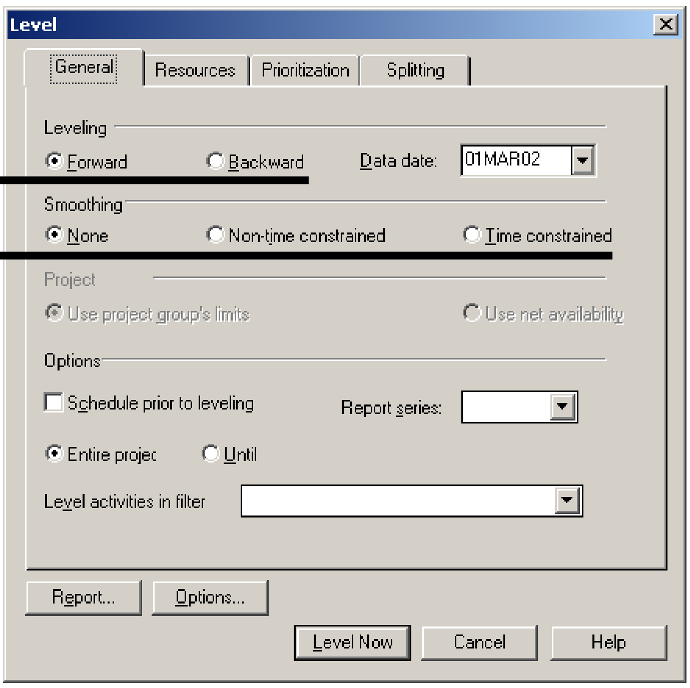
Late start
Remaining duration
Total float
Activity ID

Figure 2.17 Prioritization Dialog Box in P3

2.5.3 Leveling Option

There are two choices in the resource leveling order as shown in Figure 2.18: *forward* and *backward*. Both apply the serial scheduling method (one at a time) and any tie is broken by the activity priority order in the forward method and by the priority reverse order in the backward method. The forward resource leveling starts from the first activity in the network and ends at the last activity in the network. The early possible start times of each activity can be identified with this option.

On the other hand, the backward resource leveling finds the latest possible start times of each activity in a given schedule. This method starts from the last activity in the network and ends at the beginning of the network in the reverse sequence of the activity. If there is not enough resource for an activity, the activity will be moved backward in the backward method, while the activity will be moved forward (delayed) in the forward method. In other words, an activity can start earlier than its EST in the backward method, whereas an activity can be delayed over its LST in the forward method.



Leveling

[✓ Forward] Backward Data date: 01MAR02

Smoothing

[✓ None] Non-time constrained Time constrained

Project

[✓ Use project group's limits] Use net availability

Options

☐ Schedule prior to leveling Report series:

[✓ Entire projec] Until

Level activities in filter

Figure 2.18 Leveling and Smoothing Option Dialog Box in P3

2.5.4 Smoothing Option

There are three choices in the smoothing option as shown Figure 2.18: *none*, *none-time constrained*, and *time constrained.* These options affect leveling (forward and backward) processes. In the leveling process, P3 first tries to schedule each activity within its TF range with the *normal* resource limit, whatever the smoothing option is. If scheduling within its TF period is not available, P3 tries again to schedule the activity from its original early possible time using a different method according to the smoothing option.

1) When the option "none" is selected, P3 simply uses the *maximum* resource limit to check resource availability. If the activity still cannot be scheduled within the TF period, it will be delayed beyond the TF period.

2) When the option "non-time constrained" is selected, P3 increases available resources by 10% of the difference between *normal* and *maximum*, and tries again to schedule the activity within TF period. This process can be made repetitive until the resource availability is reached to the *maximum* limit. If the activity still cannot be scheduled within the TF period, P3 delays the activity beyond the TF and tries to schedule assuming the *maximum* resource limit.

3) When the option "time constrained" is selected, no activity can be scheduled beyond its TF period. Instead, P3 doubles the *normal* and the *maximum* limit and proceeds as the "none" option does. If the activity still cannot be scheduled within the TF period, P3 schedules the activity as its original early time and gives a warning message to users.

2.5.5 P3 Options for RCS

In order to accomplish a resource-constrained schedule in P3, the *normal* and *maximum* limit should be identical as shown in Figure 2.16. Otherwise, P3 applies resource-leveling (smoothing) procedures anywhere in the network. In addition, the "time constrained" smoothing option must not be selected as shown in Figure 2.18 to avoid any violations of resource overuse.

2.6 Summary of Literature Review

This chapter has performed background studies related to developing the RCPM algorithm. The summary of this chapter is as follows.

- CPM provides important information such as total/free floats and the critical path. Due to its sharing property and impact to the project completion time, the total float is a major concern of project management and delay impact analysis.
- Heuristic RCS techniques such as serial or parallel methods are eligible for construction projects because they provide reasonable answers in practical time. This study employs the serial RCS technique, which is the P3 RCS algorithm.
- Review of previous research provides useful information to identify floats in a resource-constrained schedule. However, each method has limitations as shown in Table 2.3. Wiest's does not provide resource dependencies between activities, Woodworth's does not handle multiple resources with multiple activities, and Bower's does not consider technological relationships. This lack of capability of each method has brought requirements for further research.
- Various P3 options, such as resource limit, priority, leveling, and smoothing options, to handle resources are examined. In order to accomplish RCS in P3, the *normal* and *maximum* resource limit should be identical and the *time constrained* smoothing option must not be selected.

Table 2.3 Functionality of Previous Research

	Wiest's	Woodworth's	Bowers'
Existence of Res. Link (RL)	X	O	O
Multi. Res. w/ Multi. Act.	O	X	N/A
RL considering Tech. Link	N/A	O	X

O : Yes, X : No

Chapter 3

RCPM Algorithm

3.1 Overview

As proposed in previous studies [Badiru 1993; Bowers 1995; Woodworth and Shanahan 1988], this study has adopted making a resource-constrained link between activities as an approach to identify real float in RCS. The RCPM algorithm is mainly composed of five steps. The overview is shown in Table 3.1. In order to make the problem simple, the maximum available resource amount is assumed constant for the whole project duration, and activity splitting and resource leveling (smoothing) are not considered. For better understanding of the processes, a simple example (SEMP) schedule, shown in Table 3.2, is used throughout all steps. Besides, a fenced bar chart technique is adopted for convenience of identifying activities' relationship and daily resource requirement [Melin and Whiteaker 1981].

Table 3.1 RCPM Algorithm Process Overview

Step	Function
1	CPM forward and backward passes
2	Serial resource constrained scheduling, making resource links
3	Backward pass with technological and resource links
4	Finding unidentified resource links
5	Finding alternative schedules

Table 3.2 Activity Data of Simple Example Schedule

ID	Duration	Resource Requirement		Successors
		Type A	Type B	
A	2	0	0	B, C, D, E
B	4	1	0	F
C	5	1	0	G
D	6	1	0	G
E	2	0	1	G
F	3	0	1	G
G	2	0	0	-

3.2 Step-by-Step Algorithm

3.2.1 Step 1: CPM

Similar to traditional RCS techniques, the CPM is employed to find the early/late start/finish times and the TF of each activity, applying forward and backward passes assuming unlimited resources. The CPM result of SEMP is shown in Figure 3.1, in which the early/late times are calculated and the critical path (bold line) is identified. The fenced bar chart in Figure 3.2 is a direct conversion of Figure 3.1, and the daily resource requirements are shown on the bottom of it. If the maximum available resources are assumed as two for *Resource A* and one for *Resource B*, day 3 through day 6 are over the given resource limit. Now, a RCS technique is required to remove these resource over-uses.

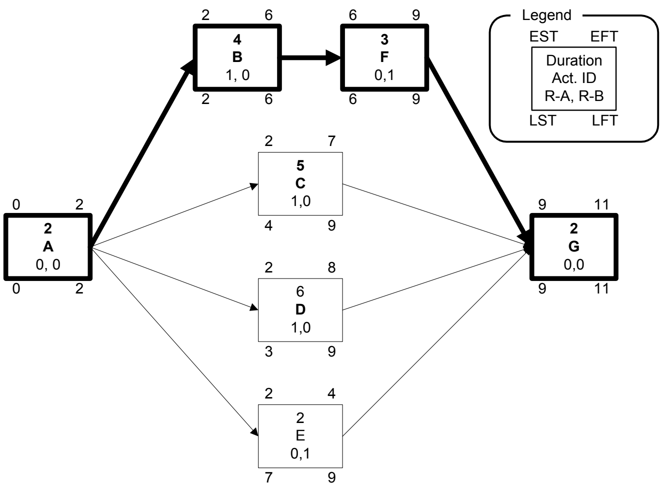
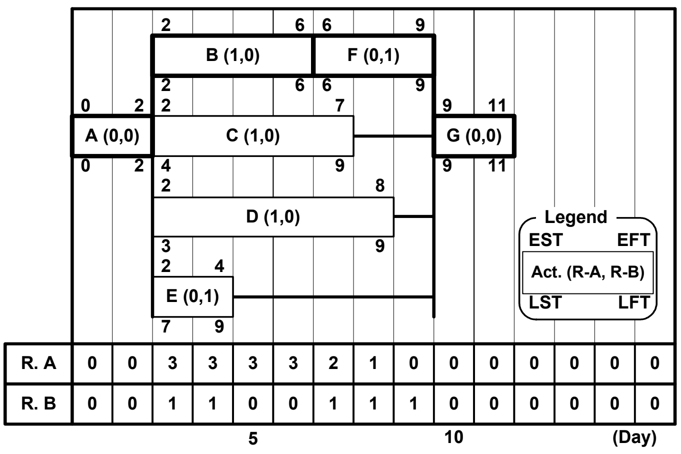


Figure 3.1 CPM Schedule in Step 1 (AON)



R. A	0	0	3	3	3	3	2	1	0	0	0	0	0	0	0
R. B	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0

Figure 3.2 CPM Schedule in Step 1 (Fenced Bar Chart)

3.2.2 Step 2: Serial Method with Creating Resource-constrained Link

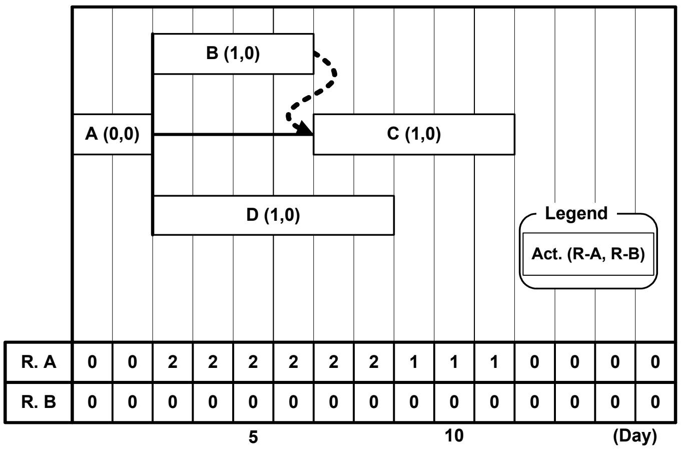
Based on the results of step 1, a serial RCS technique [Kelly 1963; Moder et. al. 1983] is applied. In RCS, all predecessors of an activity must be scheduled first prior to scheduling the successor, as in the forward pass of the CPM. Otherwise, the early start or finish time of a successor in the Precedence Diagramming Method (PDM) network could violate the relationship condition with its predecessors. Thus, for the serial method, all activities are sorted by the network sequence. If there is a tie in the sequence, it can be resolved, in this study, by the order of smaller LST, shorter duration, smaller TF, and smaller ID. Then, each activity is checked for resource availability and scheduled in the order of the sorted activity list.

The activity that is currently being evaluated in the process is referred to as a current activity. If there are enough resources, the current activity can be scheduled at its earliest possible time according to its technical relationships with precedent activities. However, if there are not enough resources for the current activity, it should be delayed until it acquires available resources. After being delayed day-by-day, the current activity is to encounter a certain date from which enough resources are available throughout its whole duration. At this certain time, all or some amount of resources that caused the current activity's delay should have been released from some other activities' completion. In other words, the completion of those activities immediately makes it possible for the current activity to take the resources and to be executed with enough resources for its whole duration. This dependency caused by the resource transfer can be treated as a logical relationship in the CPM. At this point, one or more resource links are created between the current activity and all or some of the just completed activities whose time extension directly affects the start time of the current activity due to a resource limit.

If there is no single activity that directly affects the current activity start time, then a resource link can be created for one activity according to a priority rule. This case happens when the current activity and any just completed activity can be scheduled concurrently but the current activity is delayed due to the combined resource requirement of the multiple just completed activities. For simplicity, the priority is given to the activity that requires most resources. A tie can be resolved by the reversed order of the sorted activities in step 2. Once resource links are created, any time extension of the predecessor will delay the start time of the current activity.

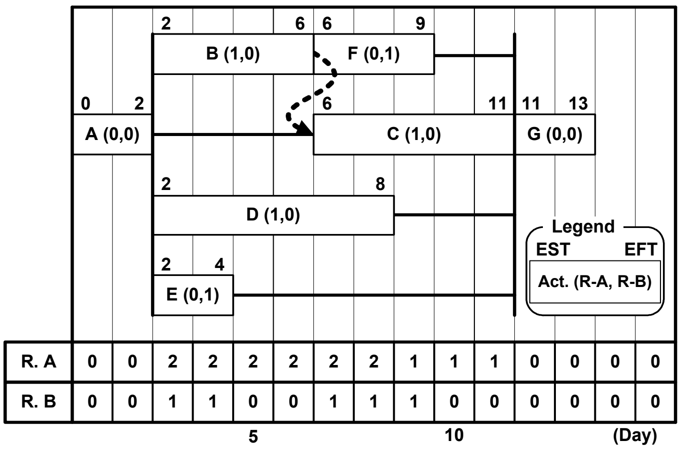
In the SEMP schedule, each activity will be checked one-by-one in the sequence of "*A, B, D, C, F, E, G*" based on the activity sequence and other priority orders. *Activity A, B,* and *D* can be scheduled as indicated by the original CPM schedule. However, *activity C* should be delayed until enough resources are available, because it cannot be scheduled as original due to the limit of *resource A*. *Activity C* can start right after *Activity B*'s completion, which is the earliest possible start time for *Activity C*. Since *Activity B*'s completion releases one of *resource A* that is enough for *Activity C*'s performance for the whole duration, a resource link is added between *Activity B* and *C* as shown in Figure 3.3. *Activity F, E,* and *G* do not encounter any resource limit, so that they can be scheduled as initial relationship. Figure 3.4 illustrates the result of step 2 showing that no activity can start earlier than the current start time and that overall duration has been extended by two days due to the chain effect of *Activity C*'s delay.

After step 2, resource links are created, and the scheduled start and end time of an activity automatically become EST and EFT, but LST and LFT are still not identified at this step.



R. A	0	0	2	2	2	2	2	2	1	1	1	0	0	0	0
R. B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.3 Resource Link Creation in Step 2



R. A	0	0	2	2	2	2	2	2	1	1	1	0	0	0	0
R. B	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0

Figure 3.4 RCPM Schedule in Step 2

3.2.3 Step 3: Backward Pass

A backward pass is required to find LST and LFT considering both resource links and original technological relationships. Figure 3.5 shows the result such as late times and the critical path. Although it appears better than the traditional resource-constrained schedule as shown in Figure 3.6 or 3.7, in which only two critical activities are identified, one more process is required to figure out correct late times.

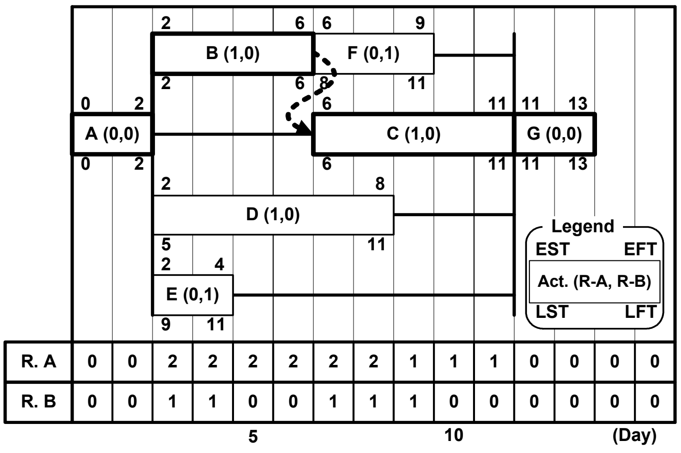


Figure 3.5 RCPM Schedule in Step 3

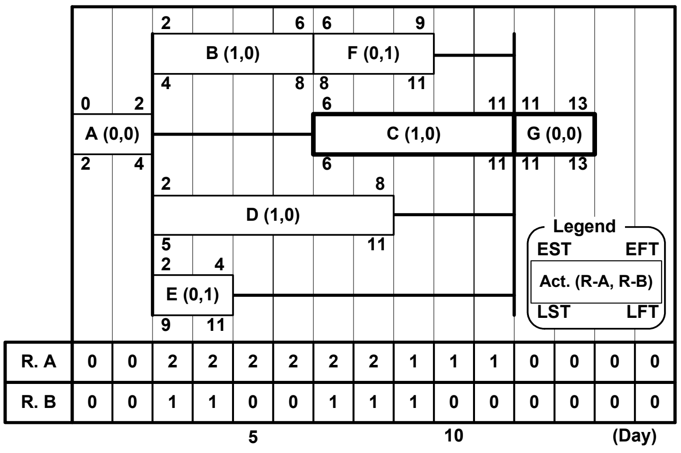
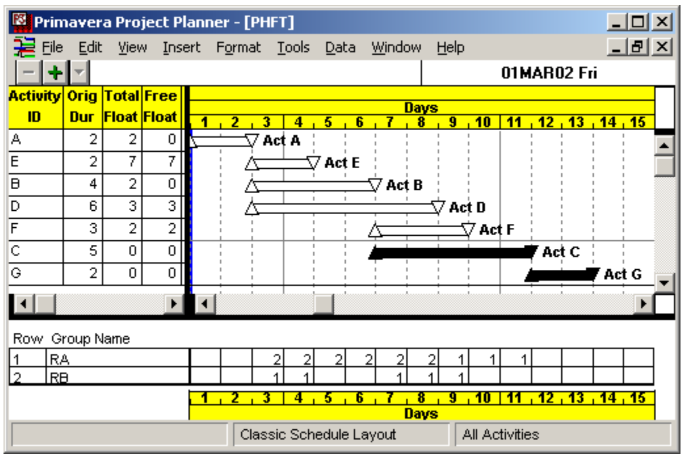


Figure 3.6 Traditional RCS Output


Primavera Project Planner - [PHFT]
File Edit View Insert Format Tools Data Window Help
01MAR02 Fri
Act A
Act E
Act B
Act D
Act F
Act C
Act G
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float	Free Float
A	2	2	0
E	2	7	7
B	4	2	0
D	6	3	3
F	3	2	2
C	5	0	0
G	2	0	0

Row	Group Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	RA			2	2	2	2	2	2	1	1	1				
2	RB			1	1			1	1	1						

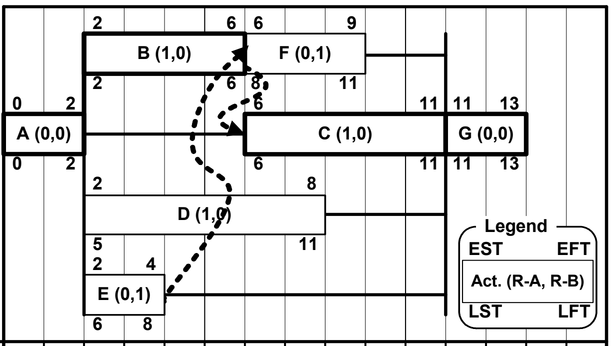
Figure 3.7 Traditional RCS Output in P3

3.2.4 Step 4: Finding More Resource Links

In the result of step 3, the TF of a critical activity is obvious in that any delay of it will extend the project completion time. This means that there is no need to check again for its correct float. However, the activity of which TF is not zero may not have its full floats if there is any resource constraint for the TF period. In Figure 3.5, for instance, *activity E* cannot have its full TF, because *activity F* requires one of *resource B* and the availability of this resource is only one. This is so because resource links in step 2 are created only if the pre-scheduled activities release resources with their completion enabling the delayed activity to start immediately. In other words, there are no resource links between activities if the completion of the prescheduled activities does not immediately affect the delayed activity's start, although the released resources are transferred to the delayed activity.

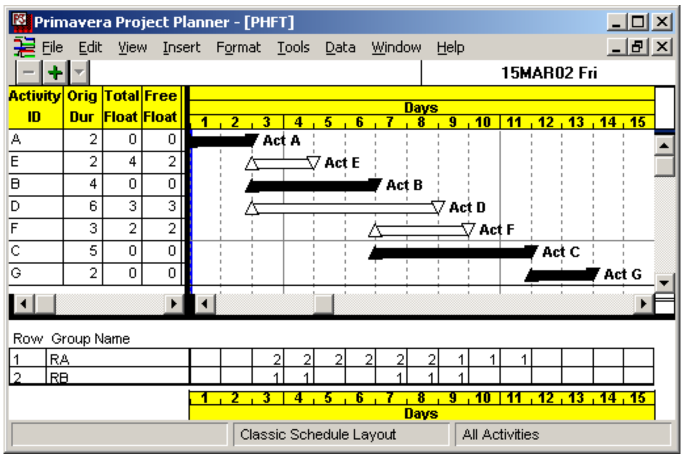
In order to find those unidentified resource dependencies, another sorted activity list is employed for a systematic approach. All activities are sorted by the reverse network sequence, and, if there is a tie in the sequence, it can be resolved by the order of shorter duration, longer TF, and larger ID. In the order of the list, each activity of non-zero TF is checked by delaying the completion time day-by-day up to its TF period. One or more resource links are created between activities that have resource dependency that affects (reduces) the TF of the activity on the process.

In the SEMP schedule, each activity will be checked one-by-one in the sequence of "*G, C, F, D, B, E, A*" based on the priorities. Among the activities, *activity F, D,* and *E* have total floats of 2, 3, and 7, respectively. *Activity F* and *D* can have their full TFs, but *activity E* could not have its full TF due to the limit of *resource B* and *activity F*. Hence, a resource-constrained link between *activity E* and *F* is created as shown in Figure 3.8.



R. A	0	0	2	2	2	2	2	2	1	1	1	0	0	0	0
R. B	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0
					5					10					(Day)

Figure 3.8 RCPM Final Schedule after Step 4



Activity ID	Orig Dur	Total Float	Free Float
A	2	0	0
E	2	4	2
B	4	0	0
D	6	3	3
F	3	2	2
C	5	0	0
G	2	0	0

Row	Group Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	RA			2	2	2	2	2	2	1	1	1				
2	RB			1	1			1	1	1						

Figure 3.9 RCPM Final Schedule in P3

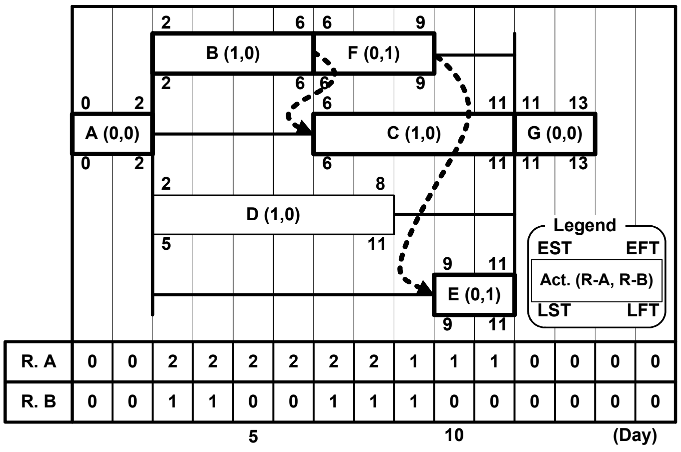
Figure 3.9 shows the P3 schedule with incorporated resource links. Now, this P3 schedule becomes an RCPM schedule that has correct floats with considering limited resources. This final schedule can be acceptable with the current known conditions such as resource limit, technological dependency, equipment condition, material delivery time, etc., but the RCPM algorithm continues to one more step to find alternative schedules. Due to the characteristics of the resource link that cuts into the TF period by technological relationship (in RCS), some activity that has successors by resource links can be delayed beyond its TF period (in RCPM) without affecting the project completion time. Next section describes how these alternative schedules can be identified.

3.2.5 Step 5: Finding Alternative Schedules

In this step, every activity that has a successor by a resource link will be checked to see if an alternative schedule is available. For each activity, all resource links to successors will be temporarily removed and a certain available period will be evaluated if the activity can be executed without violation of technological links and resource constraints. The available period starts from the next day of the LFT of the current activity and continues to the earliest LST of all successors by the technological links (or the project completion time if no successor exists). If there is no alternative schedule for the activity, the original resource links will be restored, but if there is any alternative schedule, the original resource links will be permanently removed for the alternative schedule of the activity and new resource links may be created according to the resource usage condition.

A schedule can have many alternative schedules, among which each activity can have several alternative schedules, but some alternative schedules are dependent. In other words, if one alternative schedule is selected, then some alternative schedules may not be available depending on resource constraints or technological relationships.

In the SEMP schedule from Figure 3.8, *activities B* and *E* have successors with resource links. For *activity B*, there is no available period so that the initial resource link is kept. However, for *Activity E*, day 10 and 11 are available periods and thus it can start right after *activity F*'s completion without delaying the project completion. The original resource link from *activity E* to *activity F* is removed and a new link from *activity F* to *activity E* is created as shown in Figure 3.10. In this schedule, *activity E* and *F* have become critical because of the new link caused by the *resource B*. This schedule appears inferior to the previous one since it makes two additional activities critical. This could however be valuable in some special cases. For instance, if *activity E*'s start from the original schedule (Figure 3.8) should be delayed at least five days due to some reasons that were not considered during scheduling process, such as unexpected equipment repair, material delivery delay, etc., the project completion will be extended by at least one day since the TF of *activity E* is only four. In this special case, an alternative schedule can be selected.



R. A	0	0	2	2	2	2	2	2	1	1	1	0	0	0	0
R. B	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0

Figure 3.10 Alternative Schedule of Activity E

Chapter 4

Scheduling with Multiple Calendars

4.1 Overview

Many sources of construction management literatures describe how to schedule a construction project with continuous working days. Some of them also provide how to apply one calendar to the schedule, emphasizing the importance of the calendar-based schedule. The most common way to apply one calendar is directly translating working days to calendar dates [Ahuja and Nandakumar 1985; Battersby 1967; Clough et al. 2000; Knutson 2001]. The time data of each activity from the continuous working day (non-calendar) schedule can be simply converted to the correspondent calendar date from the cumulative working days counted on the calendar. In this way, every activity can have early and late times in calendar dates.

In a construction project, however, it is common to have multiple calendars according to work properties, weather conditions, resource availabilities, and so on [Lock 2000]. In a project with multiple calendars, each calendar can contain a unique workday pattern per week. For example, a construction schedule can have three project calendars: one for 3-day workweek (Mon., Tue., and Wed.), another for 5-day workweek (Mon. through Fri.), and the other for 7-day workweek (no holidays or weekend). Of course, any activity in the same path of the schedule can use any of the calendars.

Most project management software such as P3 and MS-Project can handle multiple calendars, but the background theory has not been disclosed. Thus, users of the system simply assume without clear knowledge that the time data generated by those software packages are correct. However, through this study, it has been identified that P3 generates incorrect dates in start-to-finish with 0 lag in a schedule with two calendars and that some P3 outputs are contradictory in applying working and non-working days.

In addition, the TF and FF calculated from the software packages does not hold traditional definitions, possibly causing misunderstandings between project participants [Korman and Daniels 2003. Wallwork 2002; Wickwire et al. 2003]. For example, according to the traditional TF concept, a one-day delay to an activity having one-day of total float must not affect the project completion time. However, in a schedule with multiple calendars, this one-day delay could delay the project completion time depending on the calendars and activity relationships. This property should be reflected not only in a construction time-related specification clauses concerning change order, liquidated damage, float usage, and so on, but also in a schedule impact analysis.

This chapter provides how to do the forward and backward passes with multiple calendars. The method covers the four relationships of the Precedence Diagramming Method (PDM) with lags of zero, positive, and negative[1] values. It also shows how RCPM handles multiple calendars when it tries to find resource links.

4.2 Basic Rules

In the forward and backward passes, always two activities are considered along with the relationship type and the lag time. In the forward pass, the successor's early time (EST or EFT) is decided by the predecessor, and, in the backward pass, the predecessor's late time (LST or LFT) is decided by the successor. If they are using the same calendar, handling the calendar is straightforward. However, if they are using two different calendars, more consideration is required. The approach is slightly different in each case of relationship type and lag amount, but a set of basic rules as below is applied to most cases. In this rule, it is assumed that the scheduling unit is one day and every activity starts at the beginning of a day and finishes at the end of a day.

[1] Sometimes, the negative lag time is called the lead-time [Callahan et al. 1992; Knutson 2001]. However, there is no standardized terminology for the lead and the lag times in PDM. Ahuja (1976) applies the lag time for the relationships that start from the predecessor's finish time (finish-to-finish and finish-to-start) and the lead time for the relationships that start from the predecessor's start time: start-to-start and start-to-finish.

- The relationship lag time belongs to the predecessor's calendar in both forward and backward passes.
- The relationship lag time indicates the minimum time interval that must exist between two activities in both forward and backward passes.
- If the early time (EST or EFT) of the successor calculated directly from the predecessor is a non-working day, the early time should be postponed to the next available working date in order to satisfy the minimum time interval condition.
- If the late time (LST or LFT) of the predecessor calculated directly from the successor is a non-working day, the late time should be advanced to the previous available working date in order to satisfy the minimum time interval condition.
- The beginning of a day is identical to the end of the previous day. After applying the lag time, the beginning time is applied to find the start time of an activity and the end time is applied to find the finish time of an activity.
- The time difference between the beginning of a day and the beginning of the next day is one day. In the calendar-based schedule, one-day difference is also available on the same date. For example, the time difference between the beginning and the end of the day is one day.
- If an activity has multiple successors with the same relationship type, but with different lag time values, the early time of a successor with a bigger lag time must be greater than or equal to the early time with a smaller lag time. Below equation shows this early time property with different lag times from one predecessor.

$$ET_{n,i} \leq ET_{n+1,j}$$

where, n : lag time (integer),

$ET_{n,i}$: ET(EST or EFT) of activity i with n lag

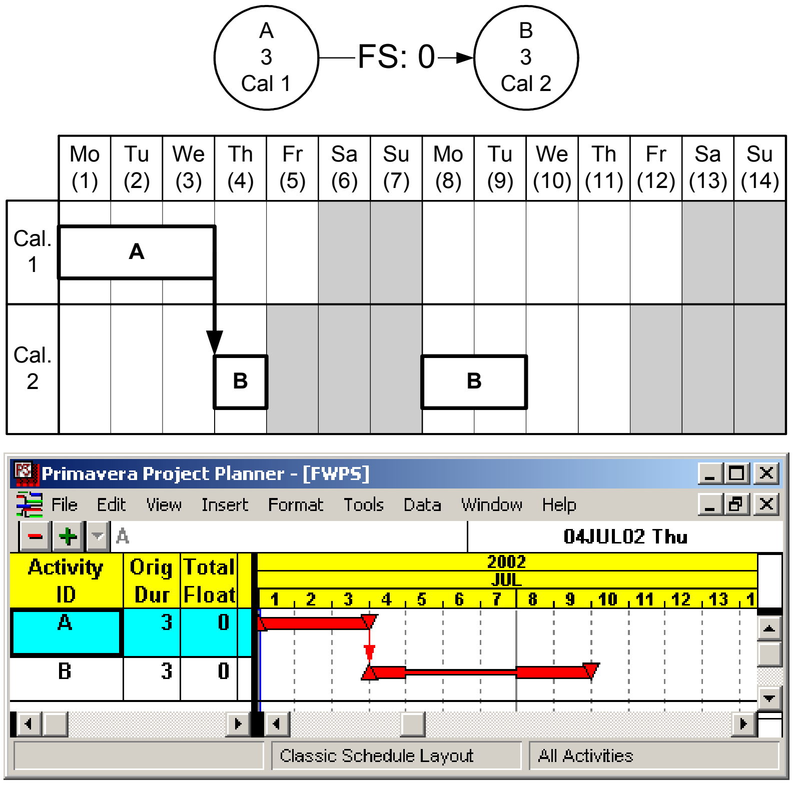
- The process in forward and backward passes in this chapter handles only two calendars, which is sufficient to control a project with three or more calendars. When an activity is involved with more than two calendars through related activities, the method is the same as in the normal CPM. The early time of an activity in the forward pass is the maximum early time among all possible early times calculated from predecessors. Each possible early time is calculated based on two calendars: one for the predecessor and the other for the successor. The same condition happens for the backward pass. The late time of an activity in the backward pass is the minimum late time among all possible late times calculated from successors. Each possible late time is calculated based on two calendars: one for the predecessor and the other for the successor.

4.3 Template Scheduling Chart

In this chapter, each scheduling process will be explained based on a template chart with an example schedule as shown in Figure 4.1. The top part of the chart is a PDM in which each node contains *Activity ID, Duration, and Calendar* ID, and the arrow shows the *relationship type* and *lag*. In this figure, *Activity A* has 3-day duration with *calendar 1* having finish-to-start relationship with zero lag to *Activity B*, which has 3-day duration with *calendar 2*. The middle part of the chart is a fenced bar chart schedule with two calendars. Each column (day) has a continuous work period and the day of the week. The gray boxes indicate non-working days. The bottom part of the figure is a P3 output of the schedule.

In Figure 4.1*, Activity A* is scheduled on *calendar 1* row starting at the beginning of *day 1* and completing at the end of *day 3*. With zero lag of finish-to-start relationship with *Activity A*, *Activity B* is scheduled on *calendar 2* row starting at the beginning of *day 4* and completing at the end of *day 9*. *Activity B* is interrupted because of three days of non-working days.

In the following sections, forward and backward passes are explored for each relationship type with negative, zero, and positive lags. Some procedures are similar, but each process is described in detail to improve clarification. The procedures mainly focus on non-working days that require adjustment to find a working day. Each process provides either start or finish time of the activity depending on the relationship type. The other time can be calculated simply applying the duration of the activity.


A 3 Cal 1
FS: 0
B 3 Cal 2
Mo (1) Tu (2) We (3) Th (4) Fr (5) Sa (6) Su (7) Mo (8) Tu (9) We (10) Th (11) Fr (12) Sa (13) Su (14)
Cal. 1
A
Cal. 2
B
B
Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
04JUL02 Thu
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12 13 1
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float
A	3	0
B	3	0

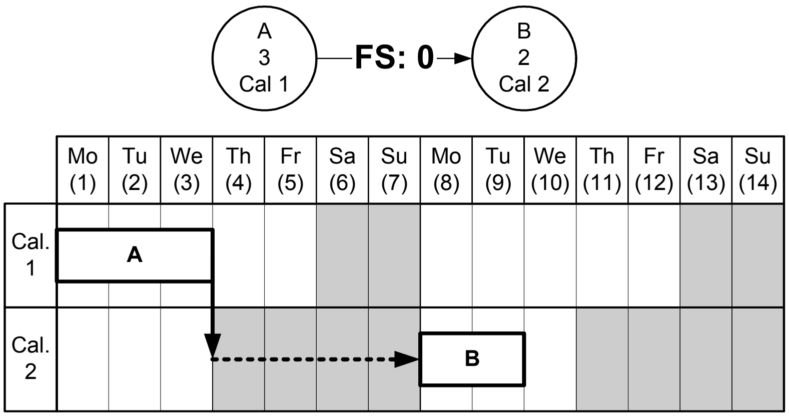
Figure 4.1 Template Scheduling Chart

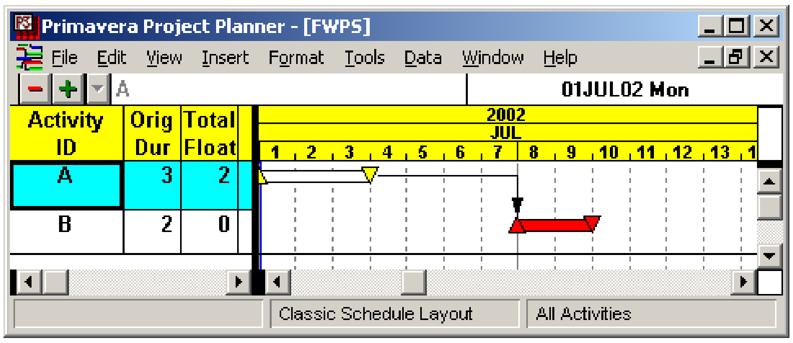
4.4 Forward Pass

4.4.1 Finish-to-Start (FS)

Zero Lag

With a zero lag of FS, the successor activity can start immediately after or later the predecessor activity finishes, so that the earliest possible start time of the successor is the next day of the predecessor's EFT. For example, as shown in Figure 4.2, *Activity A* finishes at the end of day-3, thus its successors with FS:0 can start at the beginning of day-4 or later. Since *Activity B* belongs to *calendar 2*, it cannot start on day-4. *Activity B* can start at the beginning of day-8, which is the next available working day from the non-working earliest possible start time. The P3 output shows an identical result.





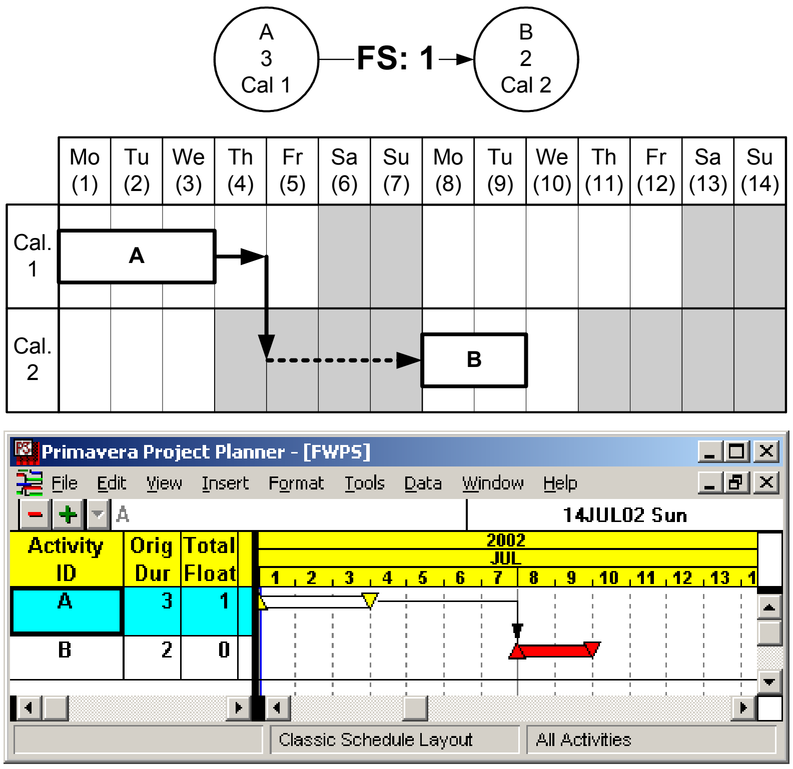
Activity ID	Orig Dur	Total Float
A	3	2
B	2	0

Figure 4.2 FP: Finish-to-Start with zero lag

Positive Lag

With a positive lag of FS, the successor can start at the lag time after or later the predecessor finishes, so that the earliest possible start time of the successor is the next day of the predecessor's EFT plus the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.3, *Activity A* finishes at the end of day-3, thus its successors with FS:1 can start at the beginning of day-5 or later. Since *Activity B* belongs to *calendar 2*, it cannot start on day-5. *Activity B* can start at the beginning of day-8, which is the next available working day from the non-working earliest possible start time.

If the lag is 3-day, then it will be counted on *calendar 1* making *Activity B* available to start at the beginning of day-9. Since day-9 is a working day in *calendar 2*, *Activity B* can start at that time.

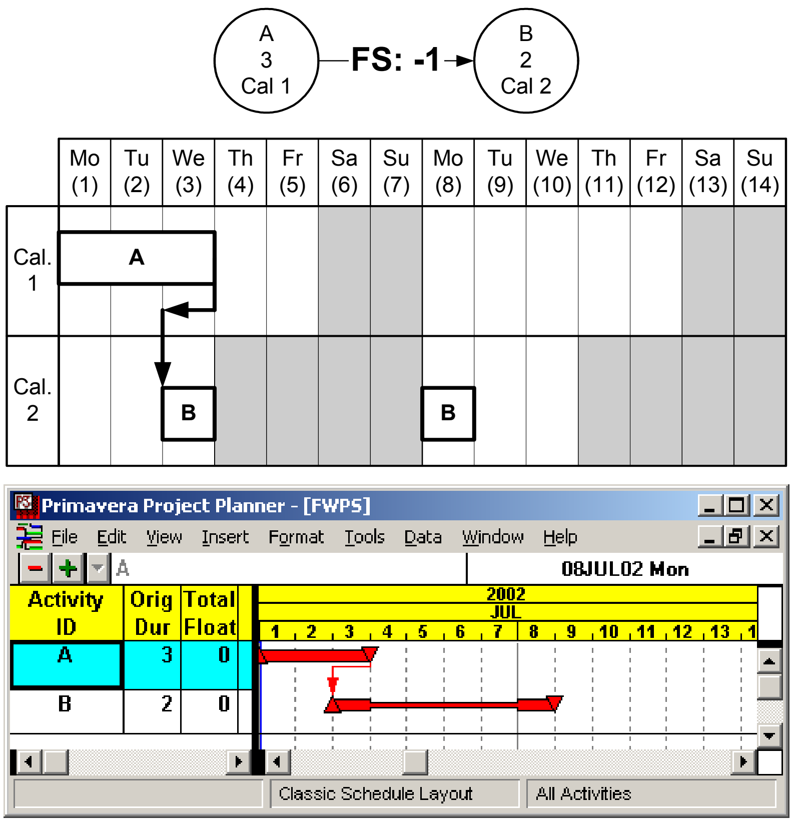


Activity ID	Orig Dur	Total Float
A	3	1
B	2	0

Figure 4.3 FP: Finish-to-Start with positive lag

Negative Lag

With a negative lag of FS, the successor can start at the lag time before or later the predecessor finishes, so that the earliest possible start time of the successor is the next day of the predecessor's EFT less the amount of the lag time counted by working days from the predecessor's calendar. In other words, the successor can start when the remaining duration of the predecessor from the predecessor's calendar is at most the lag time. For example, as shown in Figure 4.4, *Activity A* finishes at the end of day-3, thus its successors with FS:-1 can start at the beginning of day-3 or later. Since day-3 is a working day in calendar 2, *Activity B* can start at the beginning of day-3.



A 3 Cal 1
FS: -1
B 2 Cal 2
Mo (1) Tu (2) We (3) Th (4) Fr (5) Sa (6) Su (7) Mo (8) Tu (9) We (10) Th (11) Fr (12) Sa (13) Su (14)
Cal. 1
Cal. 2
A
B
B
Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
08JUL02 Mon
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12 13 1
Classic Schedule Layout
All Activities

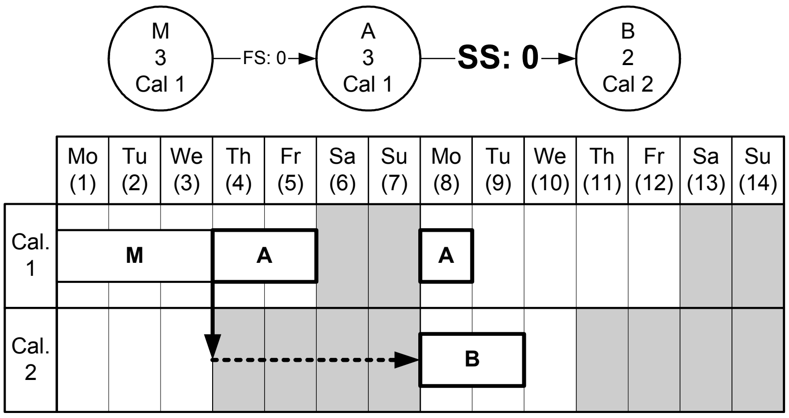
Activity ID	Orig Dur	Total Float
A	3	0
B	2	0

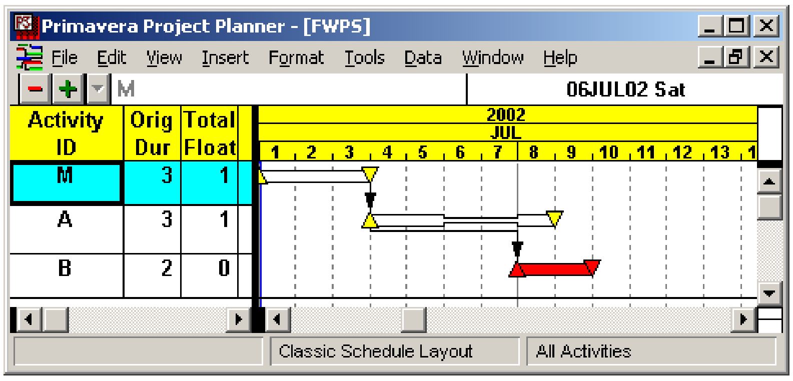
Figure 4.4 FP: Finish-to-Start with negative lag

4.4.2 Start-to-Start (SS)

<u>Zero Lag</u>

With a zero lag of SS, the successor activity can start at the same time or later the predecessor activity starts, so that the earliest possible start time of the successor is the same day of the predecessor's EST. For example, as shown in Figure 4.5, *Activity A* starts at the beginning of day-4, thus its successors with SS:0 can start at the beginning of day-4 or later. Since *Activity B* belongs to *calendar 2*, it cannot start on day-4. *Activity B* can start at the beginning of day-8, which is the next available working day from the non-working earliest possible start time.





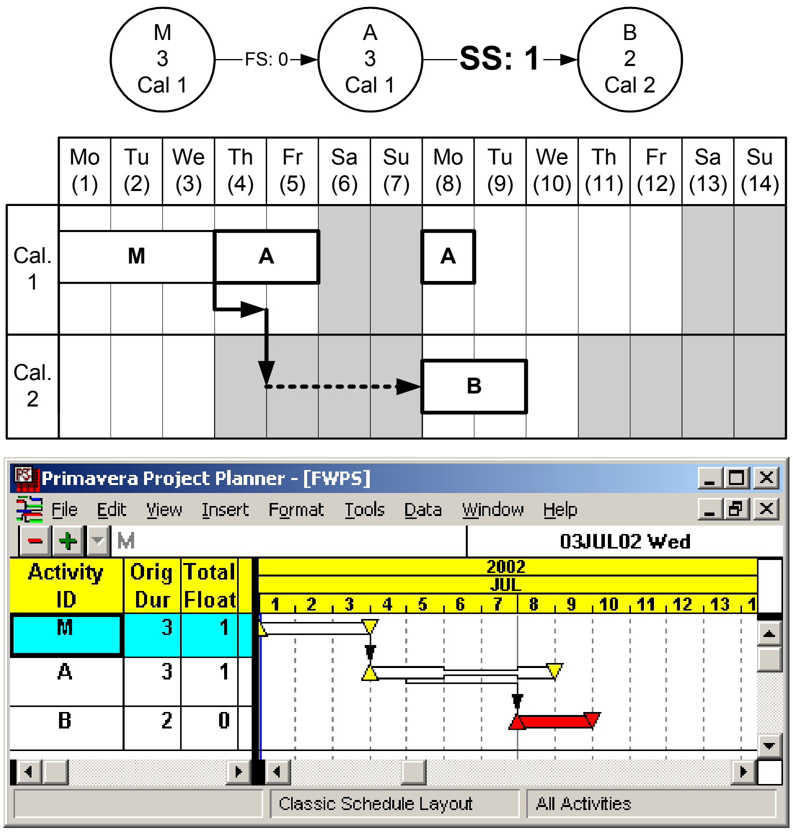
Activity ID	Orig Dur	Total Float
M	3	1
A	3	1
B	2	0

Figure 4.5 FP: Start-to-Start with zero lag

Positive Lag

With a positive lag of SS, the successor can start at the lag time after or later the predecessor starts, so that the earliest possible start time of the successor is the day of the predecessor's EST plus the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.6, *Activity A* starts at the beginning of day-4, thus its successors with SS:1 can start at the beginning of day-5 or later. Since *Activity B* belongs to *calendar 2*, it cannot start on day-5. *Activity B* can start at the beginning of day-8, which is the next available working date from the non-working earliest possible start time.

If the lag is 3-day, then it will be counted on *calendar 1*, making *Activity B* available to start at the beginning of day-9. Since day-9 is a working day in *calendar 2*, *Activity B* can start at that time.



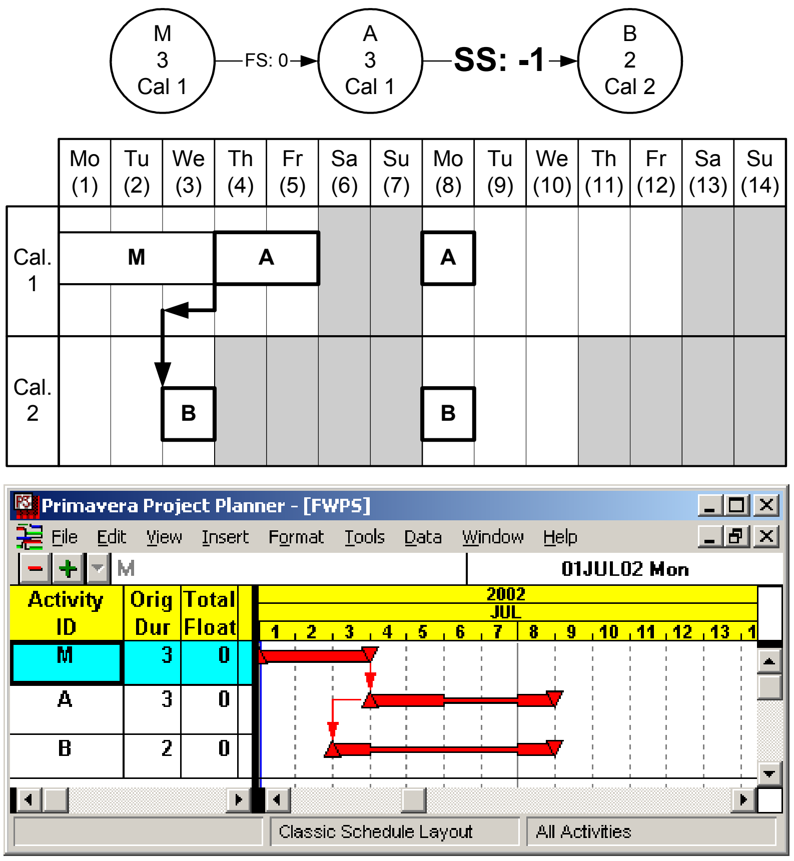
M 3 Cal 1
FS: 0
A 3 Cal 1
SS: 1
B 2 Cal 2
Mo (1) Tu (2) We (3) Th (4) Fr (5) Sa (6) Su (7) Mo (8) Tu (9) We (10) Th (11) Fr (12) Sa (13) Su (14)
Cal. 1
Cal. 2
M
A
A
B
Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
03JUL02 Wed
Activity ID
Orig Dur
Total Float
2002
JUL
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float
M	3	1
A	3	1
B	2	0

Figure 4.6 FP: Start-to-Start with positive lag

Negative Lag

With a negative lag of SS, the successor can start at the lag time before or later the predecessor starts, so that the earliest possible start time of the successor is the day of the predecessor's EST less the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.7, *Activity A* starts at the beginning of day-4, thus its successors with SS:-1 can start at the beginning of day-3 or later. Since day-3 is a working day in calendar 2, *Activity B* can start at the beginning of day-3.



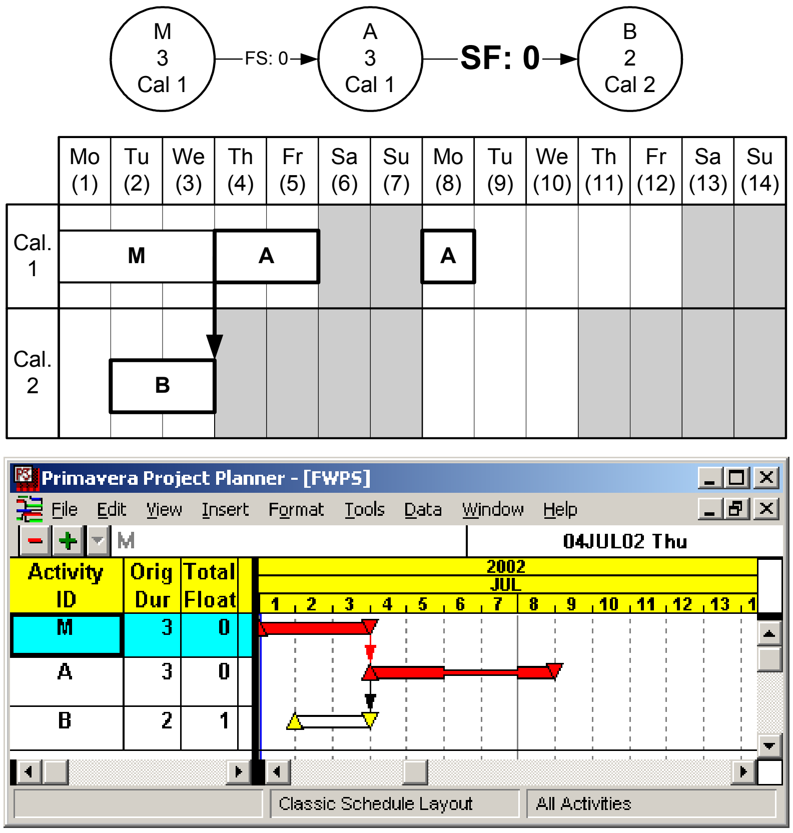
Activity ID	Orig Dur	Total Float
M	3	0
A	3	0
B	2	0

Figure 4.7 FP: Start-to-Start with negative lag

4.4.3 Start-to-Finish (SF)

Zero Lag

With a zero lag of SF, the successor activity can finish at the same time or later the predecessor activity starts, so that the earliest possible finish time of the successor is the previous day of the predecessor's EST. For example, as shown in Figure 4.8, *Activity A* starts at the beginning of day-4, thus its successors with SF:0 can finish at the end of day-3 or later. Since day-3 is a working day in *calendar 2*, *Activity B* can finish at the end of day-3.

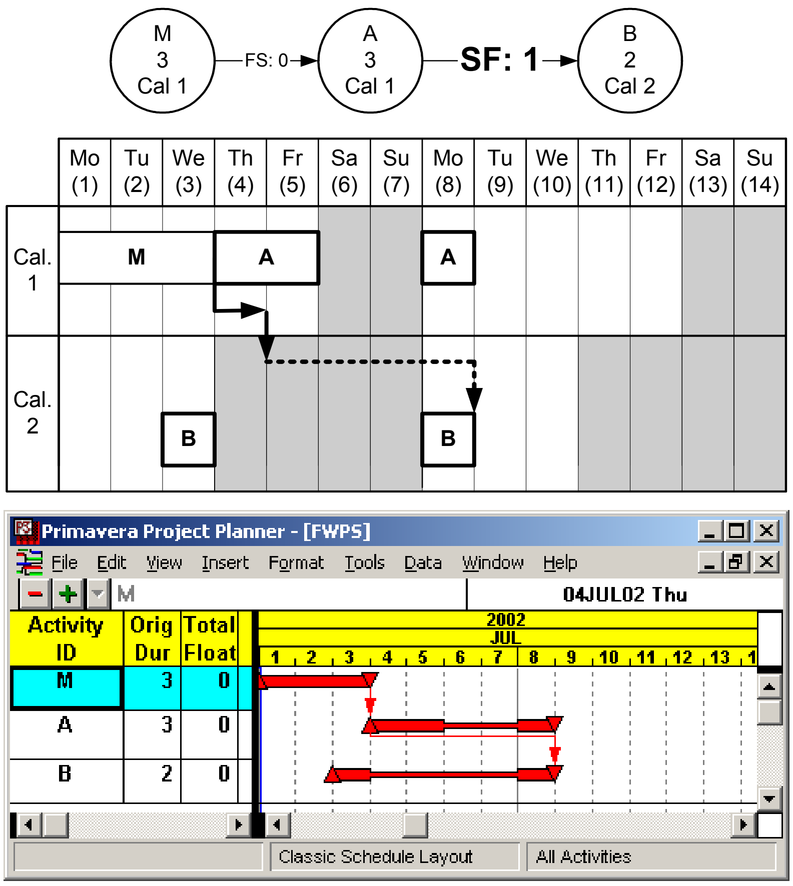


Activity ID	Orig Dur	Total Float
M	3	0
A	3	0
B	2	1

Figure 4.8 FP: Start-to-Finish with zero lag

Positive Lag

With a positive lag of SF, the successor can finish at the lag time after or later the predecessor starts, so that the earliest possible finish time of the successor is the previous day of the predecessor's EST plus the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.9, *Activity A* starts at the beginning of day-4, thus its successors with SF:1 can finish at the end of day-4 or later. Since *Activity B* belongs to *calendar 2*, it cannot finish on day-4. *Activity B* can finish at the end of day-8, which is the next available working day from the non-working earliest possible finish time.

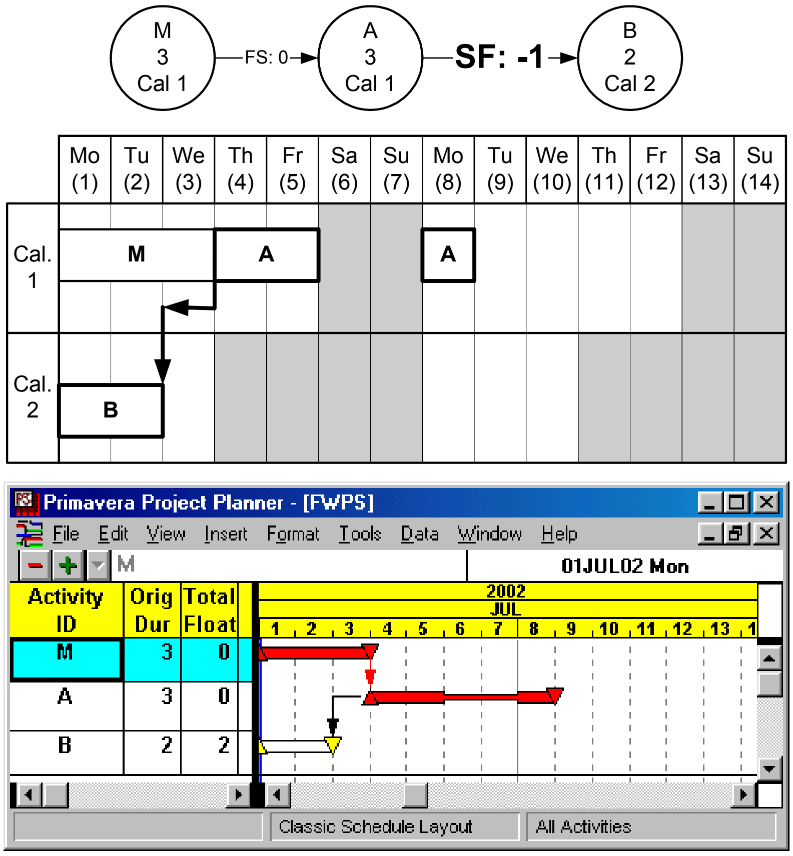

M 3 Cal 1
FS: 0
A 3 Cal 1
SF: 1
B 2 Cal 2
Mo (1) Tu (2) We (3) Th (4) Fr (5) Sa (6) Su (7) Mo (8) Tu (9) We (10) Th (11) Fr (12) Sa (13) Su (14)
Cal. 1
M
A
A
Cal. 2
B
B
Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
M
04JUL02 Thu
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12 13
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float
M	3	0
A	3	0
B	2	0

Figure 4.9 FP: Start-to-Finish with positive lag

Negative Lag

With a negative lag of SF, the successor can finish at the lag time before or later the predecessor starts, so that the earliest possible finish time of the successor is the previous day of the predecessor's EST less the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.10, *Activity A* starts at the beginning of day-4, thus its successors with SF:-1 can finish at the end of day-2 or later. Since day-2 is a working day in *calendar 2*, *Activity B* can finish at the end of day-2.



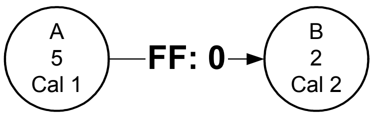
Activity ID	Orig Dur	Total Float
M	3	0
A	3	0
B	2	2

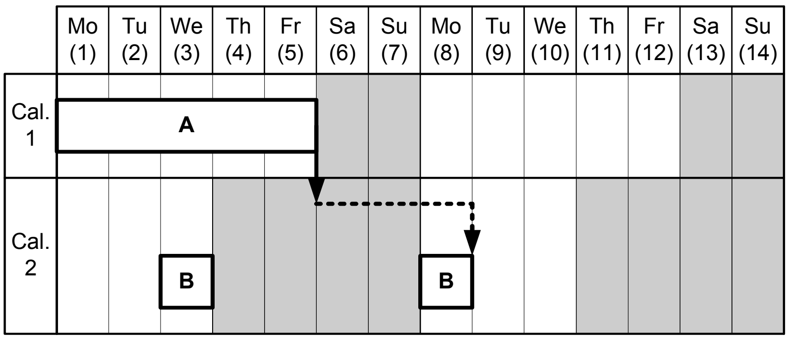
Figure 4.10 FP: Start-to-Finish with negative lag

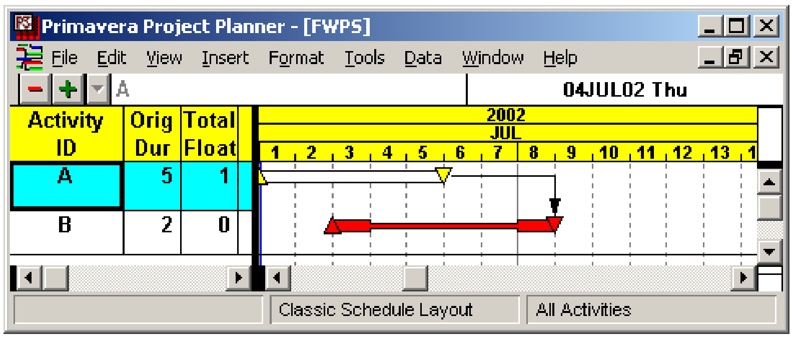
4.4.4 Finish-to-Finish (FF)

<u>Zero Lag</u>

With a zero lag of FF, the successor activity can finish at the same time or later the predecessor activity finishes, so that the earliest possible finish time of the successor is the same day of the predecessor's EFT. For example, as shown in Figure 4.11, *Activity A* finishes at the end of day-5, thus its successors with FF:0 can finish at the end of day-5 or later. Since *Activity B* belongs to *calendar 2*, it cannot finish on day-5. *Activity B* can finish at the end of day-8, which is the next available working day from the non-working earliest possible finish time.





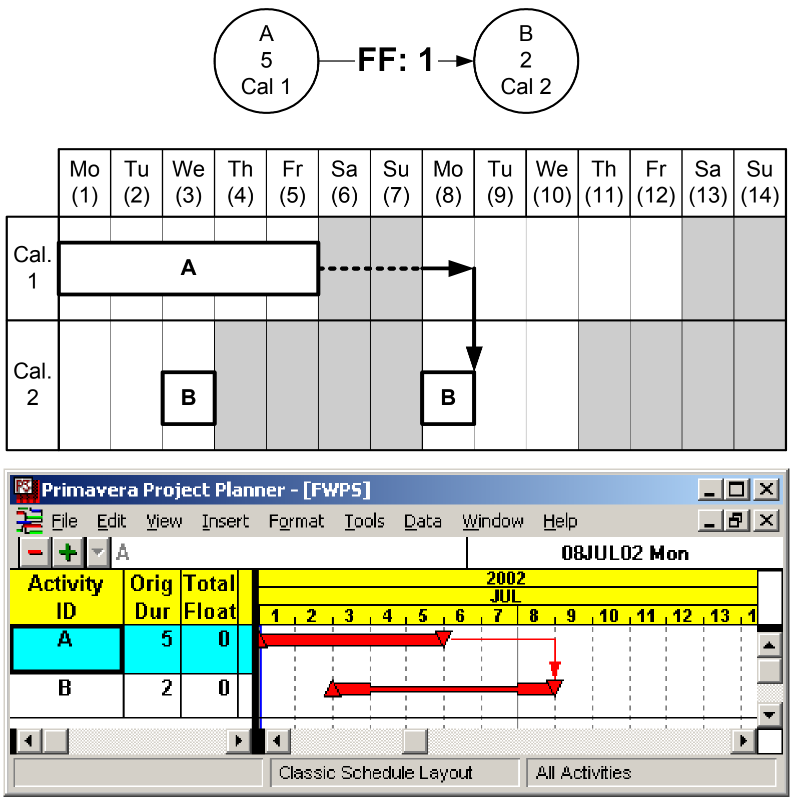


Activity ID	Orig Dur	Total Float
A	5	1
B	2	0

Figure 4.11 FP: Finish-to-Finish with zero lag

Positive Lag

With a positive lag of FF, the successor can finish at the lag time after or later the predecessor finishes, so that the earliest possible finish time of the successor is the day of the predecessor's EFT plus the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.12, *Activity A* finishes at the end of day-5, thus its successors with FF:1 can finish at the end of day-8 or later. Since day-8 is a working day in calendar 2, *Activity B* can finish at the end of day-8.



A 5 Cal 1
FF: 1
B 2 Cal 2
Mo (1) Tu (2) We (3) Th (4) Fr (5) Sa (6) Su (7) Mo (8) Tu (9) We (10) Th (11) Fr (12) Sa (13) Su (14)
Cal. 1
A
Cal. 2
B
B
Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
08JUL02 Mon
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12 13 1
Classic Schedule Layout
All Activities

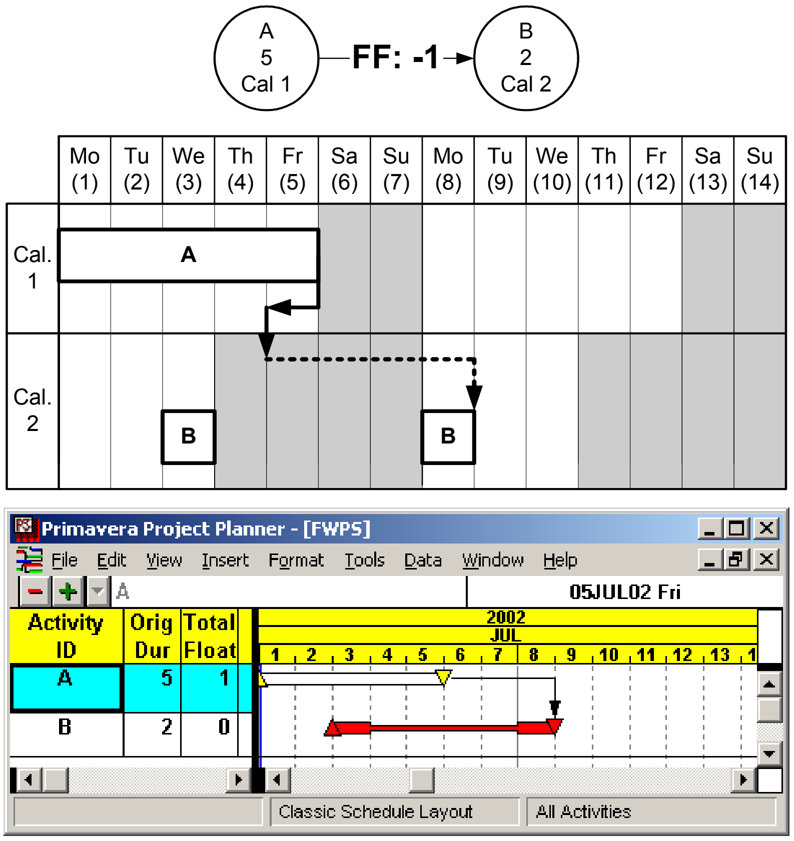
Activity ID	Orig Dur	Total Float
A	5	0
B	2	0

Figure 4.12 FP: Finish-to-Finish with positive lag

Negative Lag

With a negative lag of FF, the successor can finish at the lag time before or later the predecessor finishes, so that the earliest possible finish time of the successor is the day of the predecessor's EFT less the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.13, *Activity A* finishes at the end of day-5, thus its successors with FF:-1 can finish at the end of day-4 or later. Since *Activity B* belongs to *calendar 2*, it cannot finish on day-4. *Activity B* can finish at the end of day-8, which is the next available working day from the non-working earliest possible finish time.

If the lag is (-2)-day, then it will be counted on *calendar 1* making its successors can finish at the end of day-3 or later. Since day-3 is a working day in calendar 2, *Activity B* can finish at the end of day-3.



Activity ID	Orig Dur	Total Float
A	5	1
B	2	0

Figure 4.13 FP: Finish-to-Finish with negative lag

4.5 Backward Pass

The backward pass begins after the forward pass is completed for all activities. In the backward pass, the late times of each activity are calculated based on the late times of the successors. As mentioned in the basic rules, the relationship lag time is also the minimum time interval that must exist between two activities in the backward passes. Scheduling with multiple calendars is similar to scheduling with continuous working days, but it has additional features to handle non-working days. For the convenience, in this section, fence bar chart and P3 output show late times, and the arrows in the fenced bar chart show backward pass flow, not the original activity workflow. Besides, early times are placed on top of each node of the PDM network.

4.5.1 Finish-to-Start (FS)

<u>Zero Lag</u>

With a zero lag of FS, the successor activity can start immediately after or later the predecessor activity finishes whether the schedule is based on early times or late times. With this property, the latest possible finish time of the predecessor is the previous day of the successor's LST. For example, as shown in Figure 4.14, the LST of *Activity B* is the beginning day-5, thus its predecessors with FS:0 can finish at the end of day-4 or <u>earlier</u>. Since *Activity B* belongs to *calendar 2*, it cannot finish on day-4. *Activity B* can finish at the end of day-3, which is the first available working day from the non-working latest possible finish time.

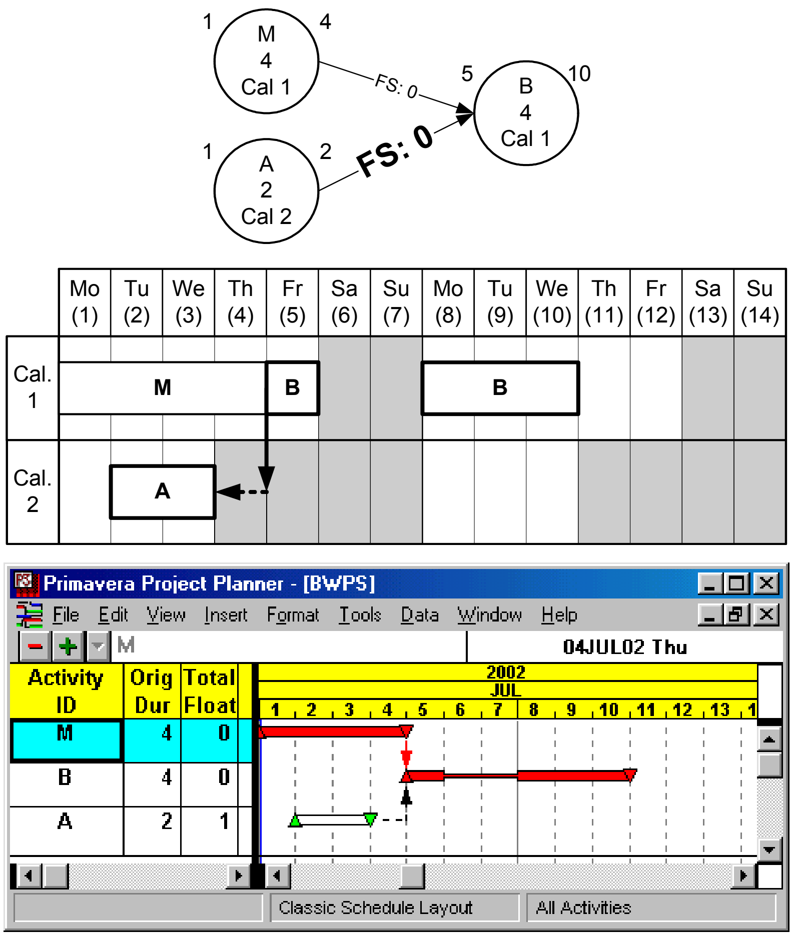
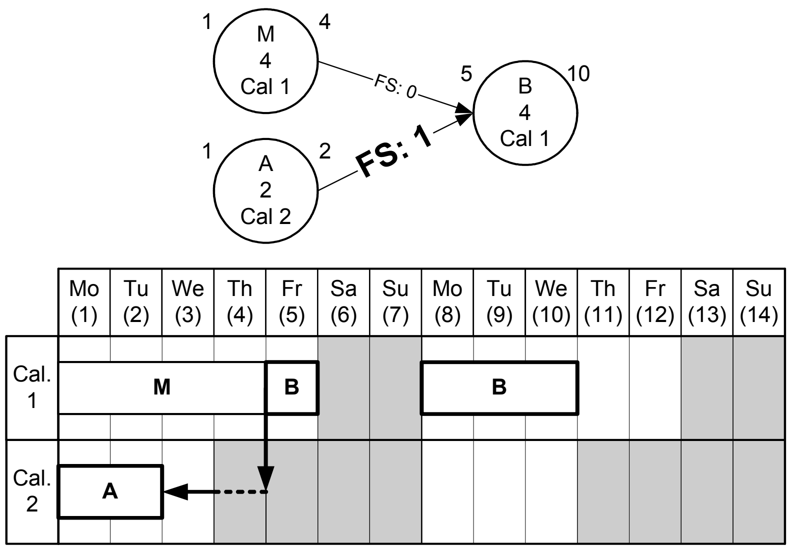
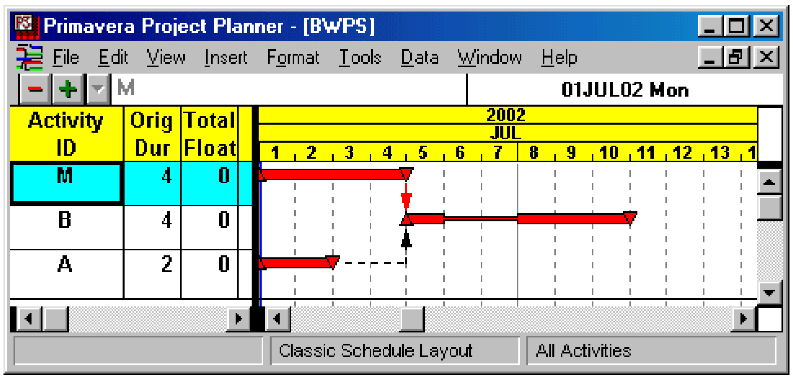


Figure 4.14 BP: Finish-to-Start with zero lag

Positive Lag

With a positive lag of FS, the successor can start at the lag time after or later the predecessor finishes. With this property, the latest possible finish time of the predecessor is the previous day of the successor's LST less the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.15, the LST of *Activity B* is the beginning day-5, thus its predecessors with FS:1 on *calendar 2* can finish at the end of day-2 or earlier. Since day-2 is a working day in *calendar 2*, *Activity A* can finish at the end of day-2.




Primavera Project Planner - [BWPS]
File Edit View Insert Format Tools Data Window Help
M
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12 13 1

01JUL02 Mon

Activity ID	Orig Dur	Total Float
M	4	0
B	4	0
A	2	0

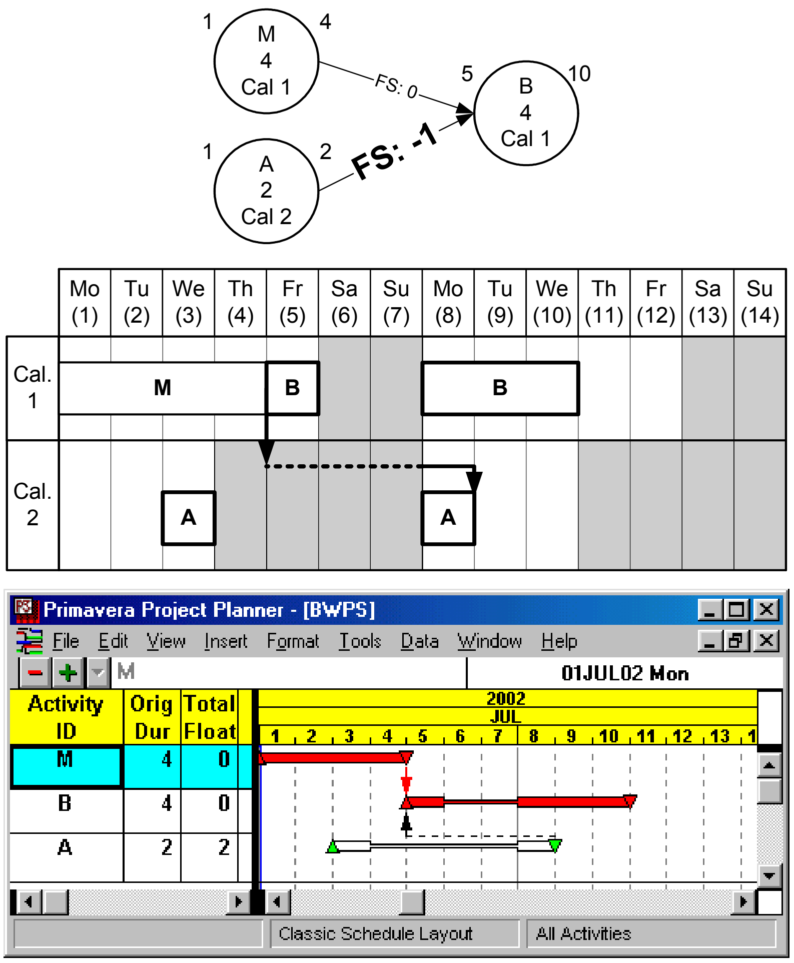
Classic Schedule Layout

All Activities

Figure 4.15 BP: Finish-to-Start with positive lag

Negative Lag

With a negative lag of FS, the successor can start at the lag time before or later the predecessor finishes. With this property, the latest possible finish time of the predecessor is the previous day of the successor's LST plus the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.16, the LST of *Activity B* is the beginning day-5, thus its predecessors with FS:-1 on *calendar 2* can finish at the end of day-8 or earlier. Since day-8 is a working day in *calendar 2*, *Activity B* can finish at the end of day-8.



1 M 4 Cal 1 4
FS: 0
5 B 4 Cal 1 10
1 A 2 Cal 2 2
FS: -1
Mo (1) Tu (2) We (3) Th (4) Fr (5) Sa (6) Su (7) Mo (8) Tu (9) We (10) Th (11) Fr (12) Sa (13) Su (14)
Cal. 1
M
B
B
Cal. 2
A
A
Primavera Project Planner - [BWPS]
File Edit View Insert Format Tools Data Window Help
01JUL02 Mon
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12 13
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float
M	4	0
B	4	0
A	2	2

Figure 4.16 BP: Finish-to-Start with negative lag

4.5.2 Start-to-Start (SS)

Zero Lag

With a zero lag of SS, the successor activity can start at the same time or later the predecessor activity starts whether the schedule is based on early times or late times. With this property, the latest possible start time of the predecessor is the same time of the successor's LST. For example, as shown in Figure 4.17, *Activity B* starts at the beginning of day-5, thus its predecessors with SS:0 can start at the beginning of day-5 or earlier. Since *Activity A* belongs to *calendar 2*, it cannot start on day-5. *Activity A* can start at the beginning of day-3, which is the first available working day from the non-working latest possible start time.

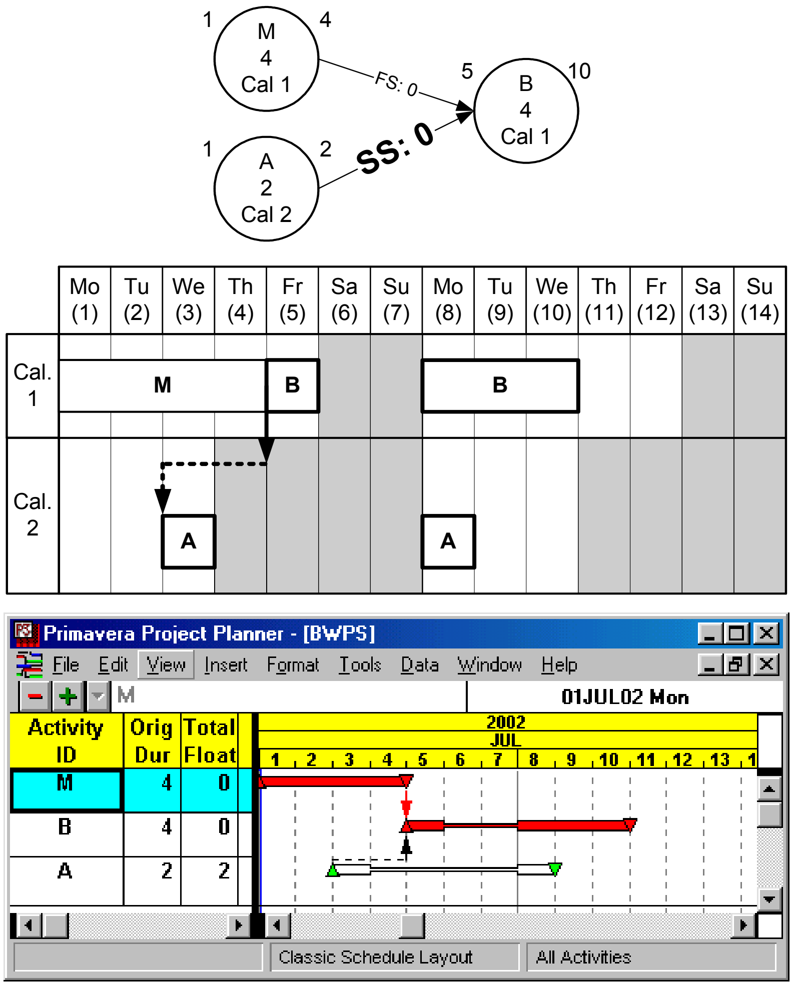
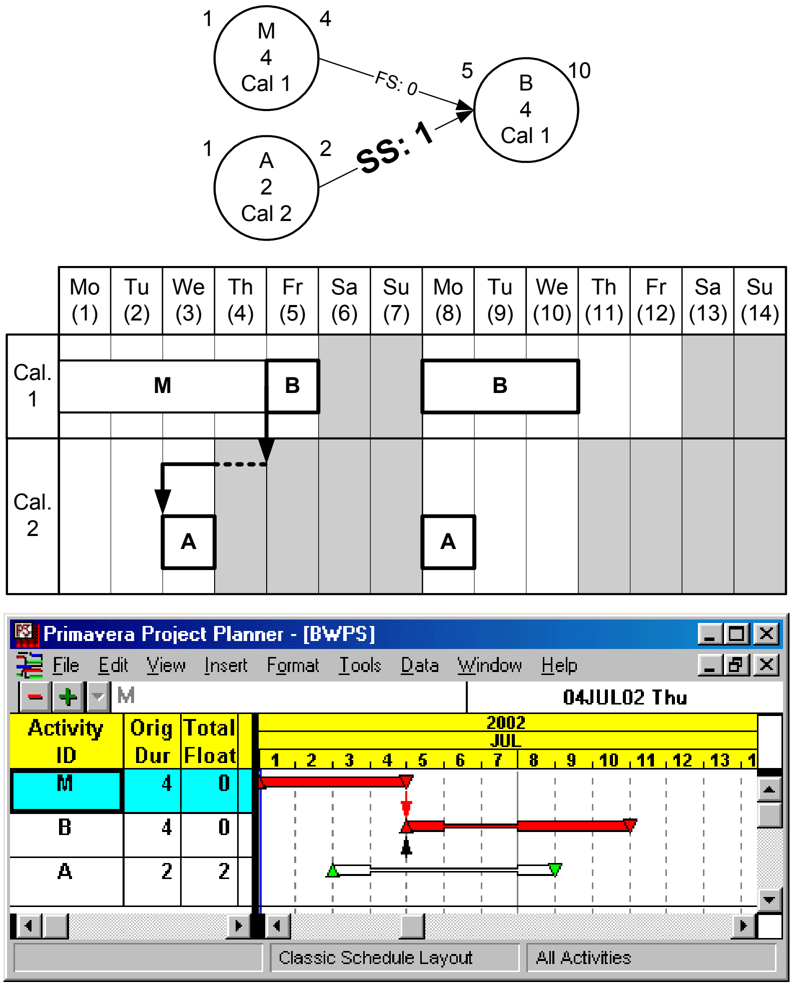


Figure 4.17 BP: Start-to-Start with zero lag

Positive Lag

With a positive lag of SS, the successor can start at the lag time after or later the predecessor starts. With this property, the latest possible start time of the predecessor is the day of the successor's LST less the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.18, the LST of *Activity B* is the beginning day-5, thus its predecessors with SS:1 on *calendar 2* can start at the beginning of day-3 or earlier. Since day-3 is a working day on *calendar 2*, *Activity A* can start at that time.

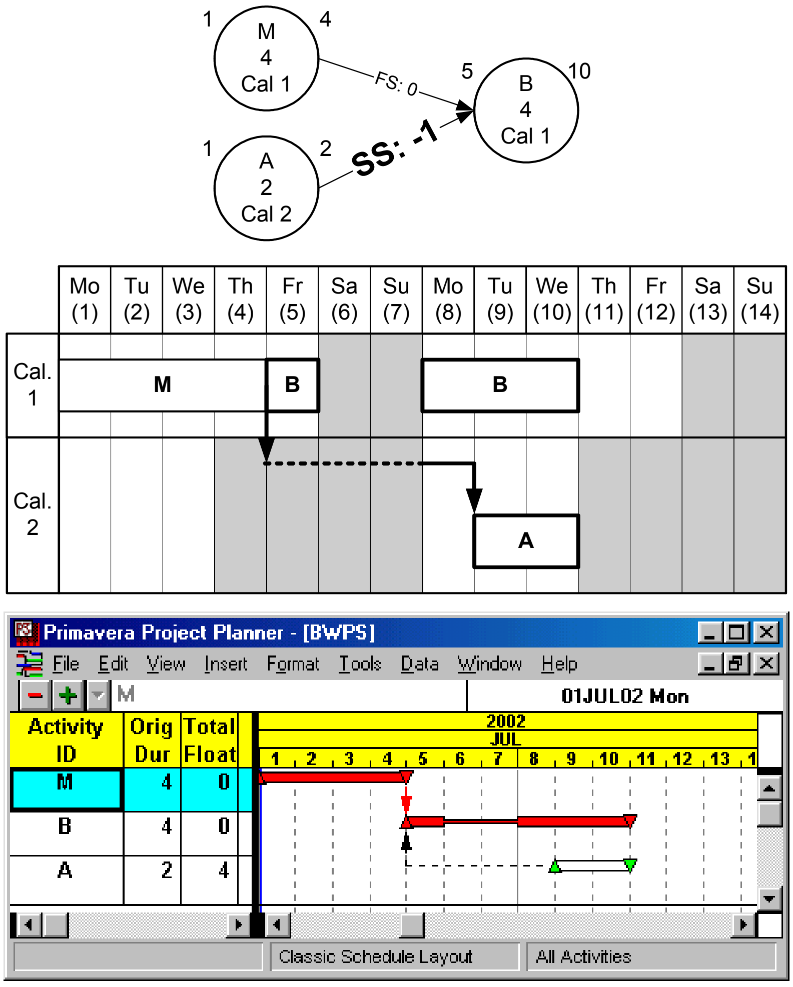


Activity ID	Orig Dur	Total Float
M	4	0
B	4	0
A	2	2

Figure 4.18 BP: Start-to-Start with positive lag

Negative Lag

With a negative lag of SS, the successor can start at the lag time before or later the predecessor starts. With this property, the latest possible start time of the predecessor is the day of the successor's LST plus the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.19, the LST of *Activity B* is the beginning day-5, thus its predecessors with SS:-1 on *calendar 2* can start at the beginning of day-9 or earlier. Since day-9 is a working day in *calendar 2*, *Activity A* can start at the beginning of day-9.



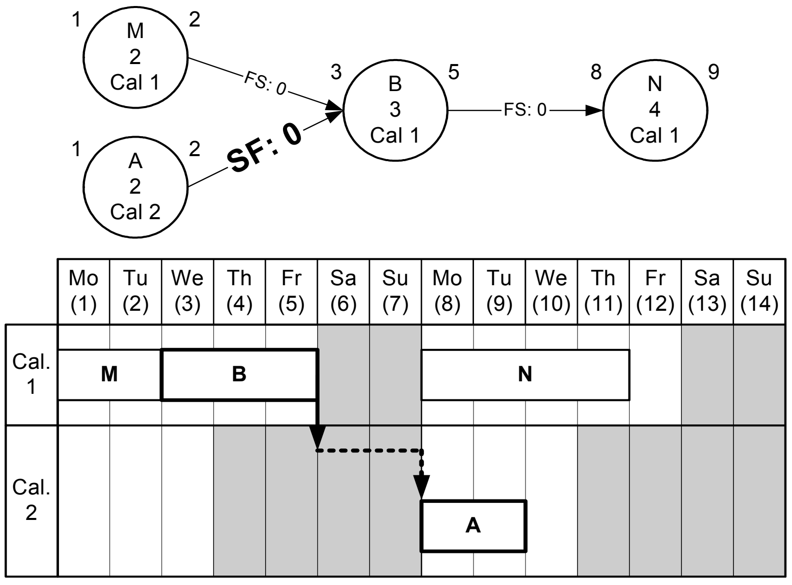
Activity ID	Orig Dur	Total Float
M	4	0
B	4	0
A	2	4

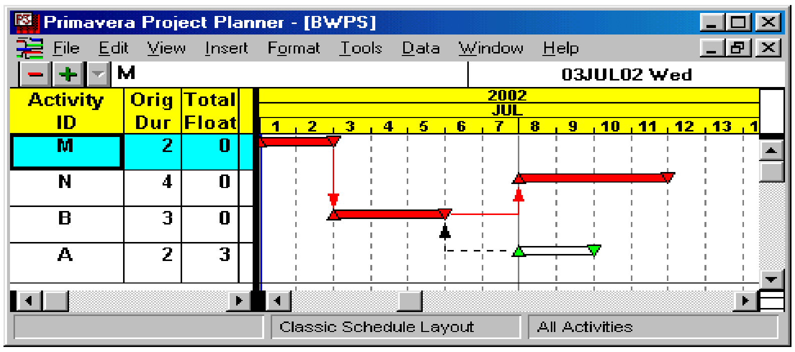
Figure 4.19 BP: Start-to-Start with negative lag

4.5.3 Start-to-Finish (SF)

Zero Lag

With a zero lag of SF, the successor activity can finish on the previous day or later the predecessor activity starts whether the schedule is based on early times or late times. Because the predecessor could start the next day of the successor's finish, special care is required in the backward pass with SF:0 especially when non-working days are involved. The latest possible start time of the predecessor is the next day of the successor's LFT or, if it is a non-working day, the next available working day. For example, as shown in Figure 4.20, *Activity B* finishes at the end of day-5, thus its predecessors with SF:0 can start at the beginning of day-6, if it is a working day. However, since *Activity A* belongs to *calendar 2* and day-6 is a non-working day, *Activity A* can start at the beginning of day-8.



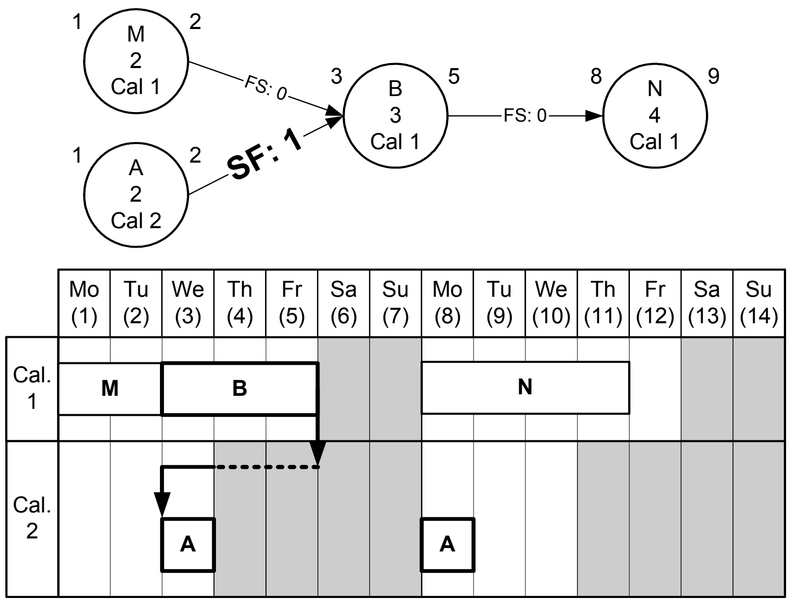


Activity ID	Orig Dur	Total Float
M	2	0
N	4	0
B	3	0
A	2	3

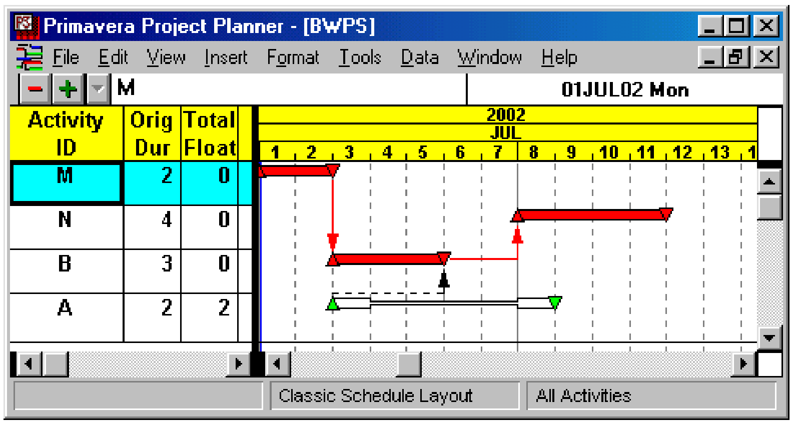
Figure 4.20 BP: Start-to-Finish with zero lag

Positive Lag

With a positive lag of SF, the successor can finish at the lag time after or later the predecessor starts. With this property, the latest possible start time of the predecessor is the next day of the successor's LFT less the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.21, the LFT of *Activity B* is the end of day-5, thus its predecessors with SF:1 on *calendar 2* can start at the beginning of day-3. Since day-3 is a working day on *calendar 2*, *Activity A* can start at that time.



1 M 2 Cal 1 2
1 A 2 Cal 2 2
3 B 3 Cal 1 5
8 N 4 Cal 1 9
FS: 0
SF: 1
FS: 0
Mo (1) Tu (2) We (3) Th (4) Fr (5) Sa (6) Su (7) Mo (8) Tu (9) We (10) Th (11) Fr (12) Sa (13) Su (14)
Cal. 1
M
B
N
Cal. 2
A
A

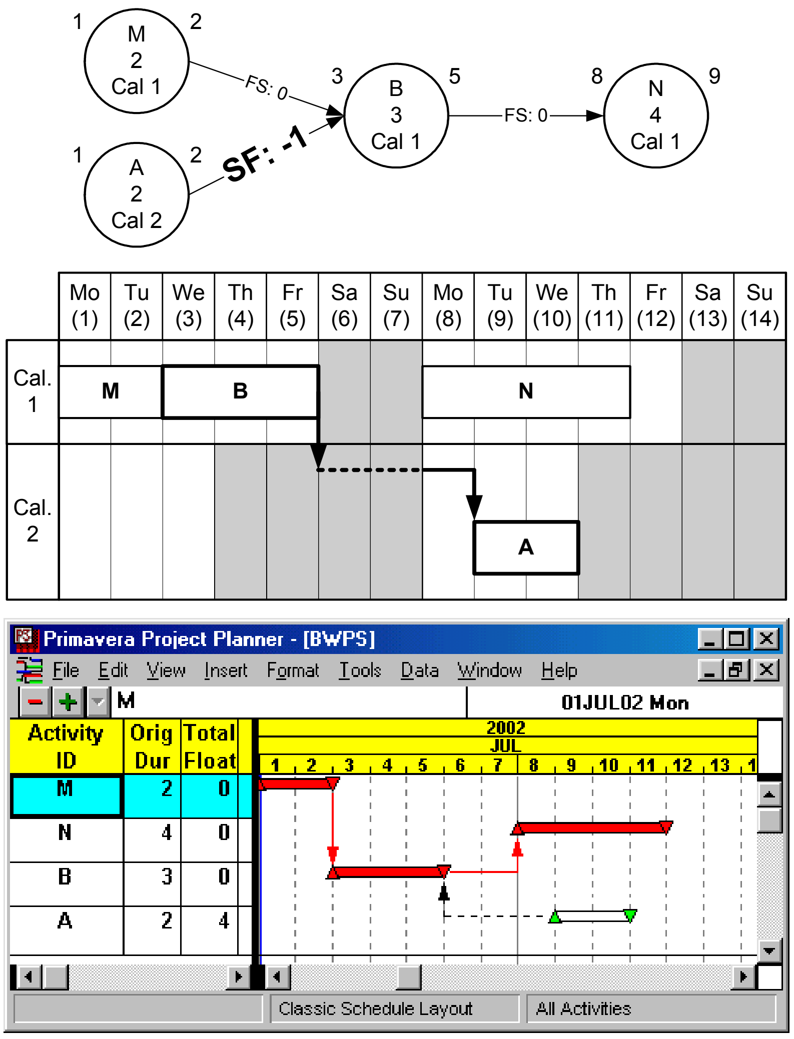


Activity ID	Orig Dur	Total Float
M	2	0
N	4	0
B	3	0
A	2	2

Figure 4.21 BP: Start-to-Finish with positive lag

<u>Negative Lag</u>

With a negative lag of SF, the successor can finish at the lag time before or later the predecessor starts. With this property, the latest possible start time of the predecessor is the next day of the successor's LFT plus the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.22, the LFT of *Activity B* is the end day-5, thus its predecessors with SF:-1 on *calendar 2* can start at the beginning of day-9 or earlier. Since day-9 is a working day in *calendar 2*, *Activity A* can start at this time.



1 M 2 Cal 1 2
FS: 0
3 B 3 Cal 1 5
FS: 0
8 N 4 Cal 1 9
1 A 2 Cal 2 2
SF: -1
Mo (1) Tu (2) We (3) Th (4) Fr (5) Sa (6) Su (7) Mo (8) Tu (9) We (10) Th (11) Fr (12) Sa (13) Su (14)
Cal. 1
M
B
N
Cal. 2
A
Primavera Project Planner - [BWPS]
File Edit View Insert Format Tools Data Window Help
M
01JUL02 Mon
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12 13 1
Classic Schedule Layout
All Activities

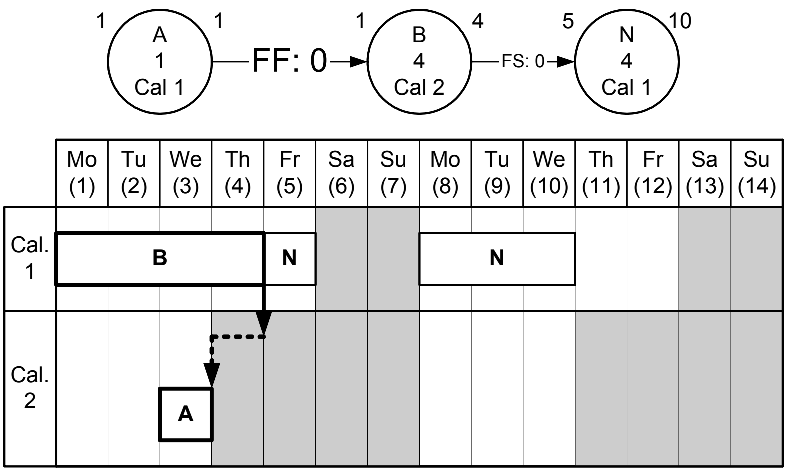
Activity ID	Orig Dur	Total Float
M	2	0
N	4	0
B	3	0
A	2	4

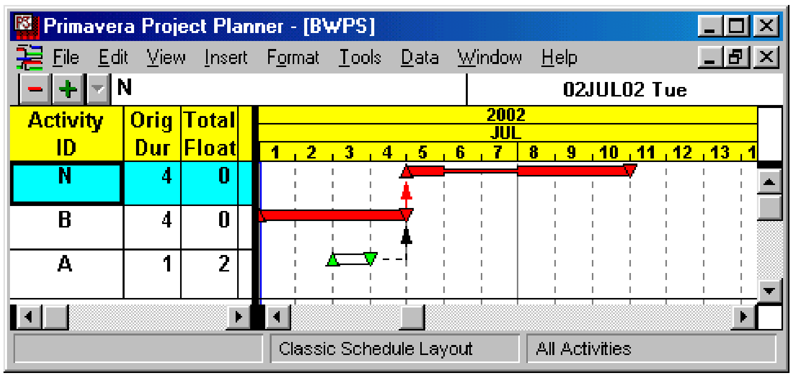
Figure 4.22 BP: Start-to-Finish with negative lag

4.5.4 Finish-to-Finish (FF)

Zero Lag

With a zero lag of FF, the successor activity can finish at the same time or later the predecessor activity finishes whether the schedule is based on early times or late times. With this property, the latest possible finish time of the predecessor is the same time of the successor's LFT. For example, as shown in Figure 4.23, the LFT of *Activity B* is day-4, thus its predecessors with FF:0 can finish on day-4 or earlier. Since *Activity A* belongs to *calendar 2*, it cannot finish on day-4. *Activity A* can finish at the end of day-3, which is the first available working day from the non-working latest possible finish time.



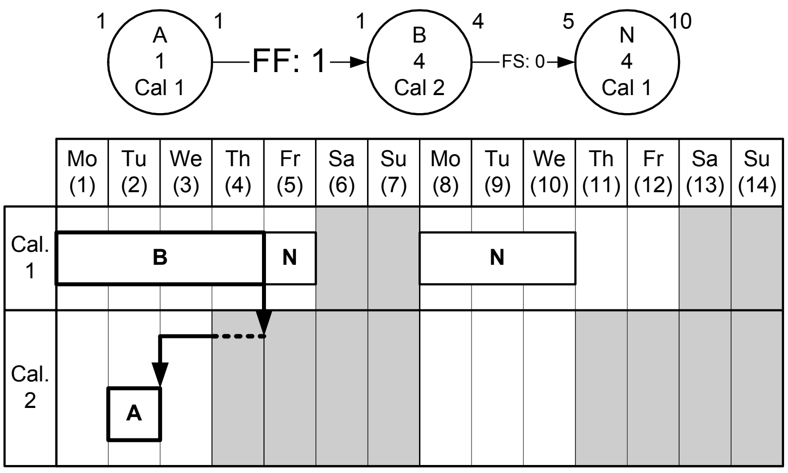

Primavera Project Planner - [BWPS]

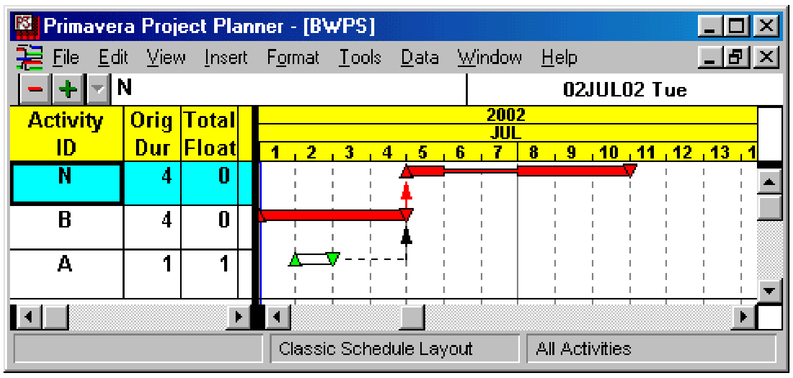
Activity ID	Orig Dur	Total Float
N	4	0
B	4	0
A	1	2

Figure 4.23 BP: Finish-to-Finish with zero lag

Positive Lag

With a positive lag of FF, the successor can finish at the lag time after the predecessor finishes, or later. With this property, the latest possible finish time of the predecessor is the day of the successor's LFT less the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.24, the LFT of *Activity B* is the end of day-4, thus its predecessors with FF:1 on *calendar 2* can finish at the end of day-2 or earlier. Since day-2 is a working day on *calendar 2*, *Activity A* can finish at the end of day-2.





Activity ID	Orig Dur	Total Float
N	4	0
B	4	0
A	1	1

Figure 4.24 BP: Finish-to-Finish with positive lag

<u>Negative Lag</u>

With a negative lag of FF, the successor can finish at the lag time before the predecessor finishes, or later. With this property, the latest possible finish time of the predecessor is the day of the successor's LFT plus the amount of the lag time counted by working days from the predecessor's calendar. For example, as shown in Figure 4.25, the LFT of *Activity B* is the end day-4, thus its predecessors with FF:-1 on *calendar 2* can finish at the end of day-8 or earlier. Since day-8 is a working day in *calendar 2*, *Activity A* can finish at that time.

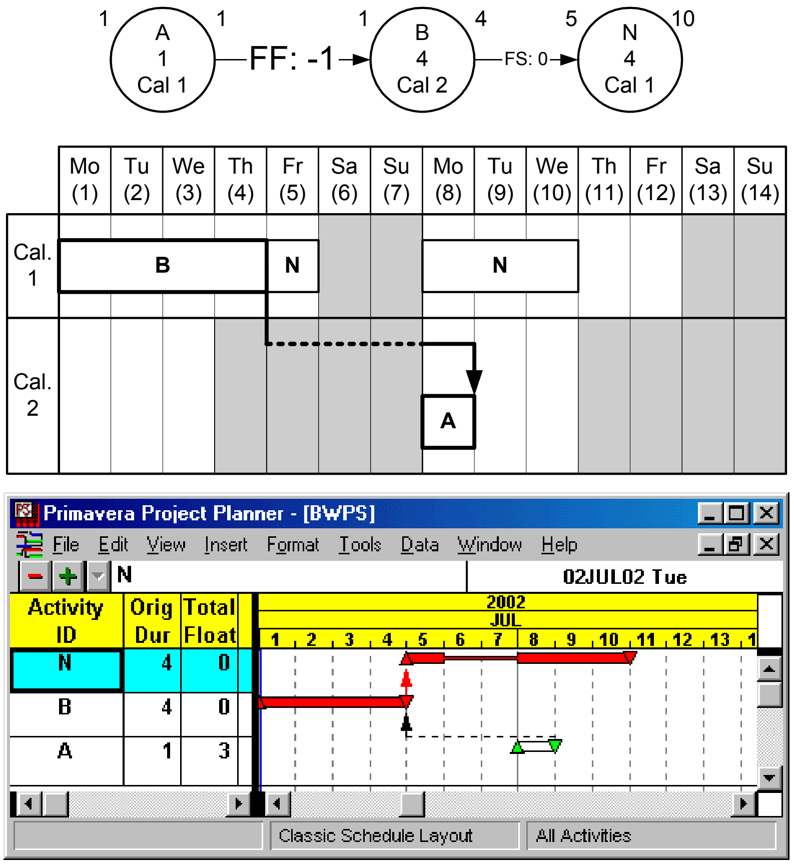


Figure 4.25 BP: Finish-to-Finish with negative lag

4.6 P3 Problems in Forward Pass

4.6.1 Negative Lag in P3

The forward pass provided earlier can be applied to most cases, but special care is required in a certain case. This happens if the lag time is negative in all relationship types and if the earliest possible time of the successor after applying lag time is the beginning of the next day of a non-working day on the predecessor's calendar.

For example, as shown in Figure 4.26, *Activity A* and *B* has a finish-to-start relationship with (-1) lag and, after applying the forward pass provided earlier, the earliest possible start time of the successor is day-8, which is the next day of a non-working day on the predecessor's calendar. There is no doubt on day-8 becoming the EST of *Activity B* if it belongs to the predecessor's calendar or if day-7 is a working day on calendar 2. However, since these conditions are not held in this case, special care is required for the EST of *Activity B*.

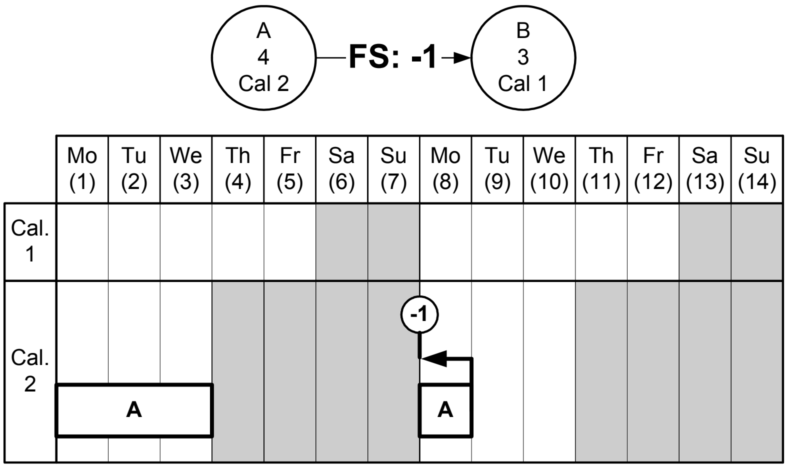


Figure 4.26 FW: Lag Time Counting in FS:(-1)

With FS:(-1) from the predecessor, the successor can start whenever the remaining duration of the predecessor is at most one day. From the schedule of Figure 4.26, while *calendar 2* has only one day (day-8) that satisfies such condition, *calendar 2* has three possible days: day-4, 5, or 8 as shown in Figure 4.27. The remaining duration of the predecessor is equally one day for each of them, since day-4 and day-5 are non-workings for the predecessor. Day-4 is the earliest time among them and can be the EST of *Activity B* satisfying the relationship condition.

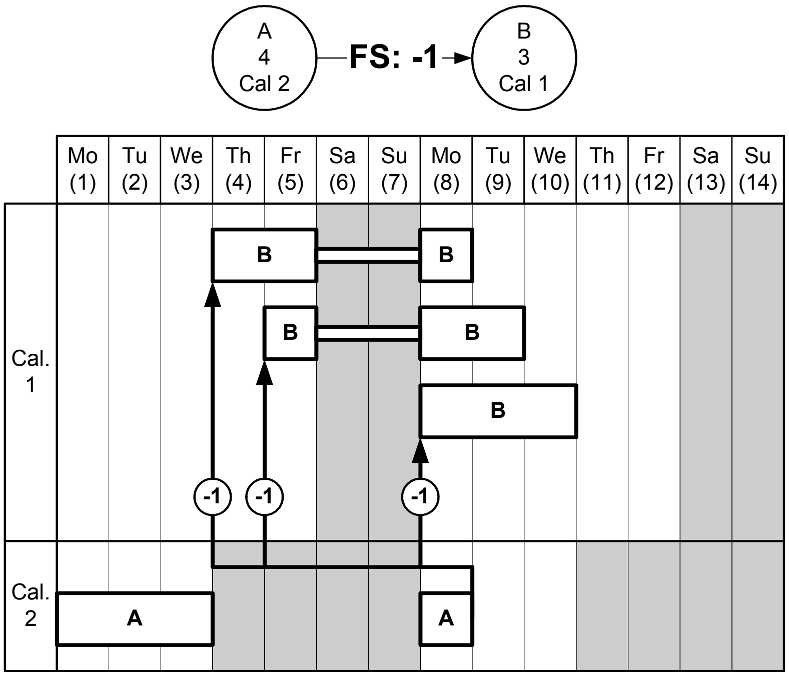
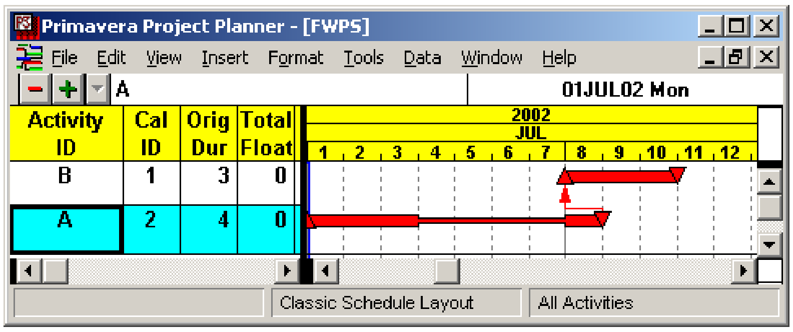


Figure 4.27 FW: Possible Early Start Times with FS:(-1)



Primavera Project Planner - [FWPS]

File Edit View Insert Format Tools Data Window Help

A | 01JUL02 Mon

Activity ID	Cal ID	Orig Dur	Total Float
B	1	3	0
A	2	4	0

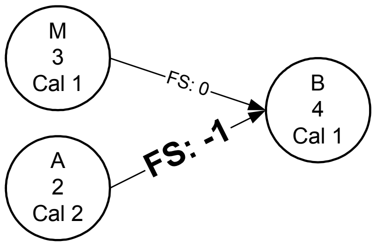
Figure 4.28 FW: P3 Output with FS:(-1)

P3, however, generates a different output as day-8 for the *Activity B*'s EST as shown in Figure 4.28. Presumably, no additional consideration is given to this special case without taking into account possible earlier times hidden by the non-working days of the predecessor's calendar. Although it is true that the P3 schedule keeps the minimum interval condition between two activities, P3 does not take advantage of possible earlier times to finish the project earlier by utilizing those hidden working days of the other calendar.

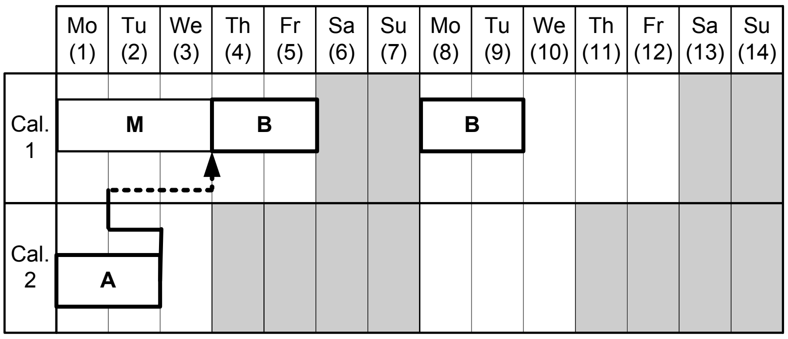
P3 itself shows that the hidden working days are valid for the successor in a late start schedule. An example is demonstrated in Figure 4.29. Figure 4.29(a) shows the initial activity relationship, and Figure 4.29(b) shows the early start schedule after the forward pass. The EST of *Activity B* calculated from *Activity A* is day-2, but because of the relationship from *Activity M*, the EST is day-4. The late start schedule (Figure 4.29(c)) calculated by the backward pass explained earlier is identical to the P3 output (Figure 4.29(d)). The TF of *Activity A* is two days, since the working day difference between EST and LST (or EFT and LFT) is two days.

Figure 4.29(c) should be observed carefully. The conditions (finish time, calendar type, and relationship to the successor) of *Activity A* (predecessor) of Figure 4.28 and 4.29(c) are same, but the start times of *Activity B* (successor) are different. The EST of *Activity B* from Figure 4.28 is day-8, which should be the earliest possible start time of *Activity B*. With the same condition, however, *Activity B* starts on day-4, which is earlier than the earliest possible start time of *Activity B*.

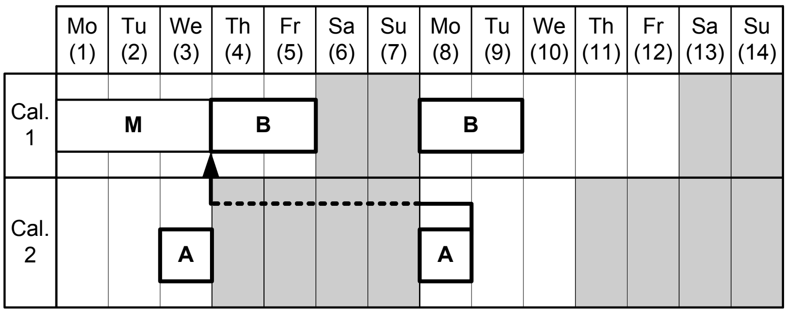
It would be arguable that Figure 4.29(c) is a backward pass output, while Figure 4.28 is a forward pass output. However, the minimum interval condition based on the activity relationship type should be satisfied in both forward and backward passes. The remaining duration of *Activity A* is one-day on the beginning of day-4, which fully satisfies the minimum interval condition of the FS:(-1) relationship from *Activity B*. If *Activity B* can start on day-4 whether it is EST or LST, day-8 cannot be the earliest possible time. Therefore, *Activity B* in Figure 4.28 can start on day-4 and end on day-8.



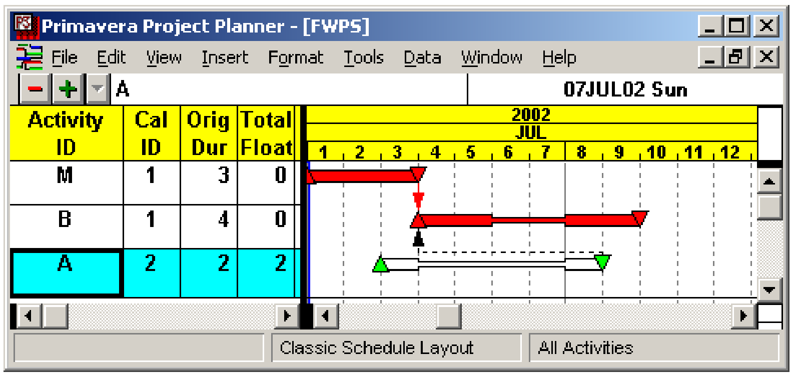
(a) activity relationship



(b) early start schedule



(c) late start schedule

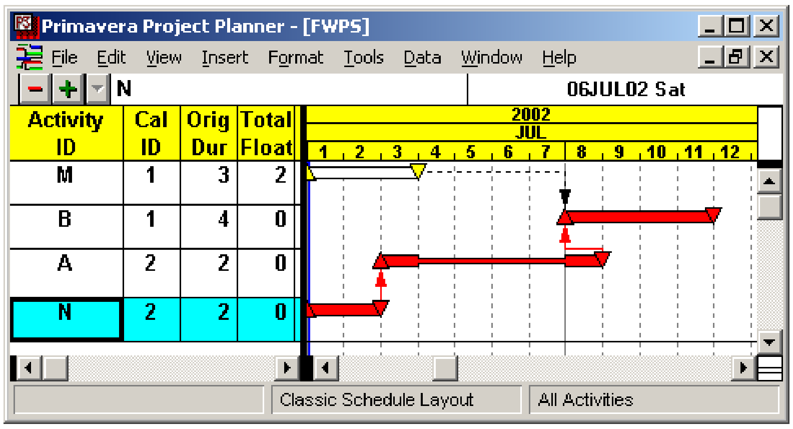

Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
A
07JUL02 Sun
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12
Classic Schedule Layout
All Activities

Activity ID	Cal ID	Orig Dur	Total Float
M	1	3	0
B	1	4	0
A	2	2	2

(d) late start schedule in P3

Figure 4.29 Example Schedule of Valid Hidden Working Days

Another example to demonstrate that day-4 is the EST of *Activity B* can be achieved by applying a simple schedule change. Suppose the start time of *Activity A* is delayed by two days, which are equal to the TF. Thus, the start and the finish of *Activity A* should be LST and LFT, respectively. There should be no time extension of the project completion of day-9 after rescheduling with this change. However, as shown in Figure 4.30[2], the finish time has been extended to day-11, two days later than the LFT. The EFT of *Activity B* should be still day-9 after rescheduling like Figure 4.29(d).


Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
N
06JUL02 Sat
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12
Classic Schedule Layout
All Activities

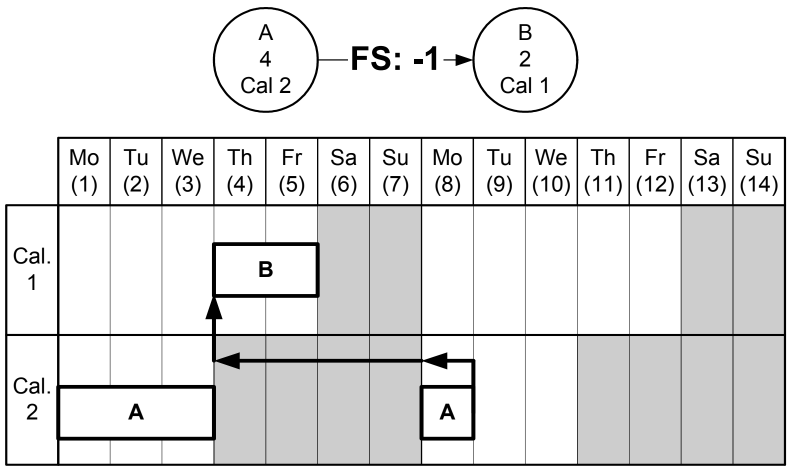
Activity ID	Cal ID	Orig Dur	Total Float
M	1	3	2
B	1	4	0
A	2	2	0
N	2	2	0

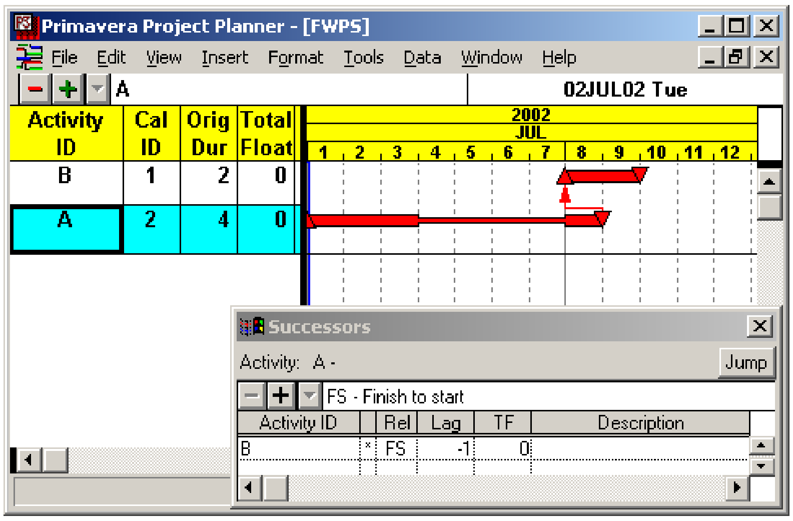
Figure 4.30 Early Start Schedule with Delayed Start

The same situations occur in all other relationship types. The process to handle this special case for all relationship types shall be as follows: when any negative lag time counting ends at the beginning of the next day of a non-working day from the predecessor's calendar, the counting continues to the beginning of the earliest non-working day passing through all continuous non-working days.

Examples for all four relationships are compared with P3 results as shown in Figure 4.31 ~ Figure 4.34. P3 generates a different (delayed) schedule for each case.

[2] In this schedule, a dummy activity *N* having a FS:0 relationship to *Activity A* is added to delay the start time of *Activity A*.

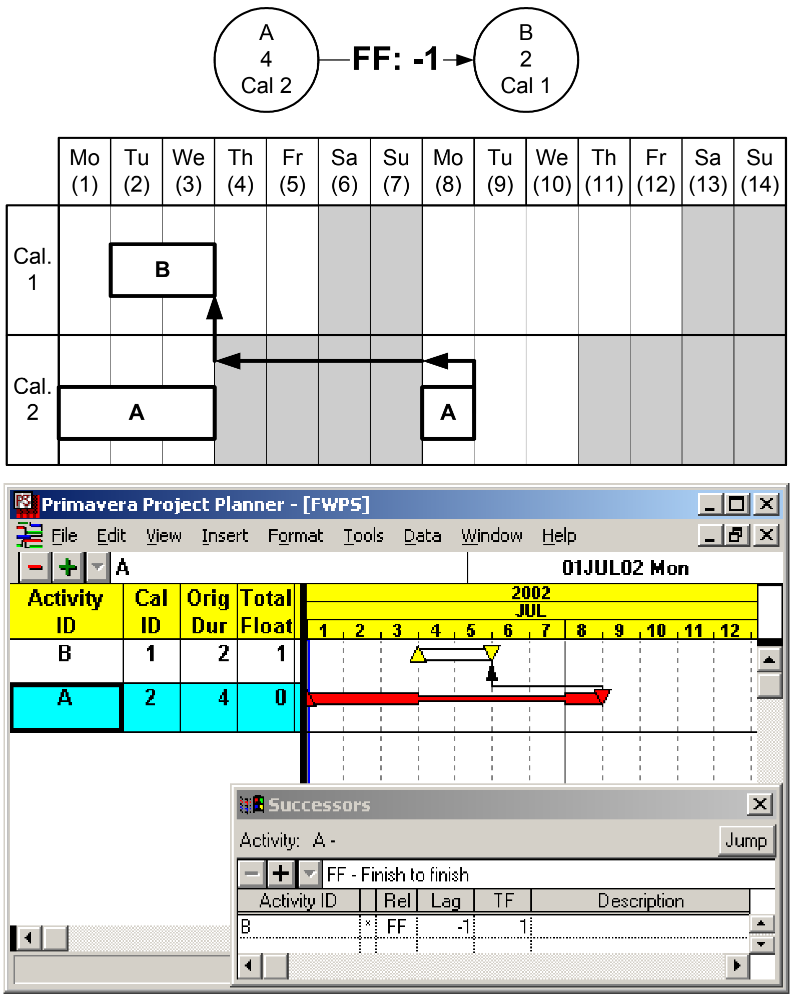




Activity ID	Cal ID	Orig Dur	Total Float
B	1	2	0
A	2	4	0

Activity ID	Rel	Lag	TF	Description
B	FS	-1	0	

Figure 4.31 Early Start Schedule in the Special Case: FS (-1)

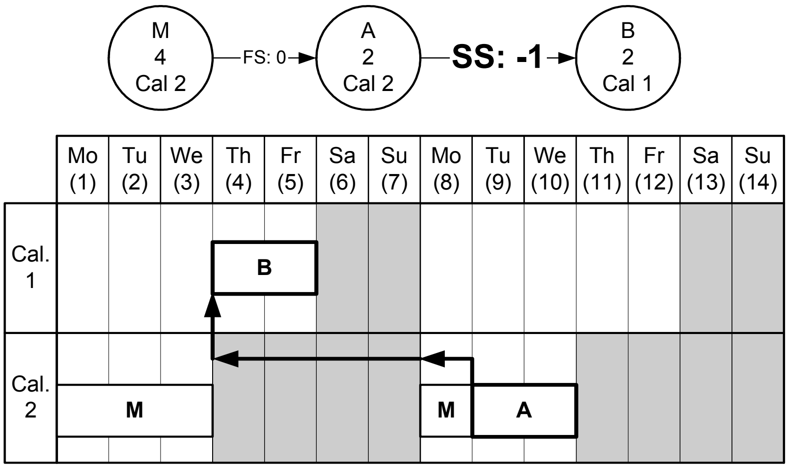


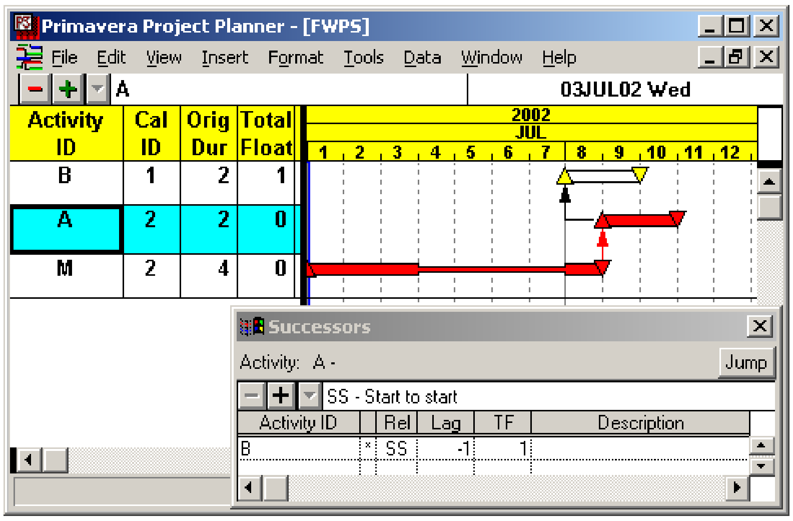
Activity ID	Cal ID	Orig Dur	Total Float
B	1	2	1
A	2	4	0

Activity: A -

Activity ID	Rel	Lag	TF	Description
B	FF	-1	1	

Figure 4.32 Early Start Schedule in the Special Case: FF (-1)




Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
A
03JUL02 Wed
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12

Activity ID	Cal ID	Orig Dur	Total Float
B	1	2	1
A	2	2	0
M	2	4	0

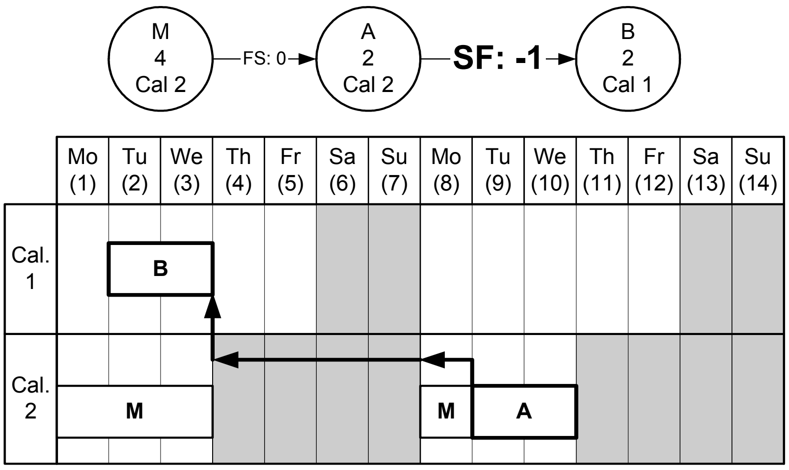
Successors

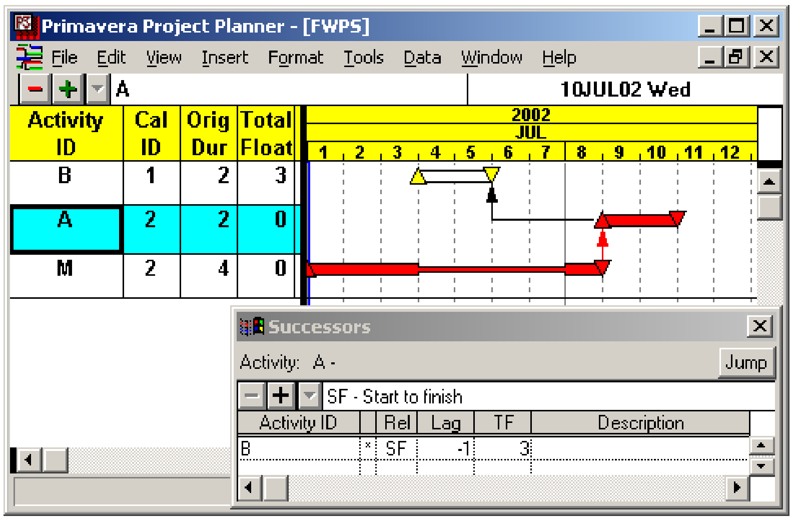
Activity: A -

SS - Start to start

Activity ID	Rel	Lag	TF	Description
B	SS	-1	1	

Figure 4.33 Early Start Schedule in the Special Case: SS (-1)




Primavera Project Planner - [FWPS]
File Edit View Insert Format Tools Data Window Help
A
10JUL02 Wed
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12

Activity ID	Cal ID	Orig Dur	Total Float
B	1	2	3
A	2	2	0
M	2	4	0

Successors

Activity: A -

SF - Start to finish

Activity ID	Rel	Lag	TF	Description
B	SF	-1	3	

Figure 4.34 Early Start Schedule in the Special Case: SF (-1)

4.6.2 Start-to-Finish with 0 Lag in P3

P3 incorrectly performs the forward pass with SF:0. Unlike other relationship types of zero lags, the earliest possible time of the successor in SF:0 is the previous day of the predecessor's time like in negative lags. Thus, the same treatment for the negative lags should be applied to SF:0.

For example as shown in Figure 4.35, *Activity A* starts at the beginning of day-8, thus its successors with SF:0 can finish at the end of day-7 or later. However, since the end of day-7 is equal to the beginning of day-8, which is the next day of a non-working day from the predecessor's calendar, the same method applied to the negative lag needs to be applied as shown on the fenced bar chart. Thus, the earliest possible finish time of *Activity B* is at the end of day-5 and no successor activity can finish earlier than day-5. In P3, however, *Activity B* has EFT of day-3, which violates the relationship condition; the interval between two activities is (-2) days, but the minimum interval by the relationship condition is zero.

Another example to prove this P3 miscalculation can be provided by applying different lag times to P3 as shown in Figure 4.36. As provided earlier, the *EFT* of an activity with different lag times must satisfy below equation.

$$EFT_{N-1} \le EFT_N$$

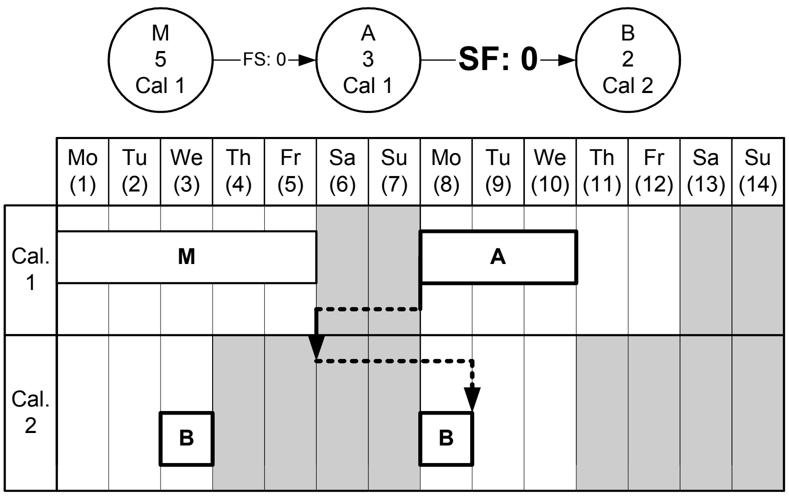
where, N *: integer,*

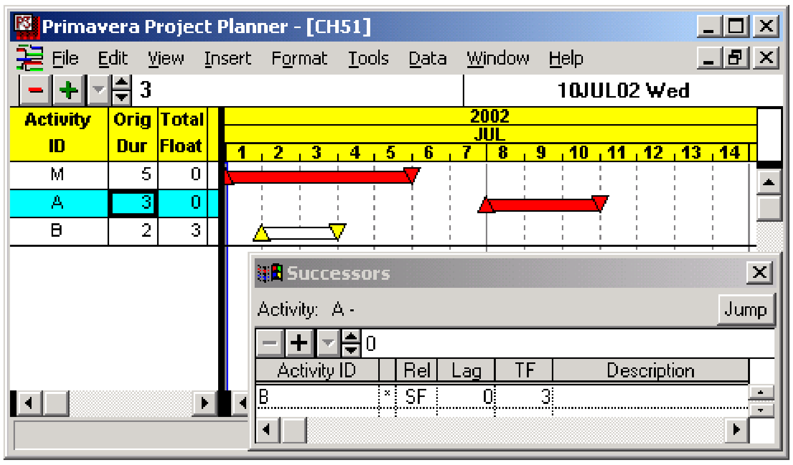
EFT_N *: EFT* of an activity with *N lag* of *SF*

The EFT of the successor that has bigger lag time must be bigger than or equal to the EFT with a smaller lag. However, Figure 4.36(b) and (c) show

$$EFT_{-1} > EFT_0$$

which violates above condition.

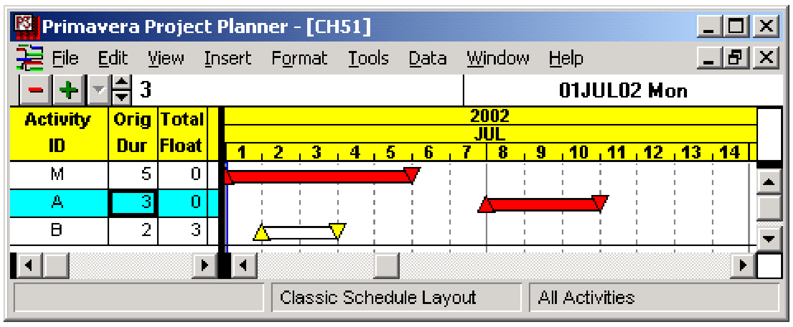



Primavera Project Planner - [CH51]
File Edit View Insert Format Tools Data Window Help
3
10JUL02 Wed
2002
JUL
1 2 3 4 5 6 7 8 9 10 11 12 13 14
Successors
Activity: A -
Jump
0

Activity ID	Orig Dur	Total Float
M	5	0
A	3	0
B	2	3

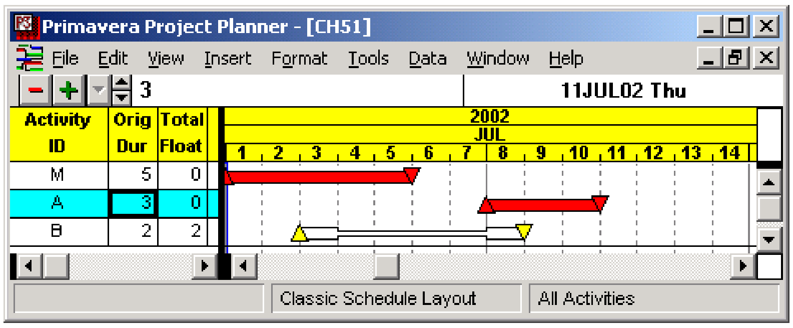
Activity ID		Rel	Lag	TF	Description
B	*	SF	0	3	

Figure 4.35 Forward Pass with SF-0 in P3



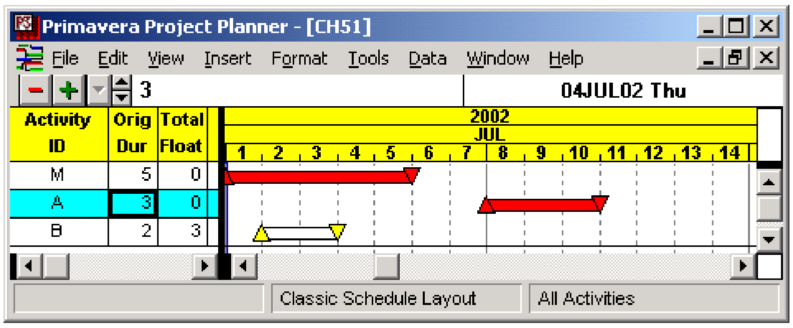
Activity ID	Orig Dur	Total Float
M	5	0
A	3	0
B	2	3

(a) SF: -2 (correct)



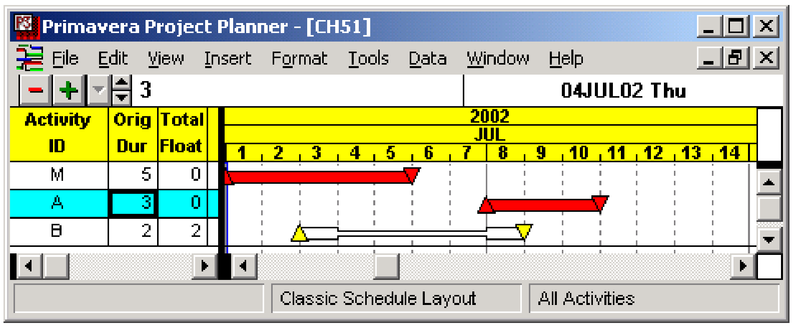
Activity ID	Orig Dur	Total Float
M	5	0
A	3	0
B	2	2

(b) SF: -1 (correct)



Activity ID	Orig Dur	Total Float
M	5	0
A	3	0
B	2	3

(c) SF: 0 (incorrect)



Activity ID	Orig Dur	Total Float
M	5	0
A	3	0
B	2	2

(d) SF: 1 (correct)

Figure 4.36 EFT with Multiple Lags in SF

4.7 RCPM with Multiple Calendars

RCPM with multiple calendars is similar to scheduling with continuous working days, but several additional features are required to apply calendars. Every calendar is created first, establishing a standard workday pattern per week, holidays, and exceptions. Then, each activity must have one calendar, and it can be scheduled only on the valid working days of its calendar. In addition, the activity search range in step 2 and 4 of the RCPM algorithm to identify resource dependency between activities is slightly expanded if non-working days are involved.

For example, when an activity is delayed due to a resource limit, step 2 of the RCPM algorithm creates one or more resource links with prescheduled activities. The search range with continuous working days in this case is only the previous day of the start time of the delayed activity. The search range with multiple calendars is also the previous day of the start time, if it is a working day. However, if the previous day is a non-working day, as shown in Figure 4.37, the search range include all continues non-working days and the first working day. In this example, *Activity A* is delayed from day-1 due to a resource limit and is scheduled starting from day-8. Some activities that finish at day 3 ~ 5 in *Calendar 1* or at day 3 ~ 7 in other calendars, could make *Activity A* scheduled, so the search is made for the whole period of day 3 ~7.

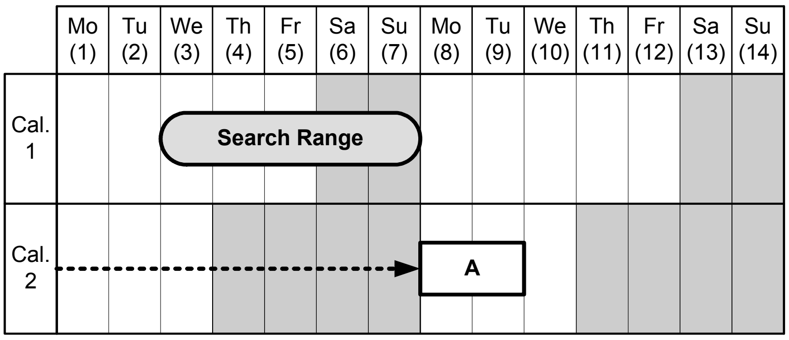


Figure 4.37 Prescheduled Activity Search Range in Step 2 with Multiple Calendars

Similar case happens in step 4 of the RCPM algorithm as shown in Figure 4.38. It is assumed that *Activity A* could have more than one day TF after step 3, but it can have only one day due to the prescheduled activities in step 4. In this case, some activities that start on day 4 ~ 8 could restrict the late finish time of *Activity A*. Thus the search range is day 4 ~ 8 to create resource links with prescheduled activities in step 4.

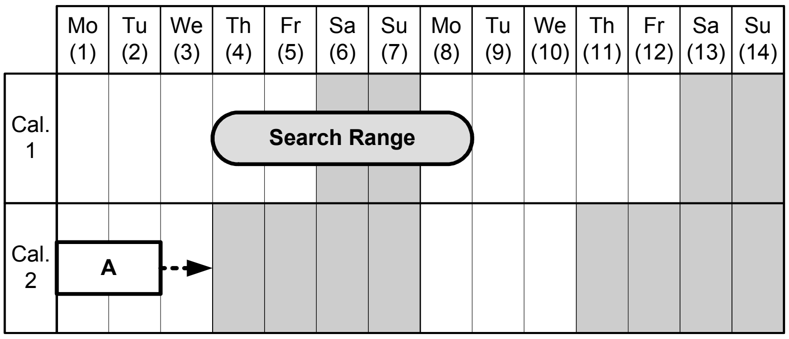


Figure 4.38 Prescheduled Activity Search Range in Step 4 with Multiple Calendars

Chapter 5

RCPM System

A prototype RCPM system is developed to implement the proposed algorithm. The main objective of this system is to evaluate the proposed algorithm with various schedules prior to its application on real projects. This chapter will explore the system architecture such as system environment, data model, processes, and functionality.

5.1 System Platform

The RCPM system is developed using Visual Basic 6.0 (VB), Visual C++ 6.0 (C++), and *Ra*™. Figure 5.1 shows their interrelationships. *Ra* is the Application Programming Interface (API) of P3, through which it is able to read and update P3 project data from outside of P3 system. VB is adopted for communication with P3 through *Ra* and the system user interface, and C++ is adopted for main algorithm implementation.

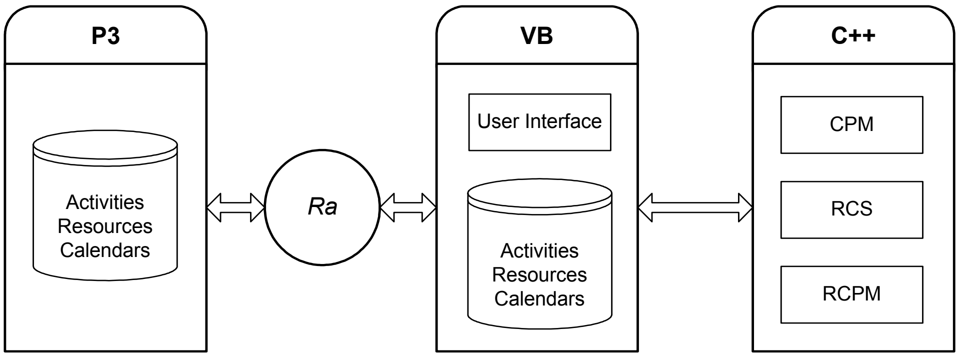


Figure 5.1 RCPM System Architecture

VB is selected because the code format of *Ra* is very similar to that of VB, and the *Ra* help system provides many useful VB code examples with P3 data access and function application [Primavera 1998]. Another benefit of using VB is its relative ease in developing a windows system, which could be appropriate for a prototype system development. On the other hand, C++ is selected because of its well-known efficiency to handle data and procedures. Most functions to implement the RCPM procedures are written in C++ and built into a DLL (Dynamic Link Libraries) file. The VB part reads required data directly from P3, executes the RCPM procedures communicating with functions in the DLL file, and updates the original P3 schedule upon user's request.

5.2 Data Model

5.2.1 Model Overview

The RCPM system is developed on the basis of the object-oriented technique. As shown on Figure 5.2, the system consists of six major classes: *Project, Activity, Calendar, Resource-Unit, Resource-Type,* and *Daily-Resource-Profile.* The classes (bold and underlined) and their interrelationships (underlined) can be identified from a high-level problem statement as follows.

> A construction <u>**project**</u> is <u>composed</u> of <u>**activities**</u> and available resources (<u>**resource type**</u>). The project should <u>have</u> at least one or more <u>**calendars**</u>. Each activity must <u>have</u> one project calendar and can have <u>precedent</u> and <u>following relationship</u> with other activities. Each activity <u>requires</u> certain resources (<u>**resource unit**</u>), each of which should <u>be one of</u> the available resource_types. <u>On every day of the project period</u>, <u>**daily resource requirement**</u> per resource type is available from <u>aggregating</u> every resource-unit required by activities that are scheduled on that day.

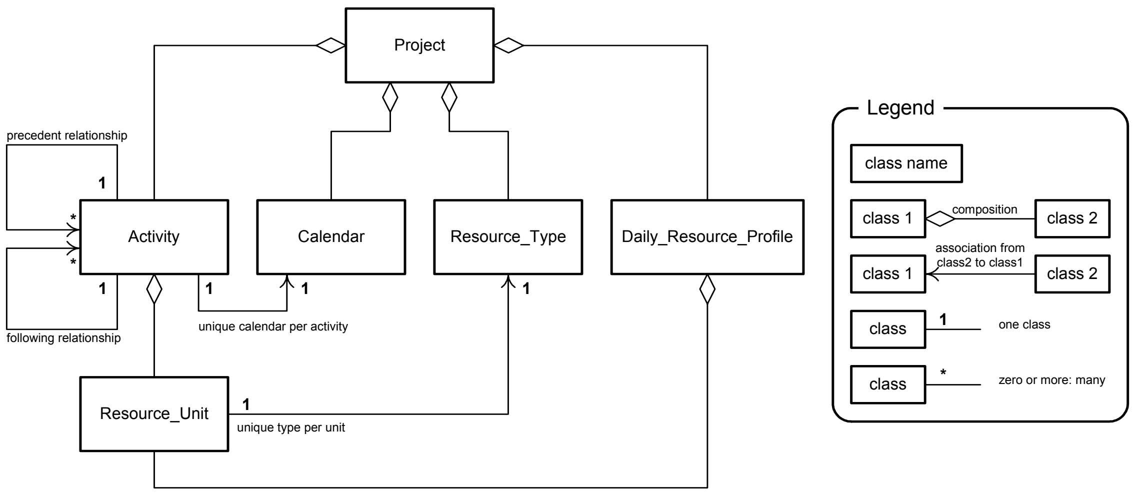


Figure 5.2 RCPM System Object Model

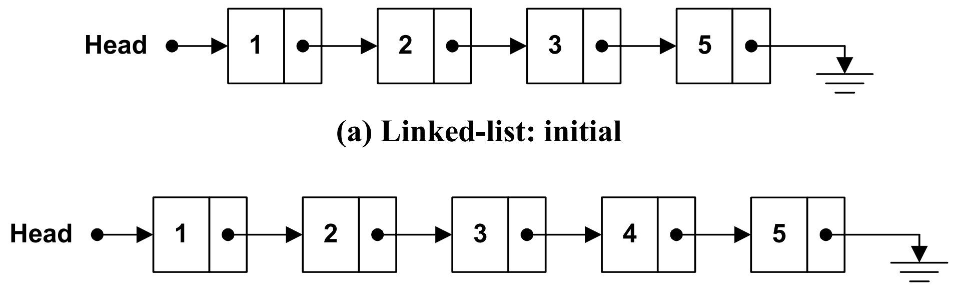
5.2.2 Linked List

A linked-list data structure is adopted for the main data structure of the RCPM system because of its well-known convenience to construct unlimited or undecided number of entities in a relatively simple way. The total numbers of entities that the RCPM system will encounter are unpredictable. Those entities such as activities, resources, calendars, predecessors, successors, etc. are project dependent so a linked-list structure can be an appropriate way to handle those uncertain numbers of data.

A simple integer linked-list example, sorted by ascending order, is shown in Figure 5.3(a). This list starts from the *Head* node and ends at *NULL* (nothing). Every entity node has a value (data) and a link to another node, constructing a linked-list data structure. In order to add a new node or delete a present node, keeping consistency of the list, a set of changes on the links is required. For example, procedures to add number 4 are as follows:

- create a new node: node of 4
- search the location of the new node: between node of 3 and node of 5
- make a link from the new node to the node of 5
- change the direction of the link from the node of 3 to the new node

Figure 5.3(b) shows the result of adding number 4. In this way, any new data can be added until the computer memory runs out, keeping the consistency of the list.



(a) Linked-list: initial

(b) Linked-list: adding a new node

Figure 5.3 Linked-list Data Structure

In C++, a node of the linked-list can contain various data types including *user defined data* types. A user defined data type can contain various data types such as integers, strings, and even other user defined data types. For example, the class *Activity* with attributes of ID, duration, a user defined predecessor linked-list, etc., is a user defined data type. Thus, any instance (object) of the classes can be built into a linked-list.

In addition, C++ provides a *template* command. With the template command, just one time declaration of a linked-list is sufficient to apply it to all data types. For example, a template linked-list declared once could be used for most types of classes in the RCPM system. Following sections describe features of the six major classes of the RCPM system.

5.2.3 Activity Class

The Activity class is the core part of the RCPM system. It contains important attributes such as durations, calendar ID, early times, late times, and relationships to other activities as shown in Figure 5.4. The attributes of the left column are directly imported from P3. Based on these data, the attributes on the right column are calculated or identified through the RCPM procedures. Each of the underlined attributes in both columns is a linked-list data type declared by the template linked-list.

Activity	
Data from P3	RCPM output
ID	EST
Description	EFT
Original Duration	LST
Remaining Duration	LFT
AST (Actual Start Time)	TF
AFT (Actual Finish Time)	FF
Calendar ID	PF (Phantom Float)
Successors (by logic)	**Predecessors (by logic)**
Daily Resource Requirement	**Predecessors (by resource)**
	Successors (by resource)
	Alternative Schedule

Figure 5.4 Activity Class Attributes

In order to save data storage and to accelerate data transfer, *pointers* are actively used in the *Activity* class as well as all other classes. The pointer in C++ points to a location (or address) of each data (including user defined data) in the computer memory. All nodes of the linked-list data type in the *Activity* class contain a pointer to the real data.

An example of the linked-list is shown in Figure 5.5. *Current_Activity* has three successors. Each node of the successor linked-list contains a relationship type, a lag time, an address of the actual activity data, and a link to another node. The numbers on the top of each activity indicates the memory address of each actual activity data. If *Current_Activity* needs information of a successor activity, it can directly access the successor through the pointer. In this way, large amounts of computer memory and computation (search) times can be saved in real construction projects which could have thousands of activities and dozens of thousands relationships.

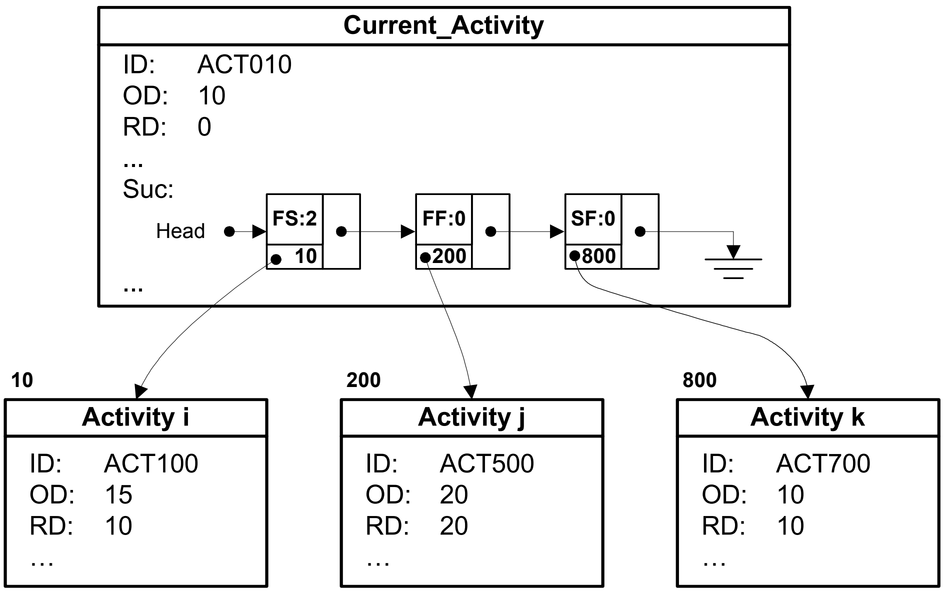


Figure 5.5 Successor Linked-List Example

5.2.4 Calendar Class

Each calendar is constructed in a static array for simplicity in calendar implementation. The size of the array is fixed to 32,000 days, which a "short" (2 bytes) data type in C++ can represent. Figure 5.6 shows an example of a 5-day workweek calendar. The array index automatically becomes cumulative days from the start time of the project, and each entity of the array contains a cumulative working days starting from day 1, Monday in this example. If the value of an entity is the same as the previous entity, the entity becomes a non-working day. In this calendar, day 6, 7, 13, and 14 are non-working days.

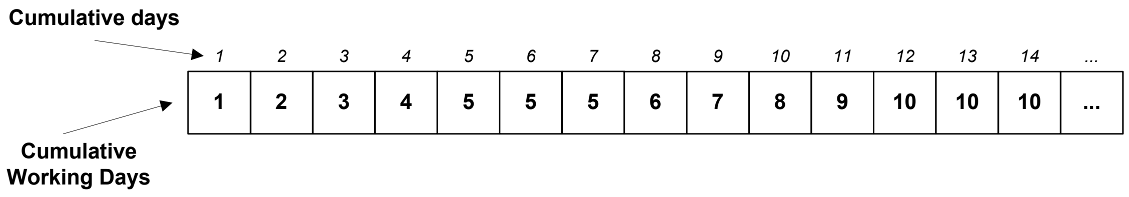
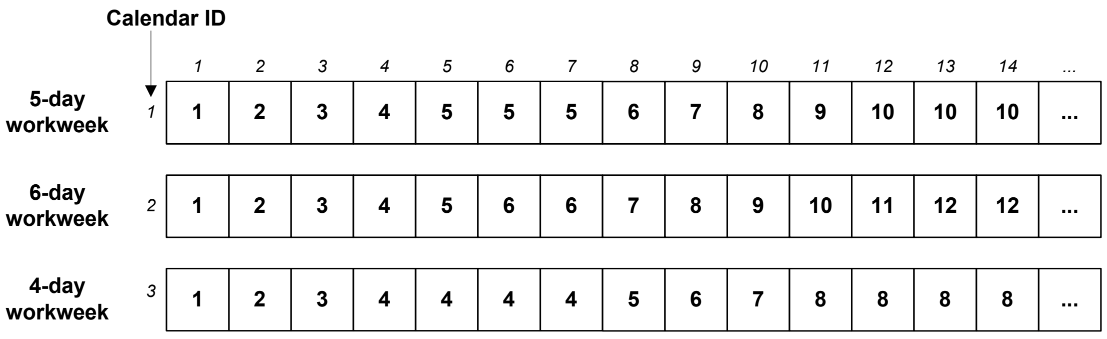


Figure 5.6 Calendar Array Example

Multiple calendars, on the other hand, are implemented by a two-dimensional dynamic array. Unlike the static array like each calendar array with a fixed size, multiple calendars are declared during an RCPM system running time by the exact number of project calendars that are imported from P3. Thus, there will be no memory waste due to unnecessary calendar arrays. Figure 5.7 shows an example of three calendars in a project. Since each activity contains a *Calendar ID* as shown in Figure 5.4, the corresponding calendar is available to all activities through the index of the multiple calendar array.



Calendar ID
1 2 3 4 5 6 7 8 9 10 11 12 13 14 ...
5-day workweek 1: 1 2 3 4 5 5 5 6 7 8 9 10 10 10 ...
6-day workweek 2: 1 2 3 4 5 6 6 7 8 9 10 11 12 12 ...
4-day workweek 3: 1 2 3 4 4 4 4 5 6 7 8 8 8 8 ...

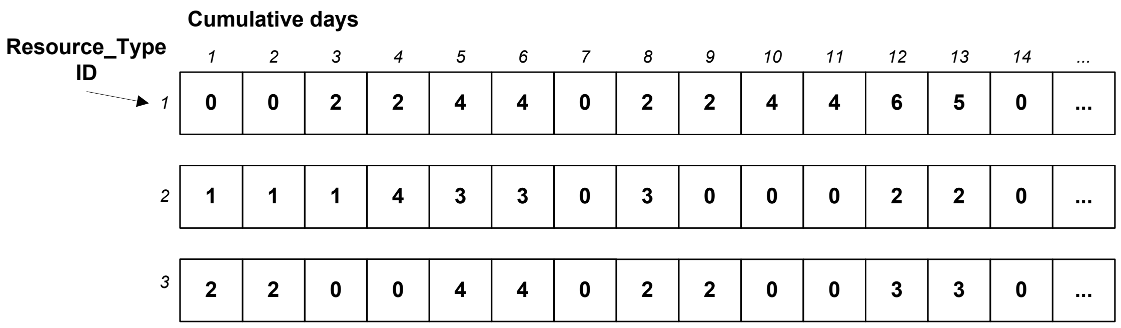
Figure 5.7 Multiple Calendar Example

5.2.5 Resource_Type and Resource_Unit Class

The *Resource_Type* class contains ID, Name, and Max-available-quantity. The *Resource_Unit* class contains a pointer to a *Resource_Type* and the daily requirement quantity of the resource type for the activity. The *Resource_Type* objects are built in a linked-list and contained in *Project* class. *Resource_Unit* is also built in a linked-list but contained in the *Activity* class.

5.2.6 Daily_Resource_Profile Class

The structure of *Daily_Resource_Profile* class is similar to *Calendar* class. Daily requirement of each resource type is constructed in a static array, and arrays for each resource type is constructed dynamically, making a two-dimensional array. Figure 5.8 shows an example of a daily resource profile structure of three resource types. Since the resource requirement (*Resource_Unit*) of each activity contains a resource ID, the resource profile can be built by aggregating them per *Resource_Type*.



Resource_Type ID \ Cumulative days	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
1	0	0	2	2	4	4	0	2	2	4	4	6	5	0	...
2	1	1	1	4	3	3	0	3	0	0	0	2	2	0	...
3	2	2	0	0	4	4	0	2	2	0	0	3	3	0	...

Figure 5.8 Daily Resource Profile Example

5.2.7 Project Class

An object of the *Project* class represents a whole project so that it contains everything to do RCPM. Figure 5.9 shows major attributes of the Project class. The attributes of the left column are directly imported from P3. Based on these data, the attributes on the right column are constructed before/during/after the RCPM procedures. Descriptions of each attribute are as follows.

Project	
Data from P3	Data in RCPM
Start_Date	End_Date
Data_Date	Start_Activity_List
Activity_List	Terminal_Activity_List
Calendar_List	Alternative_Schedule_Activity_List
Resource_Type_List	Daily_Resource_Profile

Figure 5.9 Project Class Attributes

- Start_Date: the project start date, same as in P3
- Data_Date: the start time of scheduling, same as in P3.
- Activity_List: a linked-list to store all activities. Each node of this list contains real data.

- Calendar_List: a linked-list to store all calendars. Each node of this list contains real data.
- Resource_Type_List: a linked-list to store all resource types. Each node of this list contains real data.
- End_Date: the project completion time in CPM or RCPM
- Start_Activity_List: a linked-list to store all start activities, which do not have any predecessors. Each node of this list contains a pointer to the activity object stored in the Activity_List.
- Terminal_Activity_List: a linked-list to store all terminal activities, which do not have any successors. Each node of this list contains a pointer to the activity object stored in the Activity_List.
- Alternative_Schedule_Activity_List: a linked-list to store all activities that can have alternative start and finish times. Each node of this list contains only a pointer to the activity object stored in the Activity_List.
- Daily_Resource_Profile: a two-dimensional array to store daily requirement per resource_type.

A Precedence Diagramming Method (PDM) network is internally available in the system level by the *Start_Activity_List*, *Terminal_Activity_List*, and the linked-lists of *predecessors* and *successors* in each activity. The forward pass starts from the *Start_Activity_List*, each activity of which contains pointers to successors, each of which contains pointers to successors, and so forth, traversing all activities until a terminal activity is encountered. In the same way, the backward pass starts from the *Terminal_Activity_List* traversing all activities by the predecessor linked-list of each activity.

5.2.8 Data in Visual Basic

Aforementioned data structures are built within the C++ part. Again, VB is adopted because of the benefit in *Ra* application and Windows programming. However, there is some cost in combining VB and C++, especially in transferring or referring data between the two different programming languages. In order to refer data, an identical format of corresponding data type should be declared in both VB and C++ programming codes. VB, however, does not provide the *pointer* feature that is necessary to construct a linked-list data structure or to construct the same data type as in C++. Hence, an intermediate data structure is required in both VB and C++. While the VB part contains real data, the C++ part does not contain any data in this data structure, but the declaration of them in the C++ enables the C++ codes to access the VB data.

The attributes of the intermediate data structure are the same as the main C++ data structure except the linked-list data types. Static arrays replace the linked-lists. The size of array is intuitively set after observing a number of P3 projects. The limits of the current RCPM system because of the fixed size of array are as follows.

- Number of successors (by logic) of each activity: 100
- Number of successors (by resource) of each activity: 100
- Number of resource types for each activity: 20

5.3 Process Model

The main processes of the RCPM system are data importing from P3, CPM, RCS creating resource links, finding alternative schedule, and adding resource links into the P3 project. This section will discuss these key processes.

5.3.1 Data Importing

The RCPM system imports the schedule data from P3. These data are 1) activity with ID, description, original duration, remaining duration, actual start, actual finish, successors, and daily resource requirement, 2) resource with name and maximum available quantity, and 3) calendar with ID and workweek pattern. All these data are stored in the VB environment and is available for further processes as follows.

5.3.2 CPM

The process for the CPM shown on Figure 5.10 is a direct conversion of the manual calculation. At first, all three activity lists (*Activity_List, Start_Activity_List,* and *Terminal_Activity_List*) should be created. The forward pass starts from the activity in the first node of the *Start_Activity_List* and traverses all activities by the *Forward Pass* process and the successor linked-list of each activity. In the same way, the backward pass starts form the activity in the first node of the *Terminal_Activity_List* and traverses all activities by the *Backward Pass* process and the predecessor linked-list of each activity.

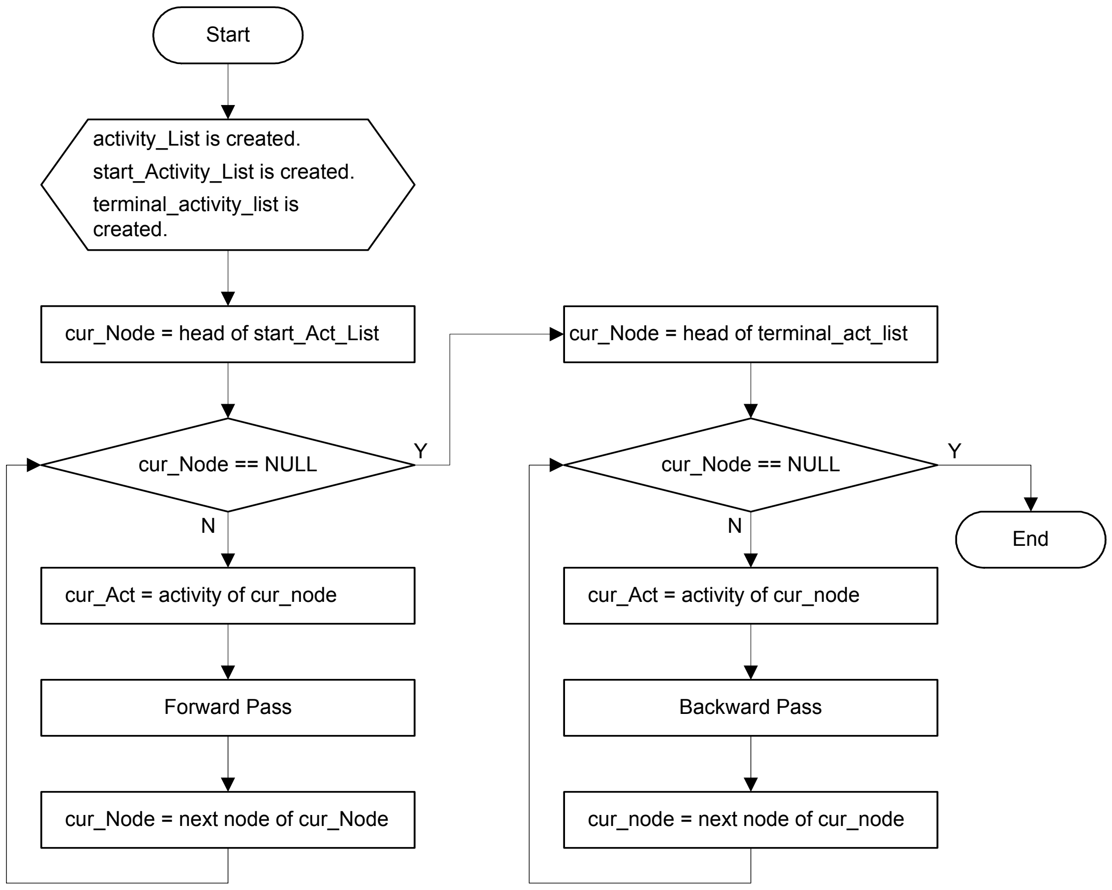

Start
activity_List is created.
start_Activity_List is created.
terminal_activity_list is created.
cur_Node = head of start_Act_List
cur_Node == NULL
Y
N
cur_Act = activity of cur_node
Forward Pass
cur_Node = next node of cur_Node
cur_Node = head of terminal_act_list
cur_Node == NULL
Y
N
End
cur_Act = activity of cur_node
Backward Pass
cur_node = next node of cur_node

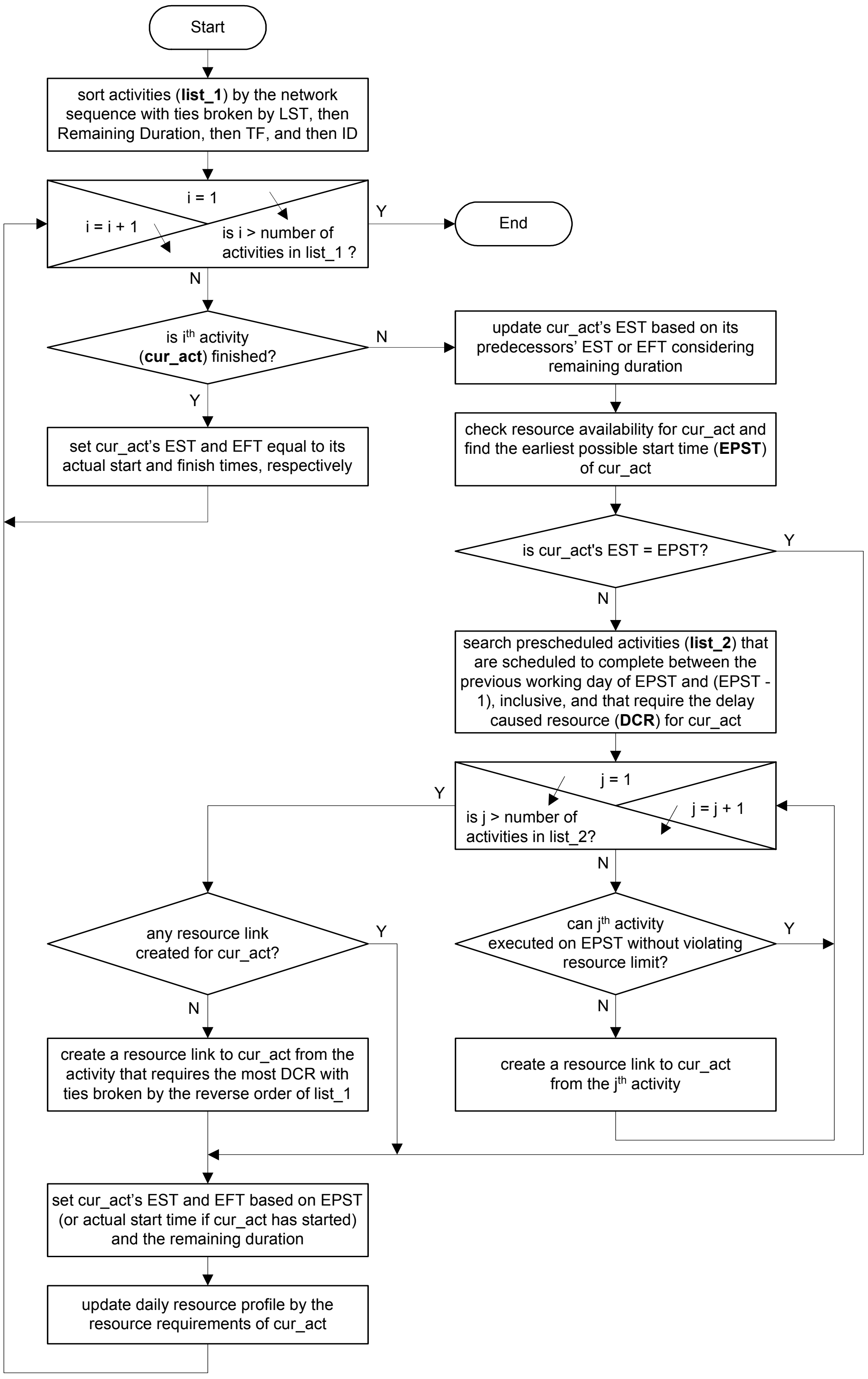
Figure 5.10 CPM Process

It should be noted that any activity that already started or/and finished at the current *Data_Date* have actual start or/and finish times instead of the CPM calculated times. However, all other activities including started but not-finished activities will have CPM calculated times that starts from the *Data_Date* considering remaining durations and activity relationships.

5.3.3 RCPM Forward Pass

The RCPM forward pass process is shown on Figure 5.11. Like the CPM process, the RCPM process considers activities that have remaining durations. Hence, the periods before the *Data_Date* is not considered for resource availability checking since these activities have zero remaining duration and already started or/and finished no matter what was the resource availability.

The RCPM forward pass process starts from sorting activities by the network sequence and other priorities (LST, Remaining Duration, TF, and activity ID). If the current activity is completed by *Data_Date*, the EST and EFT will be set as its actual start and finish times, respectively, without checking resource availability. For all incomplete activities, the "update cur_act's EST …" process enables the system to keep the updated EST of the current activity. This process is required because any precedent activity's delay caused by resource constraints makes the current activity's EST ineffective. When the current activity is delayed due to a resource limit, resource links are created to the current activity from appropriate prescheduled activities. As described earlier, these activities are scheduled to finish immediately before current activity's Earliest Possible Start Time (EPST) on the current activity's calendar, releasing certain resources that caused the current activity's delay. In order to consider non-working days, the search to find those completed activities is made from the previous working day of EPST to the previous day of EPST based on the current activity's calendar. After this RCPM forward pass, parts of all required resource links are created. Each activity's EST is set by the EPST if the activity has not started or by the actual start time if it has started, and the EFT are calculated from the EST and the remaining duration.



Start
sort activities (**list_1**) by the network sequence with ties broken by LST, then Remaining Duration, then TF, and then ID
i = 1
i = i + 1
is i > number of activities in list_1 ?
Y
End
N
is ith activity (**cur_act**) finished?
N
update cur_act's EST based on its predecessors' EST or EFT considering remaining duration
Y
set cur_act's EST and EFT equal to its actual start and finish times, respectively
check resource availability for cur_act and find the earliest possible start time (**EPST**) of cur_act
is cur_act's EST = EPST?
Y
N
search prescheduled activities (**list_2**) that are scheduled to complete between the previous working day of EPST and (EPST - 1), inclusive, and that require the delay caused resource (**DCR**) for cur_act
j = 1
j = j + 1
is j > number of activities in list_2?
Y
N
any resource link created for cur_act?
Y
N
can jth activity executed on EPST without violating resource limit?
Y
N
create a resource link to cur_act from the activity that requires the most DCR with ties broken by the reverse order of list_1
create a resource link to cur_act from the jth activity
set cur_act's EST and EFT based on EPST (or actual start time if cur_act has started) and the remaining duration
update daily resource profile by the resource requirements of cur_act

Figure 5.11 RCPM Forward Pass

5.3.4 RCPM Backward Pass

Another heuristic approach is introduced to find additional resource links and accordingly the late times of each activity. An important property of identifying activity's float is that the float amount of an early scheduled activity is decided by late scheduled activities (see *Activity E* and *F* in Figure 3.8). Based on this property, each activity in the network reverse order is checked for resource availability during TF periods. Like other previous procedures, uncompleted activities at the time of Data_Date are considered while finding resource links.

Figure 5.12 shows the RCPM backward pass process. Activities are sorted by the reversed network sequence and other priorities (bigger EFT, smaller duration, and bigger ID) from the schedule after the backward pass with partial resource links created during the RCPM forward pass. The process of "update cur_act's LST …" enables the system to keep the updated late times and total float of the current activity. This process is required because the late time change of any successor activity should be reflected to the current activity. If the full initial TF is not available, resource links are created from the current activity to appropriate late scheduled activities. These activities are scheduled to start between the next day of the Latest Possible Finish Time (LPFT) and the next working day of LPFT of the current activity because of possible non-working days. Through this heuristic approach, the late times and the TF of an activity can be calculated without complex computation processes.

5.3.5 Finding Alternative Schedules

The process to find alternative schedules is shown in Figure 5.13. The process checks every activity that has successors by resource links for alternative schedules. The checking period starts from the next day of the LFT of current activity (after RCPM backward pass) and ends on the previous day of the earliest LST of all successors by technological relationships (or the project completion time if there is no successor). There could be several alternative schedules for an activity if available periods are present intermittently throughout the checking period.

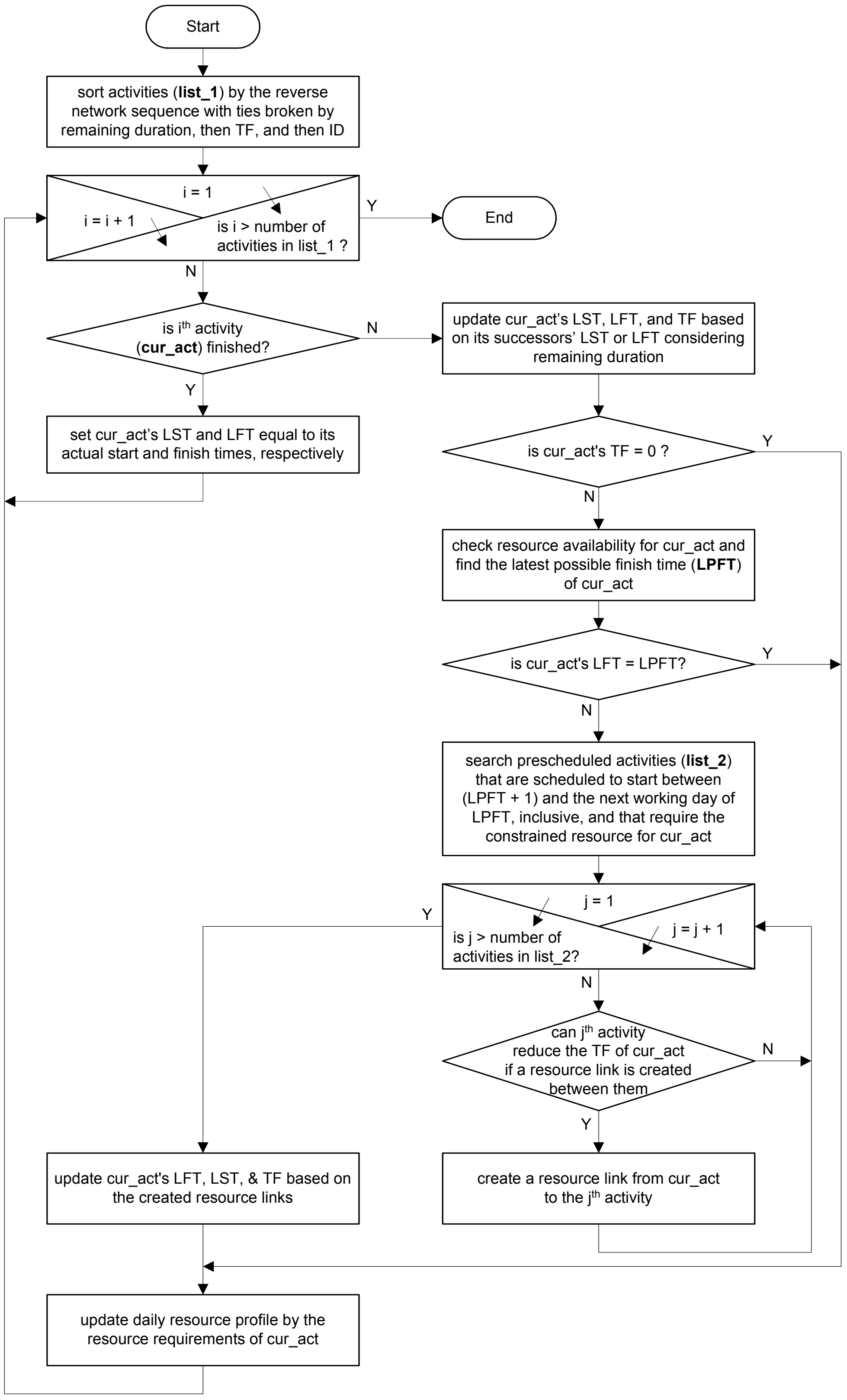

Start
sort activities (list_1) by the reverse network sequence with ties broken by remaining duration, then TF, and then ID
i = 1
i = i + 1
is i > number of activities in list_1 ?
Y
End
N
is ith activity (cur_act) finished?
N
update cur_act's LST, LFT, and TF based on its successors' LST or LFT considering remaining duration
Y
set cur_act's LST and LFT equal to its actual start and finish times, respectively
is cur_act's TF = 0 ?
Y
N
check resource availability for cur_act and find the latest possible finish time (LPFT) of cur_act
is cur_act's LFT = LPFT?
Y
N
search prescheduled activities (list_2) that are scheduled to start between (LPFT + 1) and the next working day of LPFT, inclusive, and that require the constrained resource for cur_act
j = 1
Y
is j > number of activities in list_2?
j = j + 1
N
can jth activity reduce the TF of cur_act if a resource link is created between them
N
Y
update cur_act's LFT, LST, & TF based on the created resource links
create a resource link from cur_act to the jth activity
update daily resource profile by the resource requirements of cur_act

Figure 5.12 RCPM Backward Pass

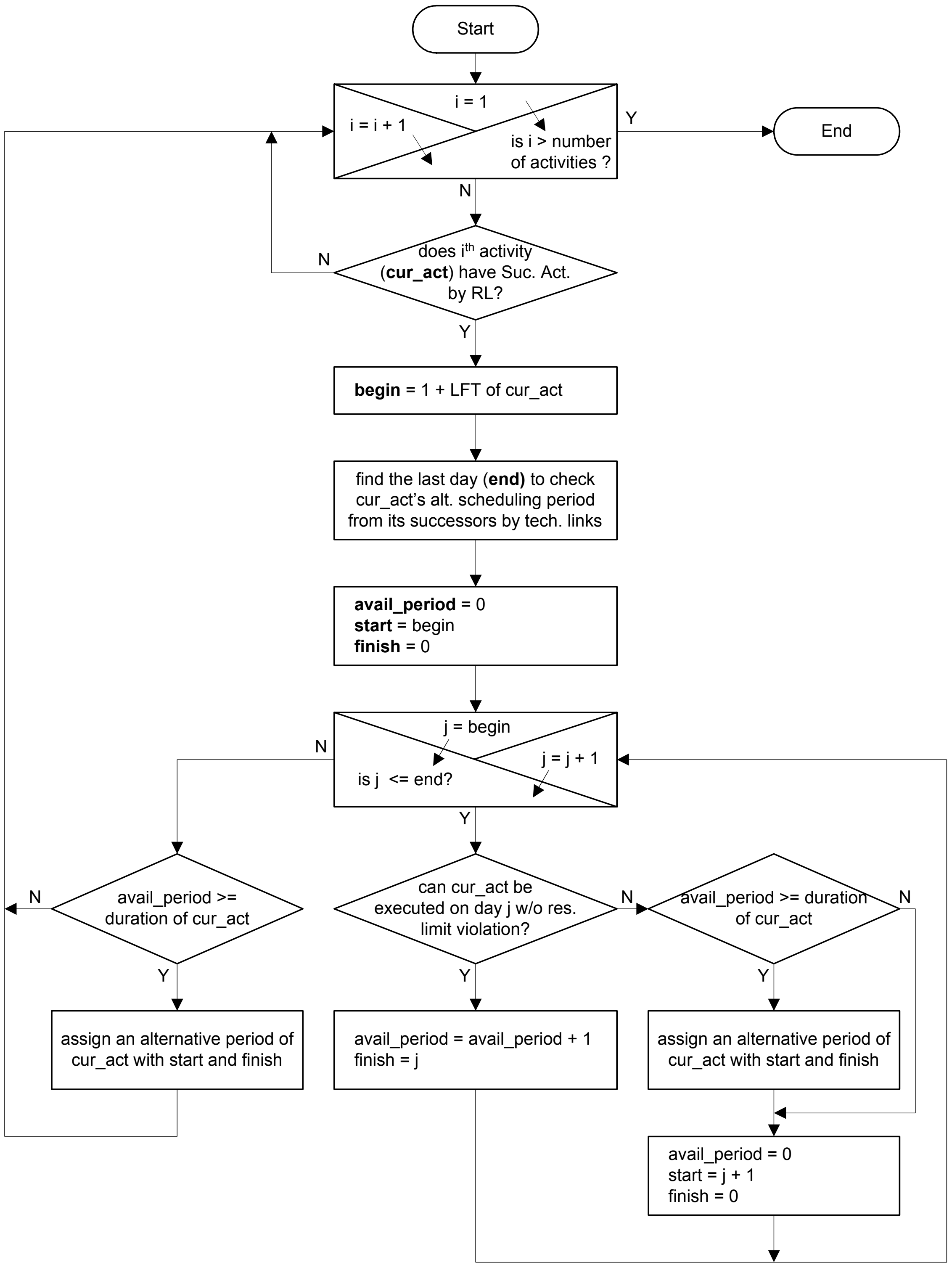
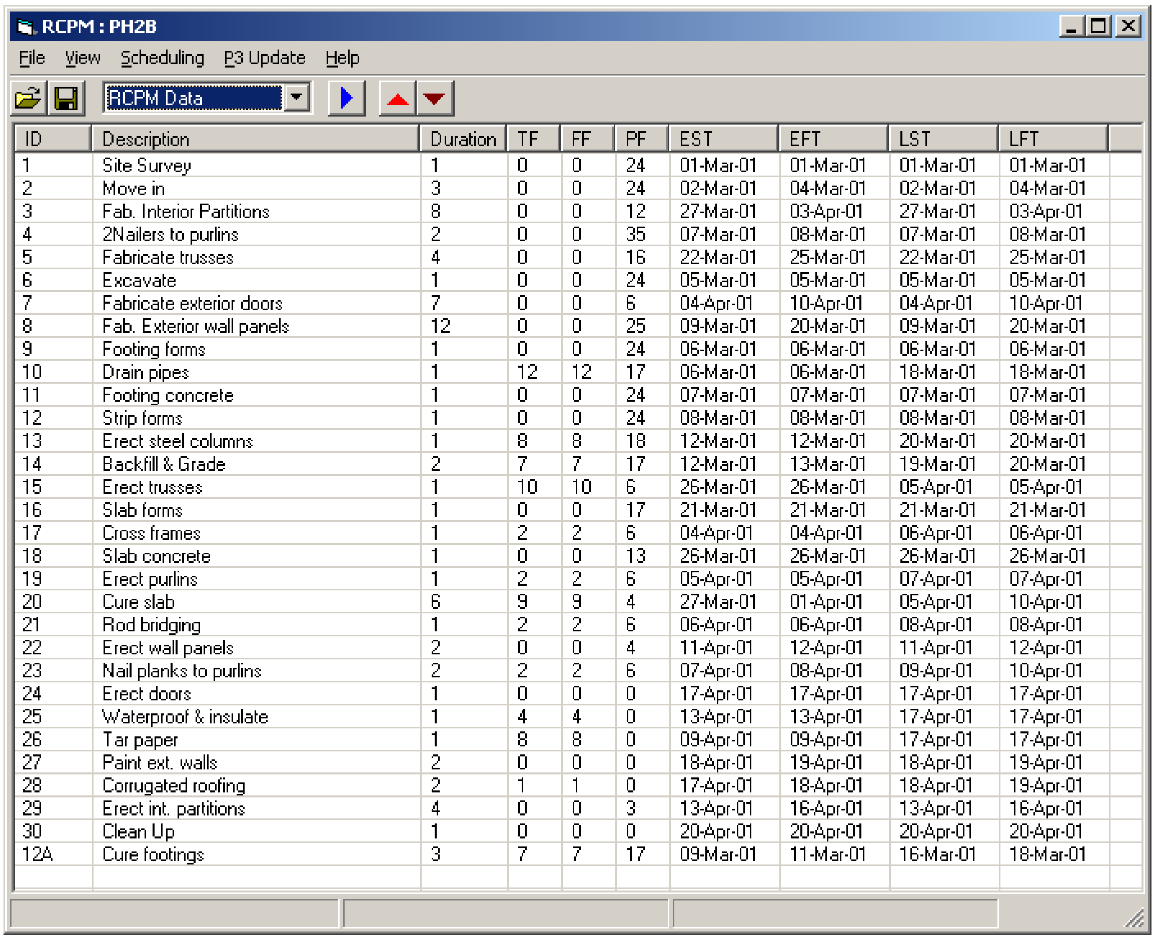

Start
i = 1
i = i + 1
is i > number of activities ?
Y
End
N
does i^th activity (cur_act) have Suc. Act. by RL?
N
Y
begin = 1 + LFT of cur_act
find the last day (end) to check cur_act's alt. scheduling period from its successors by tech. links
avail_period = 0
start = begin
finish = 0
j = begin
j = j + 1
is j <= end?
N
Y
avail_period >= duration of cur_act
N
Y
can cur_act be executed on day j w/o res. limit violation?
N
Y
avail_period >= duration of cur_act
N
Y
assign an alternative period of cur_act with start and finish
avail_period = avail_period + 1
finish = j
assign an alternative period of cur_act with start and finish
avail_period = 0
start = j + 1
finish = 0

Figure 5.13 Process of Finding Alternative Schedules

5.4 RCPM System User Interface

The RCPM system user interface is shown in Figure 5.14. The system has various menus such as *open* and *save* under *File* menu, several views under *View* menu, *CPM*, *RCS*, and *RCPM* under *Scheduling* menu, and *add* and *remove resource links* under *P3 update* menu. The system also provides several buttons to directly execute some menus. Detailed descriptions for those major functions are described as follows.



ID	Description	Duration	TF	FF	PF	EST	EFT	LST	LFT
1	Site Survey	1	0	0	24	01-Mar-01	01-Mar-01	01-Mar-01	01-Mar-01
2	Move in	3	0	0	24	02-Mar-01	04-Mar-01	02-Mar-01	04-Mar-01
3	Fab. Interior Partitions	8	0	0	12	27-Mar-01	03-Apr-01	27-Mar-01	03-Apr-01
4	2Nailers to purlins	2	0	0	35	07-Mar-01	08-Mar-01	07-Mar-01	08-Mar-01
5	Fabricate trusses	4	0	0	16	22-Mar-01	25-Mar-01	22-Mar-01	25-Mar-01
6	Excavate	1	0	0	24	05-Mar-01	05-Mar-01	05-Mar-01	05-Mar-01
7	Fabricate exterior doors	7	0	0	6	04-Apr-01	10-Apr-01	04-Apr-01	10-Apr-01
8	Fab. Exterior wall panels	12	0	0	25	09-Mar-01	20-Mar-01	09-Mar-01	20-Mar-01
9	Footing forms	1	0	0	24	06-Mar-01	06-Mar-01	06-Mar-01	06-Mar-01
10	Drain pipes	1	12	12	17	06-Mar-01	06-Mar-01	18-Mar-01	18-Mar-01
11	Footing concrete	1	0	0	24	07-Mar-01	07-Mar-01	07-Mar-01	07-Mar-01
12	Strip forms	1	0	0	24	08-Mar-01	08-Mar-01	08-Mar-01	08-Mar-01
13	Erect steel columns	1	8	8	18	12-Mar-01	12-Mar-01	20-Mar-01	20-Mar-01
14	Backfill & Grade	2	7	7	17	12-Mar-01	13-Mar-01	19-Mar-01	20-Mar-01
15	Erect trusses	1	10	10	6	26-Mar-01	26-Mar-01	05-Apr-01	05-Apr-01
16	Slab forms	1	0	0	17	21-Mar-01	21-Mar-01	21-Mar-01	21-Mar-01
17	Cross frames	1	2	2	6	04-Apr-01	04-Apr-01	06-Apr-01	06-Apr-01
18	Slab concrete	1	0	0	13	26-Mar-01	26-Mar-01	26-Mar-01	26-Mar-01
19	Erect purlins	1	2	2	6	05-Apr-01	05-Apr-01	07-Apr-01	07-Apr-01
20	Cure slab	6	9	9	4	27-Mar-01	01-Apr-01	05-Apr-01	10-Apr-01
21	Rod bridging	1	2	2	6	06-Apr-01	06-Apr-01	08-Apr-01	08-Apr-01
22	Erect wall panels	2	0	0	4	11-Apr-01	12-Apr-01	11-Apr-01	12-Apr-01
23	Nail planks to purlins	2	2	2	6	07-Apr-01	08-Apr-01	09-Apr-01	10-Apr-01
24	Erect doors	1	0	0	0	17-Apr-01	17-Apr-01	17-Apr-01	17-Apr-01
25	Waterproof & insulate	1	4	4	0	13-Apr-01	13-Apr-01	17-Apr-01	17-Apr-01
26	Tar paper	1	8	8	0	09-Apr-01	09-Apr-01	17-Apr-01	17-Apr-01
27	Paint ext. walls	2	0	0	0	18-Apr-01	19-Apr-01	18-Apr-01	19-Apr-01
28	Corrugated roofing	2	1	1	0	17-Apr-01	18-Apr-01	18-Apr-01	19-Apr-01
29	Erect int. partitions	4	0	0	3	13-Apr-01	16-Apr-01	13-Apr-01	16-Apr-01
30	Clean Up	1	0	0	0	20-Apr-01	20-Apr-01	20-Apr-01	20-Apr-01
12A	Cure footings	3	7	7	17	09-Mar-01	11-Mar-01	16-Mar-01	18-Mar-01

Figure 5.14 RCPM Program User Interface

5.4.1 File Menu

Open

This menu opens a P3 project that is **under the default P3 project directory**. When P3 is installed into a personal computer, the default project directory, e.g. C:\P3WIN\PROJECTS, is created. One benefit of the RCPM system is that it checks resource availability during project opening process. For example, if the maximum availability of a certain resource is less than the requirement of any activity, the system forces users to increase the maximum of the resource to meet the requirement[1].

Save

This menu creates a number of text files of RCPM scheduling outputs such as all activities, critical activities, alternative activity schedules, resource links, and daily resource usage table. These outputs will help users to analyze the project.

5.4.2 View Menu

This menu has various data view sub-menus such as activity input, resource input[2], CPM output, RCS output, RCPM output, phantom floats, predecessor resource links, successor resource links, and alternative activity schedules. These sub-menus are automatically turned-on and -off based on the programming running status. For example, when a user just opened a project, only activity input and resource input views are available. If a user executes *RCPM* scheduling, then all views are available.

5.4.3 Scheduling Menu

There are three submenus under this menu: *CPM*, *RCS*, and *RCPM*. These functions are the core part of the RCPM system written in C++. The RCPM system considers progressed schedules, multiple calendars, and PDM relationships such as start-to-start, start-to-finish, finish-to-start and finish-to-finish.

CPM

This menu executes the critical path method doing forward and backward passes. Like the *schedule* function under *Tools* menu in P3, CPM calculates early/late times, total floats, and free floats.

RCS

This menu executes serial resource-constrained scheduling[3] without considering resource links. Theoretically, its output should be identical to that of P3, because both are using the same algorithm for resource constrained scheduling.

RCPM

This menu executes serial resource constrained scheduling[3] with considering resource links to keep track of resource dependency. This function finds resource links, phantom floats of RCS, real floats, real critical path, and alternative activity schedules.

5.4.4 P3 Update Menu

There are two sub-menus under this menu: *Add Res Link* and *Remove Res Link*. These are the only functions of the RCPM system that modify the original P3 project. Hence, it is highly recommended to **backup the original P3 project** prior to these processes.

Add Res Link

This menu inserts resource links into the original P3 project[4]. In order to differentiate resource links with technological links, the RCPM system notes resource link information (successors only) in the *Log* of each activity.

[1] This process does not increase the maximum resource availability of the original P3 project. It only increases the maximum in the RCPM system environment.
[2] The quantity of **an unlimited resource** is shown as "0".
[3] The maximum daily resource usage per resource type should not be greater than 32,767.
[4] Since the P3 project now contains resource links, running CPM (Tools | Schedule) in P3 now considers resource availability and generates a resource-constrained CPM (RCPM) schedule.

Remove Res Link

This menu removes resource links in the selected P3 project based on the activity *Log* information. If the *Log* information were different with the initially written, the RCPM system might not recognize it and could not remove the resource link.

5.5 RCPM System Limitation

The RCPM system does not consider various scheduling options that P3 provides. As a prototype system, it is developed mainly focusing on the network logic and the constant maximum resource availability. The P3 options, identified so far, that the RCPM system does not work with are as follows.

- activity constraints such as early start/finish, late start/finish, start on, mandatory start/finish, expected finish, etc.
- immediate priority option in the leveling type for each activity
- hammock activity type; this activity type makes a loop in the network causing a system crash; all types of activities are a task type activity in RCPM
- activity split option
- resource smoothing option
- limited resource selection option; all resource that have a max limit value are limited in RCPM
- changeable resource limit during a project
- repeating options in holidays and exceptions of calendars
- automatic holiday adjusting option in the global calendar when the holiday is on a weekend

The RCPM system has been developed and tested under Windows 2000^{TM}. When an operating system crash occurs because of unexpected constraints or when the RCPM system somehow does not work properly, following applications were terminated from “Windows Task Manger” prior to re-running the RCPM system.

- *RCPM* program under *Applications* tab

- *p3engine.exe* under *Processes* tab
- *p3btrv.exe* under *Processes* tab

5.6 Required P3 Options

In addition to the RCPM system limitations, several P3 options should be set as follows in order to synchronize the P3 output with the RCPM output.

- The normal resource limit should be identical to the max resource limit; otherwise, the P3 schedule may not be identical to the RCPM schedule.
- All resources are selected for limited resources prior to RCS (leveling) in P3.
- The *level* prioritization in the P3 must be set in the order of "Late start, Remaining duration, Total float, and Activity ID".
- *Retained logic* scheduling option should be selected for progressed schedule. If *progress override* option is selected, the P3 schedule may not create an expected schedule in a progressed schedule.

Chapter 6

RCPM Evaluation

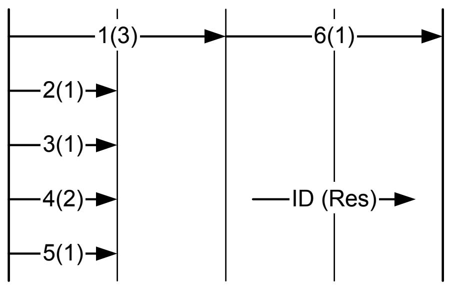
The proposed RCPM algorithm has been evaluated by comparing with other three algorithms proposed by Wiest (1964), Woodworth (1988), and Bowers (1995), respectively. The output of each sample schedule provided in the articles or generated based on those algorithms is compared to the RCPM output. This evaluation focuses on the feasibility of identified resource links, existence of required resource links, and the amount of floats if the early times of the resource-constrained schedule are the same. For alternative schedules, the RCPM output is compared with Bowers (2000).

In addition, RCPM has been tested with various schedules, including several real construction projects. These tests focus on whether the RCPM system generates an identical RCS output as in P3 and whether sufficient resource links are identified to enable RCPM scheduling with P3.

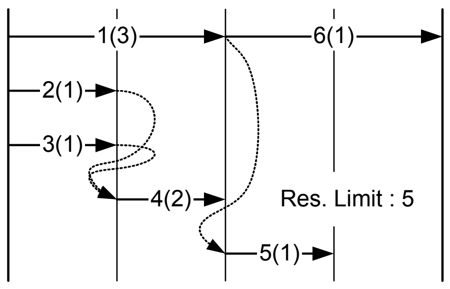
6.1 Comparison with Previous Algorithms of Identifying Floats

6.1.1 Wiest's Algorithm

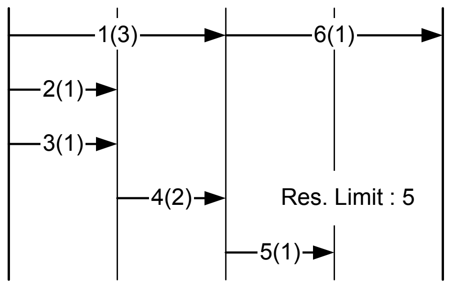
The schedule shown in Figure 6.1(a) is used for the comparison. With resource availability of 5 units, the RCPM-generated schedule is shown in Figure 6.1 (b). In addition, RCPM generates alternative schedules for *activity 2* and *3*. Either or both of them can be scheduled for day 4. This RCPM-generated schedule (Figure 6.1 (b)) without resource links is used for Wiest's left-justified schedule as shown in Figure 6.1 (c), because there is no specific RCS technique provided in Wiest (1964). Figure 6.1 (d) is the right-justified schedule from Figure 6.1 (c). Again, no activity in the left-justified schedule can start earlier by left shifting it, without violating technological relationships and resource constraints. Similarly, no activity in the right-justified schedule can start later by right shifting it. Table 6.1 shows total and free floats of each method.



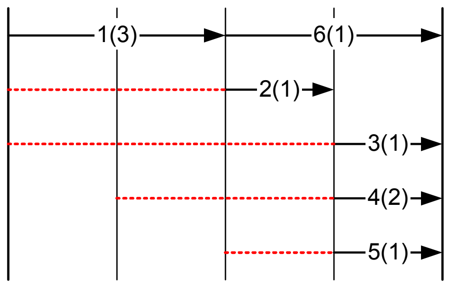
(a) initial schedule (no resource limit)



(b) RCPM result



(c) Wiest's left-justified schedule



(d) Wiest's right-justified schedule

Figure 6.1 Comparison with Wiest's Algorithm

Table 6.1 Float Comparison with Wiest's Algorithm

ID	RCPM		Wiest's	
	TF	FF	TF	FF
1	0	0	0	
2	2	0	2	
3	2	0	3	N/A
4	2	2	2	
5	1	1	1	
6	0	0	0	

The main differences between two methods are the existence of resource links. RCPM provides resource links, while Wiest's does not. Through resource links, the RCPM schedule provides more useful information than Wiest's. First, the RCPM schedule shows how total floats are shared by activities on a path through the technological and the resource links, while Wiest's does not. There is no resource availability violation for the TF periods of each activity in the RCPM schedule, while Wiest's has possible violations on day 2 and 3 because of no information on the shared total floats among activities. Second, the alternative schedules in the RCPM schedules can give more flexibility, compensating the less TF in *activity 3*. Finally, the RCPM schedule shows activity free floats, again through the technical and the resource links, while Wiest's does not.

On the other hand, a disadvantage of RCPM is that it does not show the exact resource constraint condition with schedule changes. For instance from Figure 6.1 (b), when the finish time of *activity 1* is delayed by one day, there is no need to delay the start time of *activity 5*, which should be delayed only if both *activity 1* and *4* are delayed.

This disadvantage results from lack of dynamic features in RCPM with schedule changes. RCPM mainly focus on the early times of activities in creating resource links and generates at least one resource link by a priority order (resource requirement) in step 2, when multiple activities are involved. A resource link is created even if the resource requirement of two resource-linked activities is less than the maximum availability. Otherwise representing the delayed activity in P3 is not available within the current RCPM algorithm. For instance from Figure 6.1 (b), if the resource link between *activity 1* and *5* does not exist, then the early start time of *activity 5* in the P3 schedule after adding resource links will be the beginning of day 1, which does not correctly represent the schedule.

In addition, all resource links in RCPM is currently permanent, but some resource links should be temporal at the initial schedule. For instance, the resource link between *activity 1* and *5* should be temporal. Then, if *activity 4* is delayed later, the temporal link will become permanent. If *activity 1*, on the other hand, is delayed later, then the temporal link between *activity 1* and *5* will be removed and a new permanent link between *activity 4* and *5* will be created. More research on this dynamic feature in resource links is required.

6.1.2 Comparison with Woodworth's Algorithm

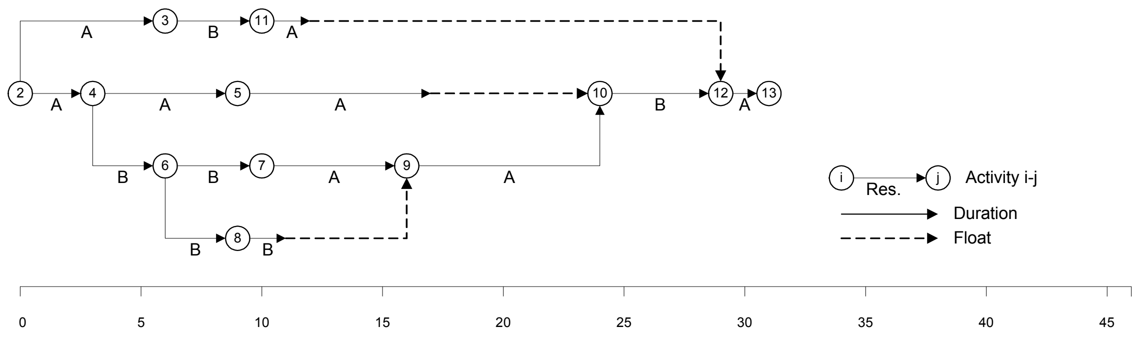
As shown in Table 6.2, the same schedule introduced in Chapter 2 is used for the comparison. This schedule is provided in Woodworth and Shanahan (1988). Figure 6.2(a) shows the time-scaled CPM. With resource availability of 1 unit for each resource *A* and *B*, the RCPM-generated schedule is shown in Figure 6.2 (b), and Woodworth's result is in Figure 6.2 (c).

In Woodworth's, a parallel RCS method is applied to resource constrained scheduling, while a serial RCS method is applied in RCPM. In addition, there is no priority rule provided when activities compete for a limited resource in this parallel method. Instead, any activity can be selected according to user's decision without consistent criteria. Thus, the sequences of activities in Figure 6.2 (b) and 6.3(c) are different and thus resource links are different resulting different completion time. Table 6.3 shows overall comparison between two schedules.

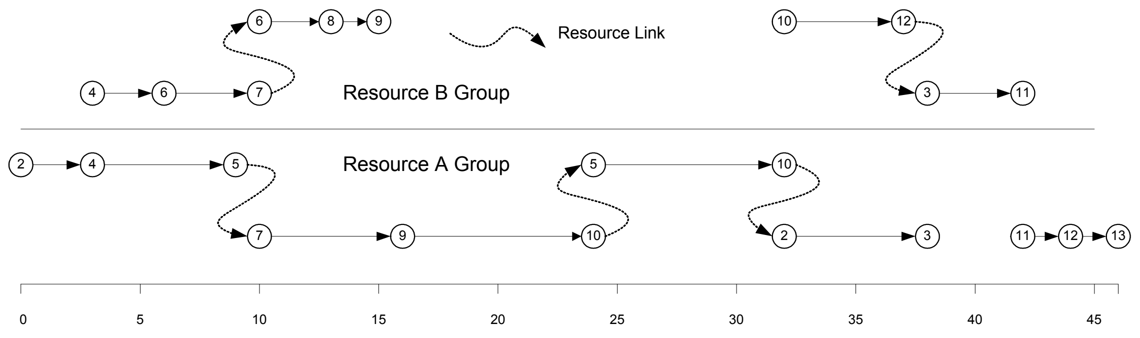
Table 6.2 Activity Data in the Woodworth's Example[1]

Activity	Duration	Resource	Quantity
2-3	6	A	1
2-4	3	A	1
3-11	4	B	1
4-5	6	A	1
4-6	3	B	1
5-10	8	A	1
6-7	4	B	1
6-8	3	B	1
7-9	6	A	1
8-9	2	B	1
9-10	8	A	1
10-12	5	B	1
11-12	2	A	1
12-13	2	A	1

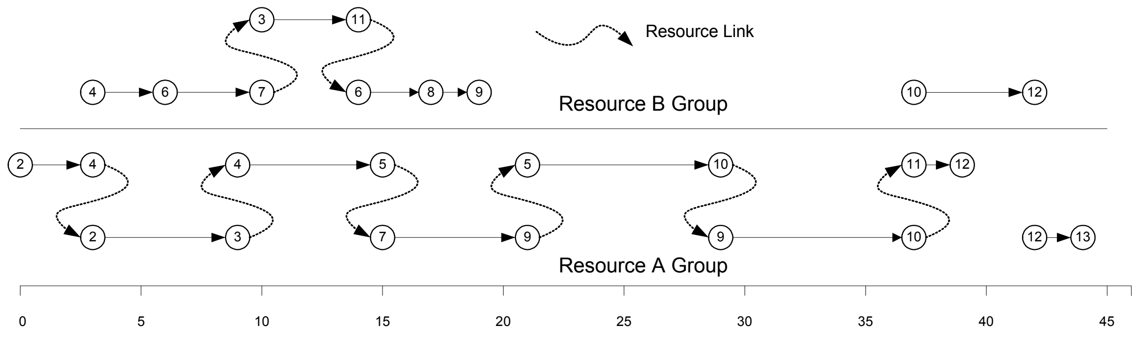
[1] Reprinted from International Journal of Project Management, Vol. 6 (2), Woodworth, B. M., and Shanahan, S., "Identifying the critical sequence in a resource constrained project", 89-96, 1988, with permission from Elsevier



(a) initial schedule (no resource limit)



(b) RCPM result



(c) Woodworth's result

Figure 6.2 Comparison with Woodworth's Algorithm

Table 6.3 Comparison with Woodworth's Algorithm

Comparison Item	Woodworth's	RCPM
Project Duration	44	46
# of resource links	8	5
Total Float	10 days in one chain (4-6, 6-7, 3-11, 6-8, 8-9) 3 days for activity 11-12	1 day for activity 4-5, 1 day for activity 6-8, 1 day for activity 8-9, 1 day for activity 10-12
Free Float	10 days for activity 8-9, 3 days for activity 11-12	1 day for activity 4-5, 1 day for activity 8-9, 1 day for activity 10-12

Since the resource constrained scheduling techniques applied are different, there are no important issues in this comparison. However, this comparison shows how a scheduling technique can affect a schedule in terms of the activity sequence, resource links, and then floats. Nevertheless, an identical schedule shown in Figure 6.2 (c) would be created, if RCPM adopted Woodworth's parallel RCS method. In this case, only step 2 would be required because every resource link in Figure 6.2 (c) has been created when an activity was delayed due to the resource availability.

Although Woodworth's performs well with the given example, which is simple in terms of resource requirement (1 unit for all activities) and availability (1 unit for all resources), there is no detailed procedure provided for handling multiple resources with multiple activities like Figure 6.1 (b) schedule. Based on the given procedure in Woodworth and Shanahan (1988), *activity 4* may have resource links with *activity 5* and *6*, which are unnecessary in that schedule. Figure 6.3 shows the resource index condition of Figure 6.1 (b) schedule.

A search from *activity 6* may find *activity 1* and *4* as its predecessors because they are immediate predecessors based on the resource index. Then, a resource link between *activity 4* and *6* will be created; no resource link is required between *activity 1* and *6* since *activity 1* is a predecessor of *activity 6*. Similarly, a search from *activity 5* may also find *activity 1* and *4* as its predecessor, making a resource link between *activity 1* and *5 and activity 4* and *5*. Detailed information about Woodworth's algorithm is discussed in Chapter 2.

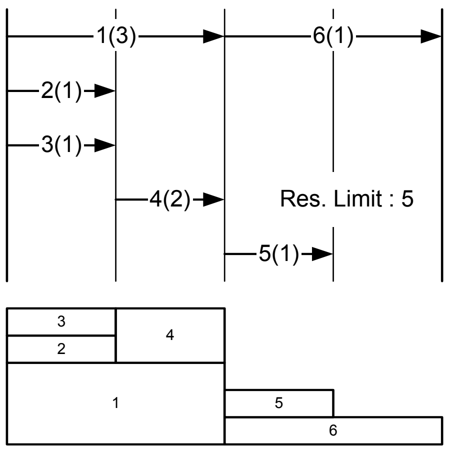
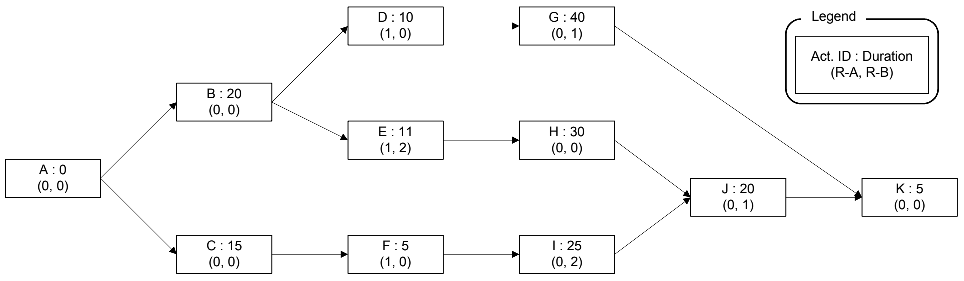


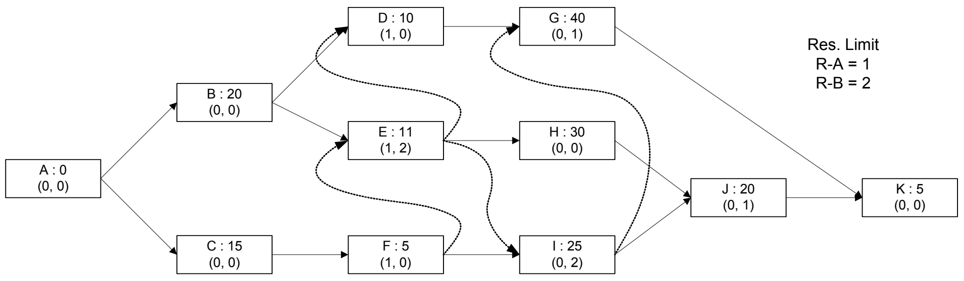
Figure 6.3 Resource Index of Figure 6.1 Schedule

6.1.3 Comparison with Bowers' Algorithm

Two examples provided in Bowers (1995) are used in comparison. The first example is relatively simple with 11 activities as shown in Figure 6.4 (a). Although Bowers' applies a parallel method while RCPM does a serial method, both methods coincidently generate the same resource-constrained schedule for this example and then identical resource links as shown in Figure 6.4 (b).



(a) Initial schedule (no resource limit)



(b) RCPM and Bowers' result *[p. 82, Bowers 1995]*

Figure 6.4 Comparison with Bowers' Algorithm: Example-I

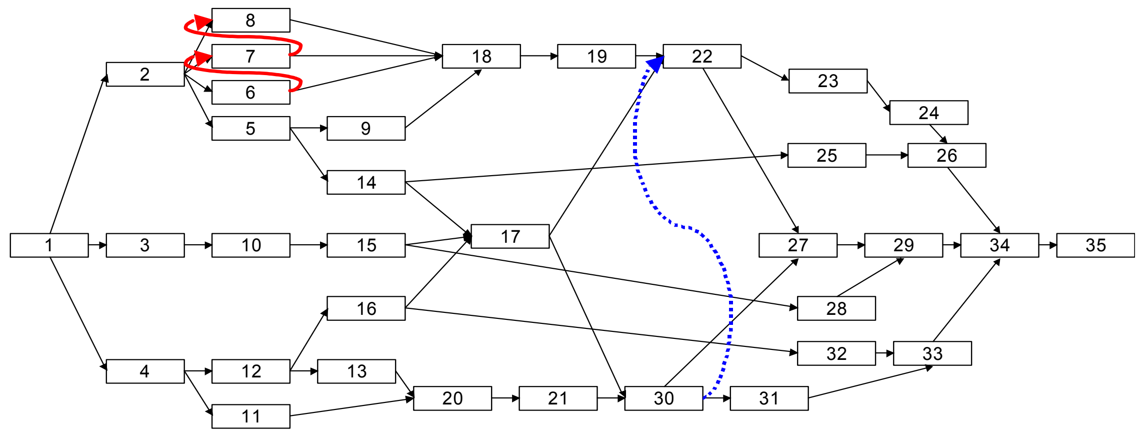
The other example schedule provided is in Table 6.4. Figure 6.5 (a) shows the RCPM schedule, and Figure 6.5 (b) shows Bowers'. Like the first example, both Bowers' and RCPM create the same activity sequence because the density of resource constraint of the schedule is relatively simple as shown in Table 6.4. Although there are several differences for *activity 5 ~ 9*, these can be disregarded because each of them has the identical information on duration and resource requirement.

Bowers', however, creates unnecessary resource links. Two unnecessary resource links are created between *activity 11* and *13*, and *activity 22* and *23*. As discussed in Chapter 2, the resource link between *activity 11* and *13* is not required because of the technological link between *activity 12* and *13*. With the unnecessary resource link, the total float of *activity 11* is 3 days in Bowers', while 15 days in RCPM. There is no total float difference for *activity 22* since the technological link between *activity 22* and *23* overlaps the resource link. These two unnecessary resource links leads to the conclusion that Bowers' does not consider technological relationships while identifying resource links.

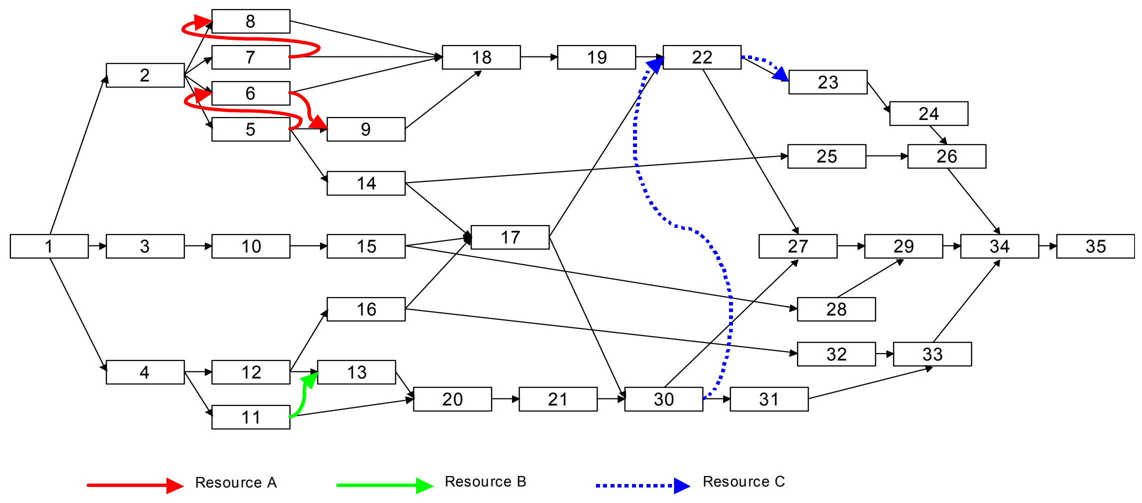
Table 6.4 Bowers' Aircraft Schedule Data: Example-II

Activity ID	Activity description	Dur.	Successors	Resource		
				A	B	C
1	General Design	12	2, 3, 4	0	0	0
2	Avionics design	12	5, 6, 7, 8	0	0	0
3	Airframe design	9	10	0	0	0
4	Engin design	18	11, 12	0	0	0
5	Avionics development (1)	12	9, 14	**1**	0	0
6	Avionics development (2)	12	18	**1**	0	0
7	Avionics development (3)	12	18	**1**	0	0
8	Avionics development (4)	12	18	**1**	0	0
9	Avionics development (5)	12	18	**1**	0	0
10	Airframe development	9	15	0	0	0
11	Engine development (1)	12	20	0	**1**	0
12	Engine development (2)	12	13, 16	0	**1**	0
13	Engine development (3)	12	20	0	**1**	0
14	Manf. d/b avionics	12	17, 25	0	0	0
15	Manf. d/b airframe	12	17, 28	0	0	0
16	Manf. d/b engine	12	17, 32	0	0	0
17	Integrate d/b aircraft	12	22, 30	0	0	0
18	Avionics integration	6	19	0	0	0
19	Avionics land trials	9	22	0	0	0
20	Engine integration	6	21	0	0	0
21	Engine land trials	6	30	0	0	0
22	Avionics trials (1)	9	23	0	0	**1**
23	Avionics trials (2)	6	24	0	0	**1**
24	Revise avionics	9	26	0	0	0
25	Prepare avionics prodn	18	26	0	0	0
26	First production avionics	12	34	0	0	0
27	Revise airframe	6	29	0	0	0
28	prepare airframe prodn	18	29	0	0	0
29	First production airframe	9	34	0	0	0
30	Engine trials	6	31	0	0	**1**
31	Revise engine	9	33	0	0	0
32	Prepare engine prodn	18	33	0	0	0
33	First engine production	12	34	0	0	0
34	Integrate prodn aircraft	12	35	0	0	0
35	Flight of prodn aircraft	3		0	0	0
Resource availability				2	2	1

[p. 85, Bowers 1995]



(a) RCPM result



(b) Bowers' result

Figure 6.5 Comparison with Bowers' Algorithm: Example-II
[reproduced from the initial AON network from p. 84, Bowers 1995]

6.1.4 Comparison Summary of Identifying Floats

Table 6.5 shows a summary of the algorithm comparison for identifying floats in resource-constrained schedules. The Wiest's algorithm does not generate resource links, so the float sharing condition, the effect of schedule changes, and free floats are not available. The Woodworth's algorithm does not provide a process to handle multiple resources with multiple parallel activities, so activities could create unnecessary resource links. Bowers' algorithm does not consider technological relationships when creating resource links, and it is not explicit enough to determine whether Bowers' algorithm can handle multiple resources with multiple activities due to lack of description in Bowers (1995). RCPM performs well compared to these three algorithms. However, RCPM, like the other algorithms, does not provide the dynamic feature of resource links after schedule changes.

Table 6.5 Summary of Algorithm Comparison

	Wiest's	Woodworth's	Bowers'	RCPM
Existence of Res. Link	X	O	O	O
Multi. Res. w/ Multi. Act.	O	X	N/A	O
RL considering Tech. Link	N/A	O	X	O
Dynamic Feature of RL	N/A	X	X	X

O : Yes, X : No

6.2 Comparison with Bowers' Alternative Schedules

Bowers (2000) employs the same aircraft schedule introduced in Bowers (1995) to demonstrate alternative schedules or scheduling flexibility. Table 6.6[2] shows activities that can be scheduled in different periods from the initial schedule with the same project completion time. Six among thirty five activities could have alternative schedules.

Table 6.6 Alternative Schedules of Activities with Flexible Floats

ID	Activity description	Dur.	initial schedule		float				
			ES	LS	(1)	(2)	(3)	EES	LLS
5	Avionics development (1)	12	24	24	0	9	9	24	33
6	Avionics development (2)	12	36	36	0	12	24	24	48
7	Avionics development (3)	12	24	36	12	12	24	24	48
8	Avionics development (4)	12	36	48	12	12	24	24	48
9	Avionics development (5)	12	48	48	0	12	12	36	48
11	Engine development (1)	12	30	33	3	12	15	30	45

[reproduced from the original table from p. 860, Bowers 2000]

[2] Detailed information of each value is provided in the review of Bowers (2000) in Chapter 2.

Activity 5, 7, and *11* can be scheduled later than the initial late start time since latest late start time (LLS) > initial late start time (LS). *Activity 8* and *9* can be scheduled earlier than the initial early start time since earliest early start time (EES) < initial early start time (ES). *Activity 6* can be scheduled earlier than the initial early start time or later than the initial late start time, since EES < ES and LLS > LS.

These activity data provide a baseline to reorganize the initial schedule (Figure 6.6) when an unexpected event happens, resulting in delays of some critical activities. For example, if *activity 5* (initially critical) should be delayed, a best-fit schedule that has 33 for its LLS can be selected after processing 140 (= 4*35) resource-constrained schedules, and then there will be no completion time extension.

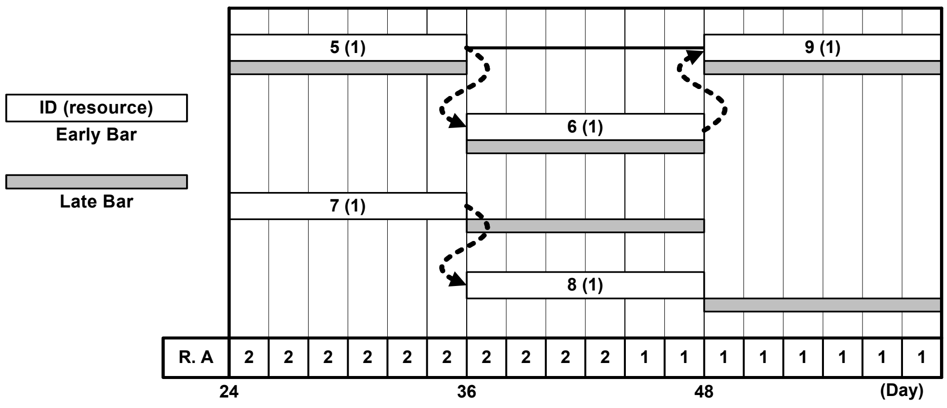


Figure 6.6 Initial Schedule of Activities with Alternative Schedules in Bowers'

With the same schedule, RCPM has identified two alternative schedules as shown in Table 6.7. *Activity 6* is a critical activity in the initial schedule (Figure 6.7), but it can be scheduled over its LFT; the period of 36 ~ 48 is available since *activity 9* can be delayed within its TF range, and the period 48 ~ 60 is available since current daily resource requirement is one. *Activity 7* is also a critical activity in the initial schedule, but it can also be scheduled over its LFT; the period 48 ~ 60 is available since current daily resource requirement is one. However, both cannot be scheduled for the period 48 ~ 60. If *activity 8* starts earlier than the initial schedule, both activities can be scheduled for the period, but currently RCPM does not consider this earlier start because it is beyond the scope of this study.

Table 6.7 Alternative Schedules of Activities in RCPM

ID	Activity description	Dur.	initial schedule		alt. period	
			ES	LS	Start	End
6	Avionics development (2)	12	24	24	36	60
7	Avionics development (3)	12	36	48	48	60

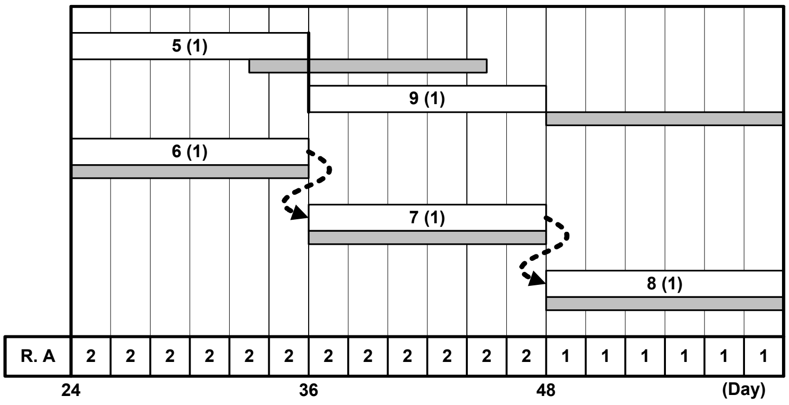


Figure 6.7 Initial Schedule of Activities with Alternative Schedules in RCPM

Bowers' allows an activity to start earlier than the EST from the initial schedule, so that it provides more exhaustive alternative schedules than RCPM. However, only one schedule can be selected in Bowers. Once one schedule is selected, the other data (EES and LLS) are no longer valid unless they are on the selected schedule. On the other hand, most alternative schedules in RCPM are available except some activities whose alternative schedule periods are overlapped by the same resources with resource overuse, like *activity 6* and *7* from Table 6.7 for the period 48 ~ 60.

6.3 Evaluation with Test Schedules

The RCPM system has been tested with a number of schedules that are obtained from several construction companies or generated for a test purpose. Table 6.8 shows test results with four notable schedules. *Schedule A* is from Fondahl (1991), *Schedule B* is created for a test purpose, and *Schedule C and D* are obtained from construction companies.

Table 6.8 Test Schedule Status and Result

ID	# of Act.	# of Calendar	# of Res. Type	Duration* (days)		# of Res. Link
				CPM	RCPM	
A	31	1	3	27	30	13
B	17	2	2	26	35	7
C	300	3	16	394	394	58
D	1141	3	38	1705	2062	363

* continuous working days including all weekends and holidays.

The criteria to test are i) whether the RCS outputs[3] (early times, late times, TF, and FF) from both P3 and the RCPM system are identical, ii) if there are resource-overuses after running P3 CPM with added resource links, and iii) whether P3 CPM output (after resource links are incorporated) and the initial RCPM output are identical. The test results on these criteria are summarized in Table 6.9.

Table 6.9 Results on Test Criteria

ID	Test Criteria		
	I	II	III
A	O	O	O
B	O	△	△
C	O	△	△
D	O	O	O

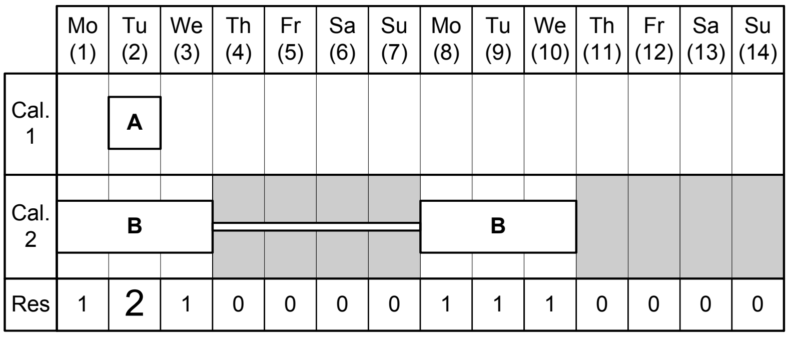
O : 100% satisfied
△ : most data are satisfying, but few discrepancies exist

The RCPM system performs well with all these projects. First, early times, late times, TF and FF of RCPM-generated schedules on both CPM and RCS are exactly matching with those of P3 output. Second, the CPM in P3, after adding identified resource links into the original P3 project, performs as anticipated for most activities. Few activities

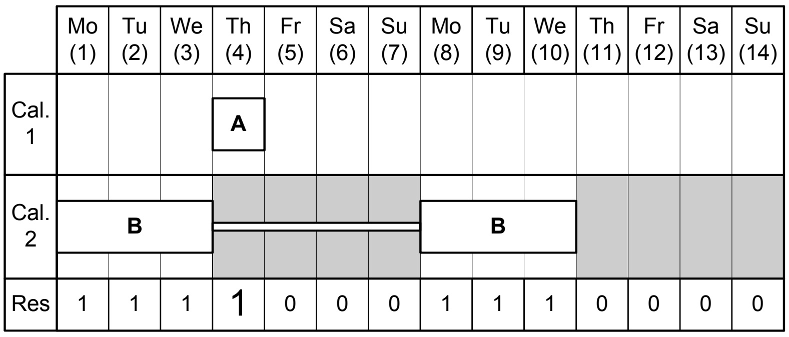
[3] As described in Chapter 5, the RCPM system generates three scheduling outputs: CPM, RCS, and RCPM. The outputs of RCS and RCPM are resource-constrained schedules and have identical early start/finish times for each activity. However, their late start/finish times are different because the RCPM schedule considers both technological and resource links, but the RCS schedule considers the technological links only. P3 and the RCPM system should generate identical schedules for RCS, since they employ the same serial resource constrained scheduling technique and P3 does not create resource links.

have slightly different data due to the characteristics of scheduling with multiple calendars. Figure 6.8 shows a simplified example of that case.

In this schedule, each activity requires one resource unit and the limit is one. Figure 6.8 (a) is an initial CPM schedule (all other activities are omitted), and Figure 6.8 (b) is the RCPM output. Currently, RCPM identifies only FS:0 type resource links, so that no resource link is created. After adding all identified resource links, P3 CPM generates Figure 6.8 (a) condition, since no resource link into *activity A* exists.



(a) initial CPM schedule



(b) RCPM output

Figure 6.8 Resource Link not Identified in RCPM

Except this special condition, there is no resource overuse in the CPM in P3. Finally, the initial RCPM and P3 CPM with resource links generate identical schedules except the same special condition.

At the beginning this study, there were a number of mismatched early times in the *schedule C* and *D*, but they are solved by removing various constraints and options of P3.

Those removed constraints are hammock activity, early/late start/finish, mandatory start/finish, and immediate priority in the leveling type of each activity. In addition, changeable resource limit are removed, and every resource normal limit is set to equal to its max limit.

Chapter 7

Case Studies

Two sample schedules are explored in detail to evaluate the RCPM system. One schedule has only finish-to-start with a zero lag and a continuous working day calendar without any holidays or non-working days during the project period. With this schedule, a change order condition at the beginning the project is examined. The other schedule, on the other hand, has all four Precedence Diagramming Method (PDM) relationships and two independent project calendars. With this schedule, a progressed schedule is examined.

7.1 Single Calendar Schedule Example

7.1.1 Original Schedule

A warehouse project schedule from Fondahl (1991), obtained through personal communication [Melin 1995], is used for this case study. Table 7.1 shows the initial activity data. There is a maximum of four resource units available per day for each of the three resource types. The AON network of this schedule is shown in Figure 7.1. A 27-day CPM schedule in P3 is shown in Figure 7.2 in which the periods that overrun resource limits are underlined at the bottom of the chart. Figure 7.3 shows a 30-day schedule, after resource constrained scheduling is applied in P3. There is no resource over-run, but the critical path is broken (only the last two activities are critical) and most TF values contain phantom floats.

Table 7.1 Case-I Schedule Activity Data

ID	Dur.	Resources			Successors
		Carp.	Labor	Iron Work	
1	1				2
2	3				3,4,5,6,7,8
3	2	4			29
4	2	2			19
5	4	4			15
6	1		4		9, 10
7	2	2			24
8	3	4			22
9	1	4	2	2	11
10	1				14
11	1	1	2		12
12	1	1	1		12a
12a	3				14,13
13	1			4	15, 18
14	2		2		16
15	1			4	17
16	1	2		2	18
17	1	2		2	19
18	1	1	2		20
19	1			4	21
20	6				22, 29
21	1			4	23
22	2	4			24, 25
23	2	2			26
24	1	2			27
25	1		2		27
26	1		2		28
27	2				30
28	2	2			30
29	1	4			30
30	1		4		
Max. Resource		4	4	4	

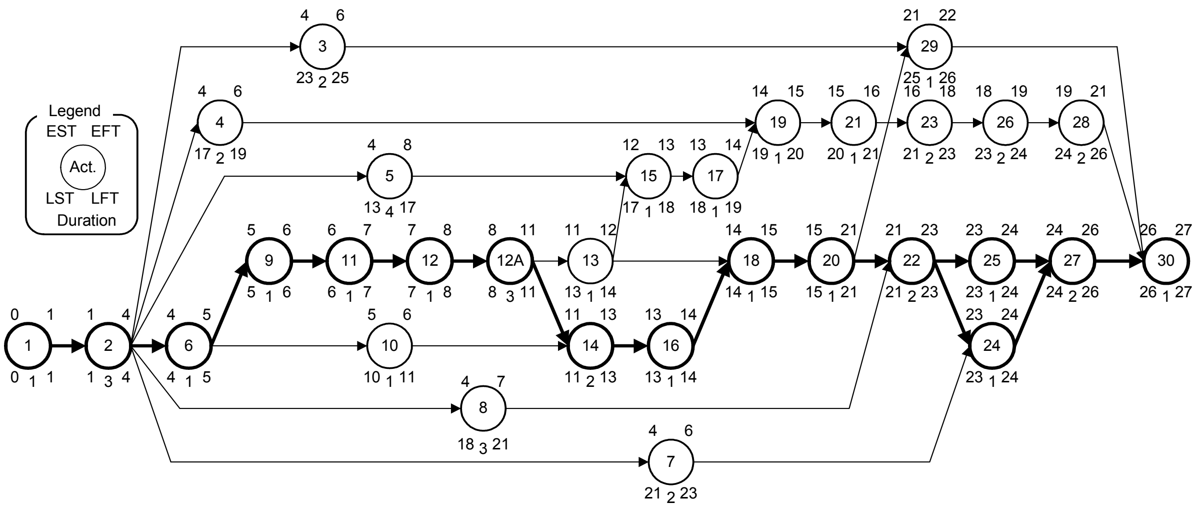
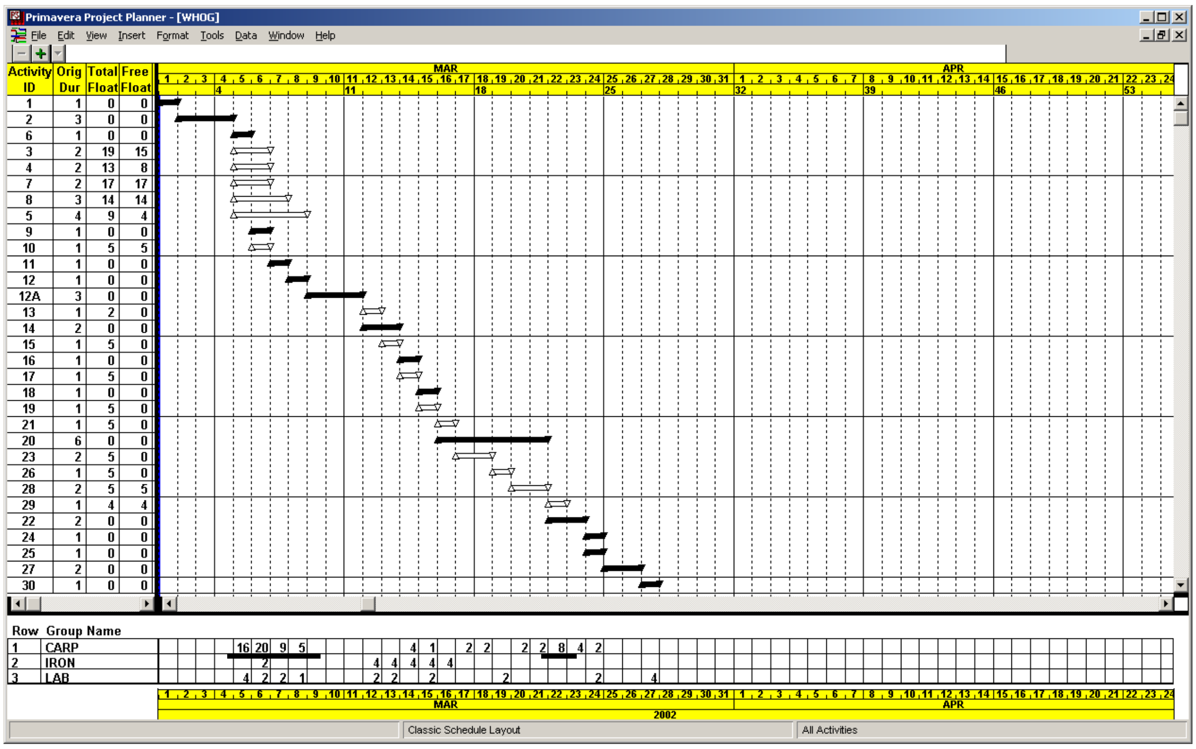


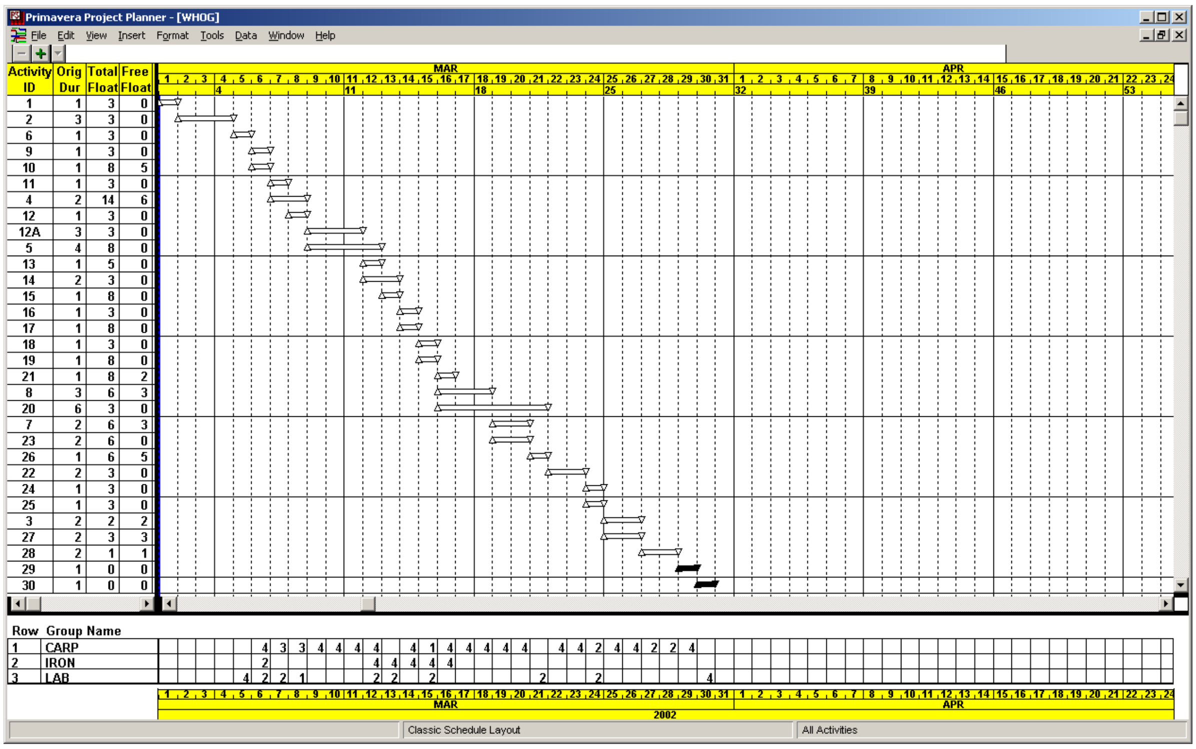
Figure 7.1 CPM Network of Case-I Schedule


Primavera Project Planner - [WHOG]
File Edit View Insert Format Tools Data Window Help
MAR
APR
2002
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float	Free Float
1	1	0	0
2	3	0	0
6	1	0	0
3	2	19	15
4	2	13	8
7	2	17	17
8	3	14	14
5	4	9	4
9	1	0	0
10	1	5	5
11	1	0	0
12	1	0	0
12A	3	0	0
13	1	2	0
14	2	0	0
15	1	5	0
16	1	0	0
17	1	5	0
18	1	0	0
19	1	5	0
21	1	5	0
20	6	0	0
23	2	5	0
26	1	5	0
28	2	5	5
29	1	4	4
22	2	0	0
24	1	0	0
25	1	0	0
27	2	0	0
30	1	0	0

Row	Group Name
1	CARP
2	IRON
3	LAB

Figure 7.2 CPM Output of Case-I Schedule in P3



Primavera Project Planner - [WHOG]
File Edit View Insert Format Tools Data Window Help
MAR
APR
2002
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float	Free Float
1	1	3	0
2	3	3	0
6	1	3	0
9	1	3	0
10	1	8	5
11	1	3	0
4	2	14	6
12	1	3	0
12A	3	3	0
5	4	8	0
13	1	5	0
14	2	3	0
15	1	8	0
16	1	3	0
17	1	8	0
18	1	3	0
19	1	8	0
21	1	8	2
8	3	6	3
20	6	3	0
7	2	6	3
23	2	6	0
26	1	6	5
22	2	3	0
24	1	3	0
25	1	3	0
3	2	2	2
27	2	3	3
28	2	1	1
29	1	0	0
30	1	0	0

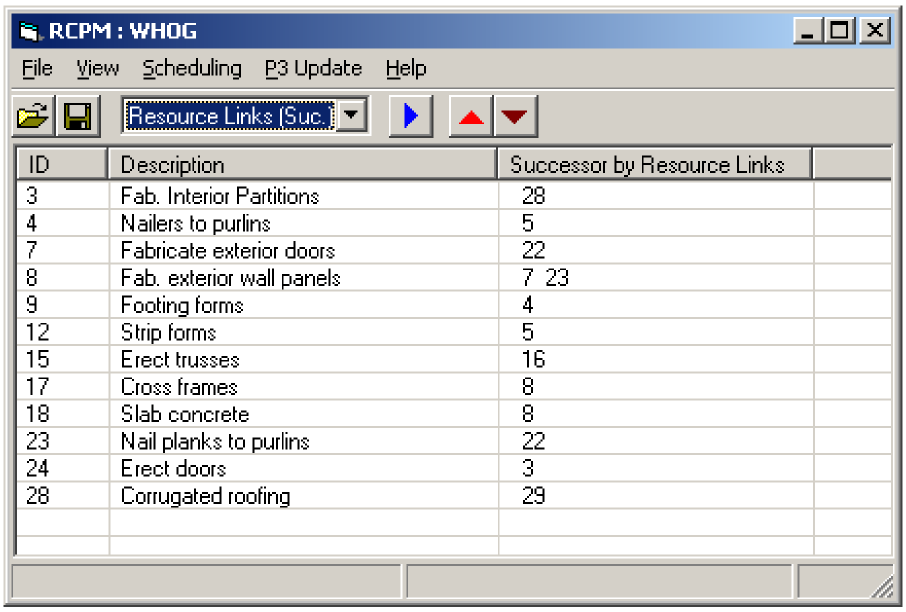
Row	Group Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	CARP						4	3	3	4	4	4	4		4	1	4	4	4	4	4		4	4	2	4	4	2	2	4		
2	IRON						2						4	4	4	4	4															
3	LAB					4	2	2	1				2	2		2						2			2						4	

Figure 7.3 RCS Output of Case-I Schedule in P3

Once the RCPM technique is applied, resource links can be identified. Figure 7.4 shows a screen copy of these identified resource links in the RCPM system, and Figure 7.5 shows the modified AON network with the resource links. The resource links are automatically incorporated into the initial P3 schedule from the RCPM system upon user's request, and then the P3 schedule becomes an RCPM schedule. From now, the CPM in P3, [Tools | Schedule], becomes RCPM. After running RCPM, the Phantom Float is removed and the critical path is continuous from the start through the end of the project as shown in Figure 7.6. The early times of the RCPM schedule is exactly same as the output of P3 RCS (Figure 7.3) resulting in a 30-day project period with the same work sequence.

In addition to resource links, RCPM identifies alternative schedules for some activities, which can provide more flexibility to the schedule when unexpected events happen on these activities. Figure 7.7 shows alternative schedules. Each activity can have alternative period beyond its LFT. For instance, *activity 4* requires 2 carpenters per day and has 8 March of LFT, but it can have an alternative schedule for 13 ~ 16 March as marked in Figure 7.6. The total daily requirement of carpenters is 0, 4, 1, and 4 for 13, 14, 15 and 16

March, respectively. On 14 and 16 March, the total daily requirements are already maximum (4). However, *activity 17* that requires 2 carpenters on 14 March has a 2-day TF, and *activity 8* that requires 4 carpenters on 16 March has 1-day TF. Both activities can be delayed making the alternative periods available for *activity 4*.



ID	Description	Successor by Resource Links
3	Fab. Interior Partitions	28
4	Nailers to purlins	5
7	Fabricate exterior doors	22
8	Fab. exterior wall panels	7 23
9	Footing forms	4
12	Strip forms	5
15	Erect trusses	16
17	Cross frames	8
18	Slab concrete	8
23	Nail planks to purlins	22
24	Erect doors	3
28	Corrugated roofing	29

Figure 7.4 Resource Links in Case-I Schedule Identified by RCPM System

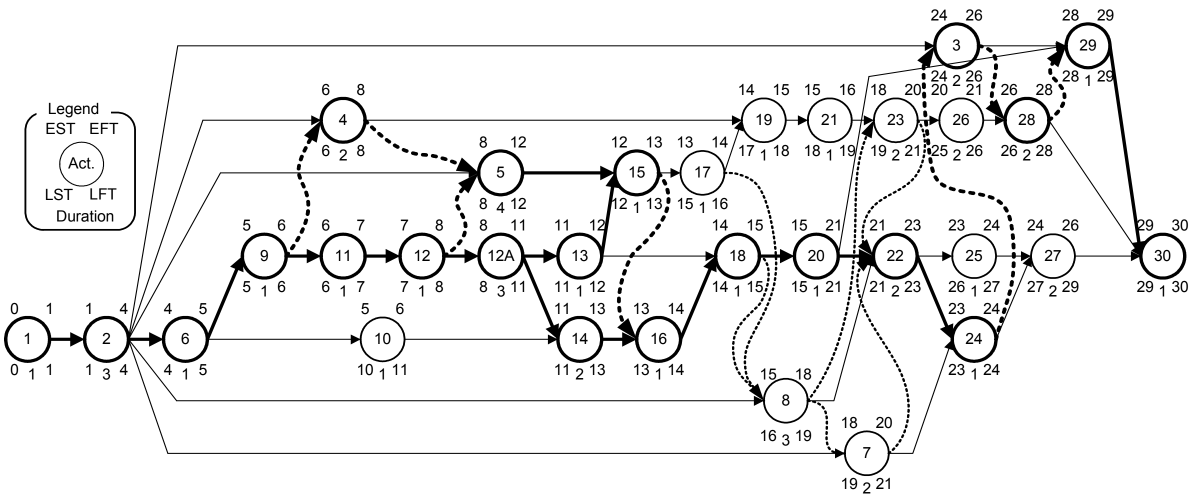
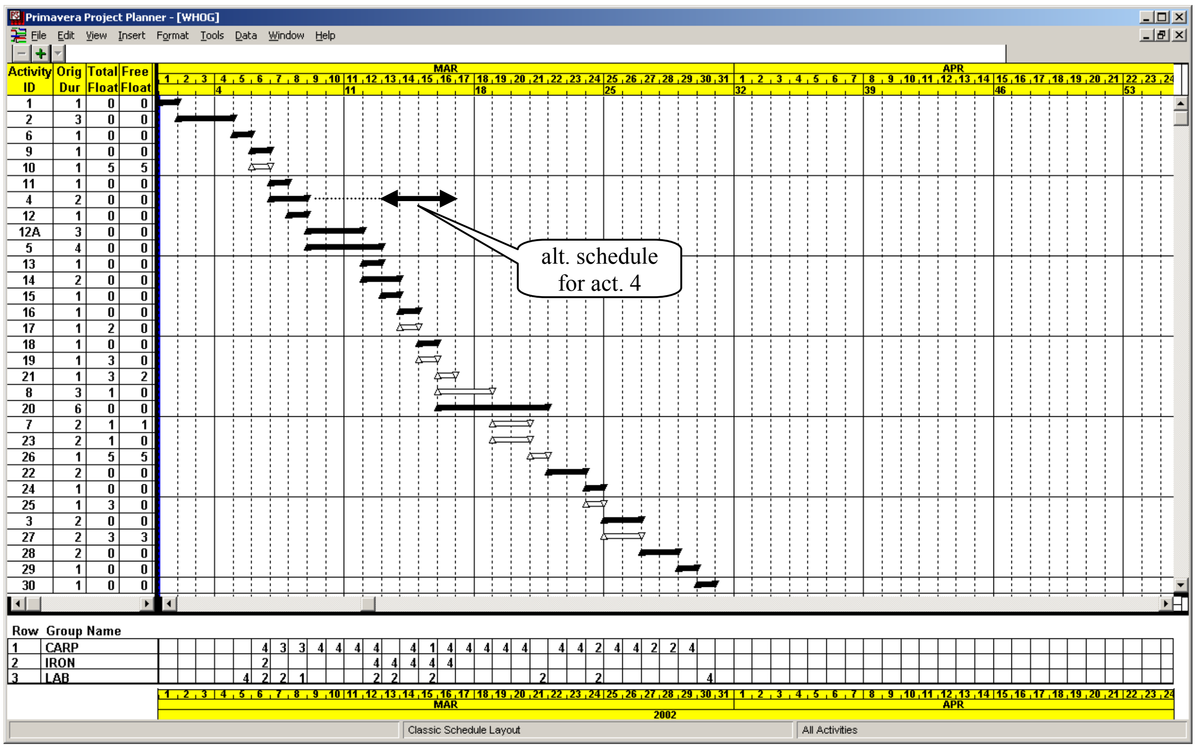


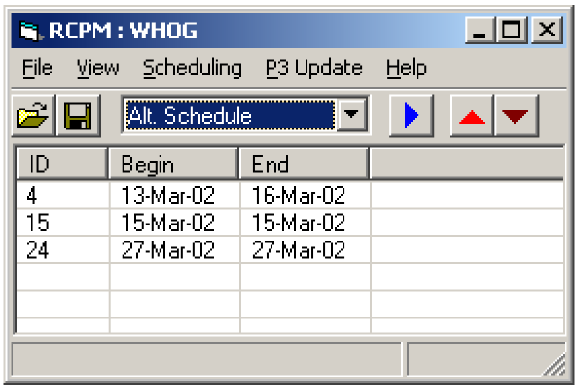
Figure 7.5 RCPM Network of Case-I Schedule


Primavera Project Planner - [WHOG]
File Edit View Insert Format Tools Data Window Help
MAR
APR
2002
alt. schedule for act. 4
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float	Free Float
1	1	0	0
2	3	0	0
6	1	0	0
9	1	0	0
10	1	5	5
11	1	0	0
4	2	0	0
12	1	0	0
12A	3	0	0
5	4	0	0
13	1	0	0
14	2	0	0
15	1	0	0
16	1	0	0
17	1	2	0
18	1	0	0
19	1	3	0
21	1	3	2
8	3	1	0
20	6	0	0
7	2	1	1
23	2	1	0
26	1	5	5
22	2	0	0
24	1	0	0
25	1	3	0
3	2	0	0
27	2	3	3
28	2	0	0
29	1	0	0
30	1	0	0

Row	Group Name
1	CARP
2	IRON
3	LAB

Figure 7.6 RCPM Output of Case-I Schedule in P3



ID	Begin	End
4	13-Mar-02	16-Mar-02
15	15-Mar-02	15-Mar-02
24	27-Mar-02	27-Mar-02

Figure 7.7 Alternative Schedules in Case-I Schedule from RCPM

7.1.2 Change Order

It is assumed that at the time of notice to proceed, the owner has informed the contractor that several components have been redesigned. With this change order, there is no network logic change, but the estimated durations of 4 affected activities have been increased with the same daily resource requirement as shown in Table 7.2. In the original

CPM network, *activity 3* is a predecessor of *activity 29*, but there is no technological relationship among *activity 3*, *7*, and *8* (or *activity 29, 7,* and *8*) as shown in Figure 7.1. Two issues related to this time extension example are the estimated project completion time and the schedule stability.

Table 7.2 Time Extension of Case-I Schedule

Act. ID	Initial Dur.	Change Order Dur.	New Dur.
3	2	6	8
7	2	5	7
8	3	9	12
29	1	3	4

7.1.2.1 Estimated Completion Time

There are three schedule types used to estimate the project completion time as a result of the change order, namely, 1) resource limits are ignored (CPM); 2) resource limits are considered but the concept of Phantom Float is ignored (RCS in P3); and 3) both resource limits and Phantom Float are considered (RCPM).

Without re-scheduling after the change order, the revised project completion time for each of the three schedule types can be estimated by adopting widely used "float ownership" and "time extension" scheduling specification language, like:

> *"... total float is not time for the exclusive use or benefit of either the owner or the contractor ..."*

> *"... extensions of time will be granted only to the extent that excusable delays cause activities on the affected paths to exceed the available total float in effect at the time of the delay ..."*

In Table 7.3, column 4 shows the TF reported for each schedule type before the change order, column 7 shows the estimated project completion time after the change order but without re-scheduling, and column 8 shows the predicted project completion time after the change order but with re-scheduling.

Table 7.3 Project Completion Time with Change Order in Case-I Schedule

Schedule Type (1)	Init. Compl. Time (2)	Activ ity ID (3)	TF (4)	Change Order Dur. (5)	Effect (6)	Estimated Compl. Time (7)	Rescheduled Compl. Time (8)
CPM	27	3	19	6	0	27 (= 27 + 0)	27
		7	17	5	0		
		8	14	9	0		
		29	4	3	0		
RCS in P3	30	3	2	6	4	37 (= 30 + 4 + 3)	51
		7	6	5	0		
		8	6	9	3*		
		29	0	3	3		
RCPM	30	3	0	6	6	52 (= 30 + 8 + 5 + 6 + 3)	52
		7	1**	5	5		
		8	1**	9	8		
		29	0	3	3		

*Since *activity 7* and *8* are concurrent with *activity 3* and *29*, these delays are absorbed by 7-day delay by *activity 3* and *29*.

** As shown in Figure 7.5, all four activities are on one path (*8 – 7 – 3 – 29*) because of resource links, so that this one day TF is shared.

A comparison of columns 7 and 2 in Table 7.3 reveals that:

1) Contractors who ignore resource limits altogether would forecast a completion time of 27 days before and after the change order.
2) Contractors, who consider resource limits but ignore resource links, hence producing and disseminating schedules filled with Phantom Float, would forecast a completion time of 30 days before the change order and of 37 days after the change order. Therefore, the change order should yield a 7-day time extension.
3) Contractors who consider both resource limits and links would forecast a completion time of 30 days before the change order and of 52 days after the change order. Therefore, the change order should yield a 22-day time extension.

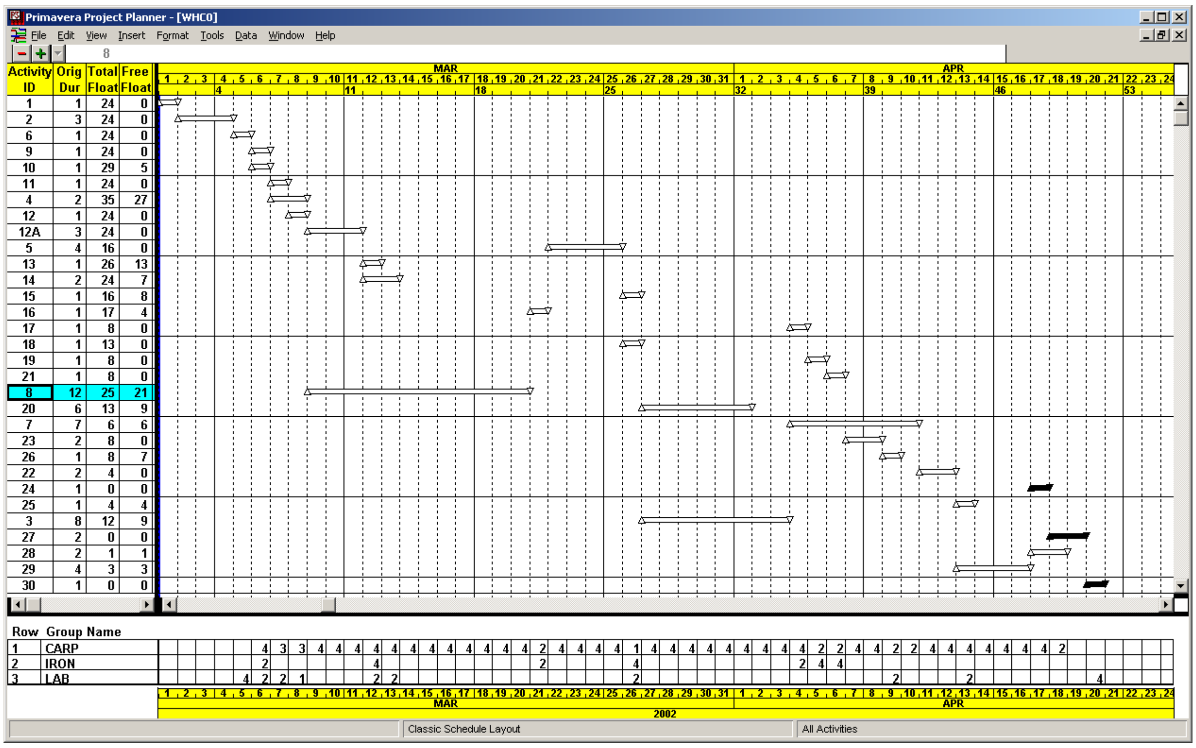
Rescheduling after the change order will readily compute a revised project completion time (column 8 in Table 7.3). A comparison of columns 7 and 8 in Table 7.3 reveals that:

1) Contractors who ignore resource limits altogether would still project a completion time of 27 days after the change order.
2) Contractors who consider resource limits but ignore resource links would project a completion time of 51 days. The difference between 51 days and 37 days is attributed to P3 determining at the time of re-scheduling that the TF calculated and reported in column 4 was never real to begin with—hence the term Phantom Float—and P3 is forced to extend the completion time an additional 14 days (51 – 37). There are numerous reasons why this scenario would yield a great deal of anxiety between the owner and the contractor. First, the project completion time should be the same with or without re-scheduling—why it isn't? It has now been shown that the reason these projections are not the same is due to the presence of Phantom Float in the TF reported by P3 prior to the change order. Second, why did the contractor submit a schedule prior to the change order containing Phantom Float within the TF? Claiming ignorance of its presence is not likely to suffice because any contractor who incorporates resource limits in the schedules is likely to earn the respect of the owner. It is this earned respect that will make owners trust the other information provided within the schedule, i.e., Phantom Float. Finally, the magnitude of the impact of the change order on the completion time will add to the anxiety.
4) Contractors who consider both resource limits and links would project a completion time of 52 days after the change order. There is no Phantom Float built into the schedule produced by RCPM, hence the values of TF in column 4 of Table 7.3 are real, which explains why the estimated completion time is the same with and without rescheduling—as it should be. The anxiety between the owner and the contractor will now be solely attributed to the magnitude of the impact.

This example shows how an unlimited resource CPM schedule is unrealistic in a resource-constrained condition and how Phantom Float could mislead project participants. Scheduling with limited resources (RCS and RCPM) after the change order has made the project duration nearly double compared to the CPM schedule. In addition, there is a significant difference in completion times between the estimated (column 7 in Table 7.3) and the rescheduled (column 8 in Table 7.3) for the RCS due to the presence of Phantom Float, while there is no difference in the RCPM schedule. This completion time difference in RCS could cause considerable disputes between the owner and the contractor if they do not acknowledge the impact of Phantom Float.

7.1.2.2 Schedule Stability

The RCPM result with the time extension could be preferable to the RCS result in this example, although its completion time is one day longer than RCS. This is because RCPM provides more stable schedule than P3 because of identified resource links in rescheduling. Figure 7.8 presents the rescheduled P3 output with the time extension. It is observed that the work sequence of about 30% of activities is changed. Those changed activities are *activity 5, 16, 17, 18, 8, 7, 23, 24, 3, 28,* and *29*. As mentioned earlier, this changed activity sequence occurs because the RCS technique in P3 employs certain priority rules (LST, TF, FF, etc.) and the updated activity data affect those activity priority orders, then generating a different work sequence. Reorganizing these activities may require additional cost and time. However, the work sequence in the RCPM schedule with fixed resource links is identical with the schedule before the time extension (see Figure 7.9). In general, a trade-off will be required to decide whether to go with RCPM with fixed resource links or with P3, based on the difference of completion times and various site conditions.


Primavera Project Planner - [WHC0]

File Edit View Insert Format Tools Data Window Help

8

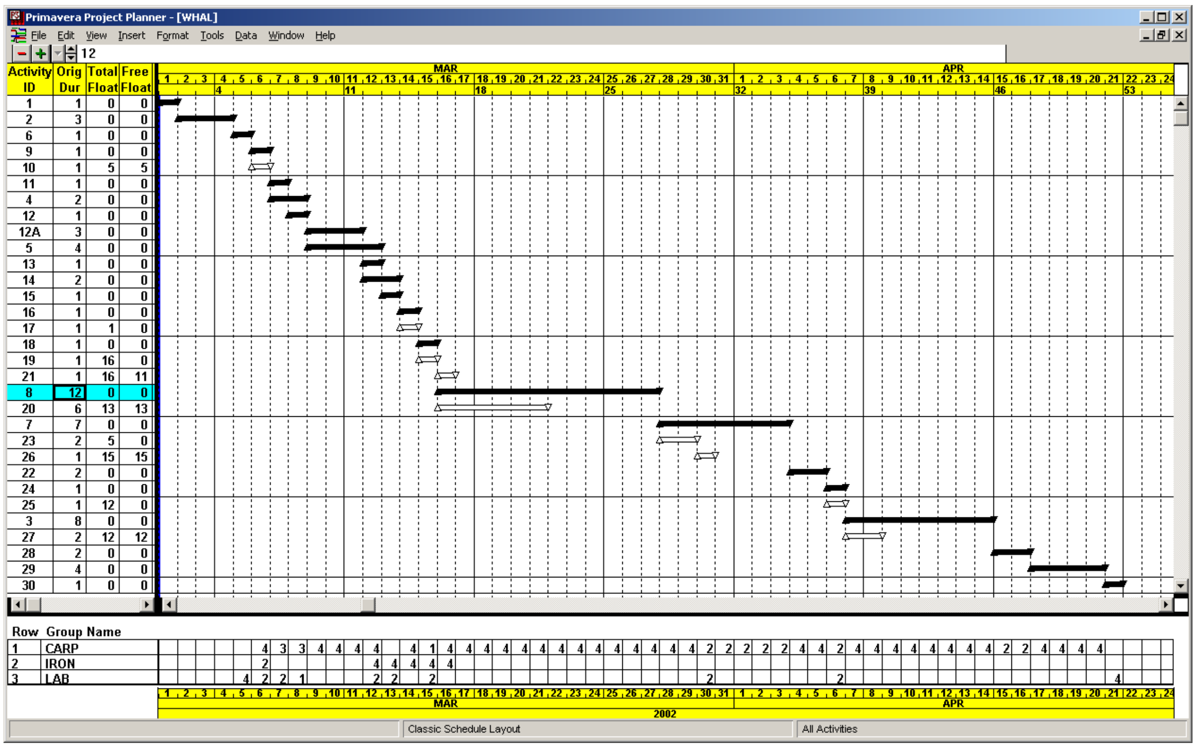
Activity ID	Orig Dur	Total Float	Free Float
1	1	24	0
2	3	24	0
6	1	24	0
9	1	24	0
10	1	29	5
11	1	24	0
4	2	35	27
12	1	24	0
12A	3	24	0
5	4	16	0
13	1	26	13
14	2	24	7
15	1	16	8
16	1	17	4
17	1	8	0
18	1	13	0
19	1	8	0
21	1	8	0
8	12	25	21
20	6	13	9
7	7	6	6
23	2	8	0
26	1	8	7
22	2	4	0
24	1	0	0
25	1	4	4
3	8	12	9
27	2	0	0
28	2	1	1
29	4	3	3
30	1	0	0

Row	Group Name	
1	CARP	4 3 3 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 1 4 4 4 4 4 4 4 4 4 2 2 4 4 2 2 4 4 4 4 4 4 4 2
2	IRON	2 4 2 4 2 4 4
3	LAB	4 2 2 1 2 2 2 2 2 4

MAR APR 2002

Classic Schedule Layout | All Activities

Figure 7.8 Rescheduled P3 Output w/o Resource Link in Case-I Schedule


Primavera Project Planner - [WHAL]

File Edit View Insert Format Tools Data Window Help

12

Activity ID	Orig Dur	Total Float	Free Float
1	1	0	0
2	3	0	0
6	1	0	0
9	1	0	0
10	1	5	5
11	1	0	0
4	2	0	0
12	1	0	0
12A	3	0	0
5	4	0	0
13	1	0	0
14	2	0	0
15	1	0	0
16	1	0	0
17	1	1	0
18	1	0	0
19	1	16	0
21	1	16	11
8	12	0	0
20	6	13	13
7	7	0	0
23	2	5	0
26	1	15	15
22	2	0	0
24	1	0	0
25	1	12	0
3	8	0	0
27	2	12	12
28	2	0	0
29	4	0	0
30	1	0	0

Row	Group Name	
1	CARP	4 3 3 4 4 4 4 4 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 2 2 4 4 2 4 4 4 4 4 4 4 4 2 2 4 4 4 4
2	IRON	2 4 4 4 4 4
3	LAB	4 2 2 1 2 2 2 2 2 4

MAR APR 2002

Classic Schedule Layout | All Activities

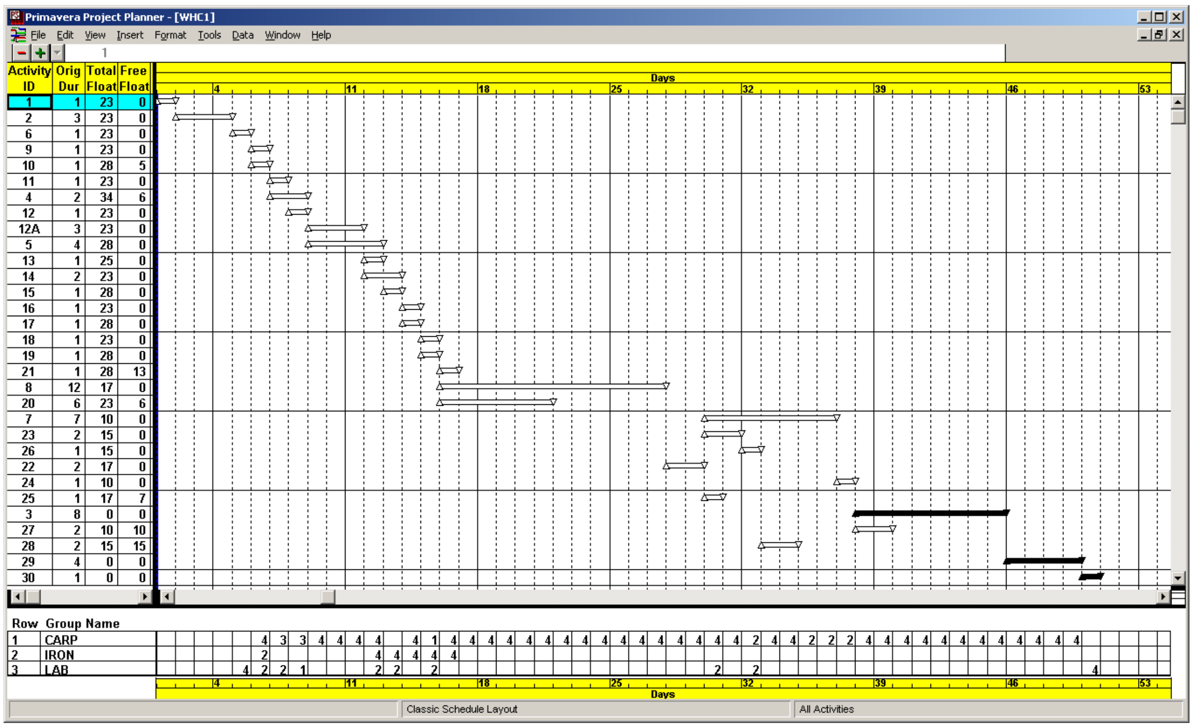
Figure 7.9 Rescheduled RCPM Output in Case-I Schedule

7.1.3 Rescheduling Methods in P3

Two methods are available to reschedule a resource-constrained schedule in P3. The method applied in this study is CPM first and then RCS next. The other method is simply rerunning RCS without CPM. There will be no dramatic work sequence change in the latter method for time extension cases. However, the major pitfall of this method is no activity can start earlier than the earliest start time of the base schedule, whether it is a CPM or a RCS schedule. If there were only time extensions in activities, the RCS only method would be acceptable in terms of work sequence stability (no earlier start than the schedule before change orders), although this schedule still contains phantom floats for most activities. However, if the duration of a critical activity, which can be easily identified through RCPM, is reduced, there will be no project completion time reduction, since no activity in P3 can start earlier than the earliest start of the base schedule.

For example, Figure 7.10 shows the output of the RCS only method after the change order. This schedule is surely more stable than rescheduled RCS after CPM (Figure 7.8). It is assumed at this point that, after reviewing of the schedule, the duration of *activity 8* can be reduced to 6 days with the same daily resource requirement. Figure 7.11 shows the output of the RCS only method after the duration of *activity 8* is reduced from Figure 7.10. There is no work scheduled from day 22 to 27 due to aforementioned pitfalls in the RCS only method. In order to eliminate the period, the CPM should be preceded prior to RCS. On the other hand, the RCPM schedule shown in Figure 7.12 is still stable and continuous without phantom floats.

In addition, a minor pitfall of the RCS only method is that it is painful to return to the output of the RCS only method, once CPM scheduling is applied. For example, when a user accidentally runs the CPM after completion of all progress updates, he/she may have to reenter all the progress data again into the base schedule to achieve the same RCS output. Because of these pitfalls, the CPM precedes RCS for rescheduling in this study.

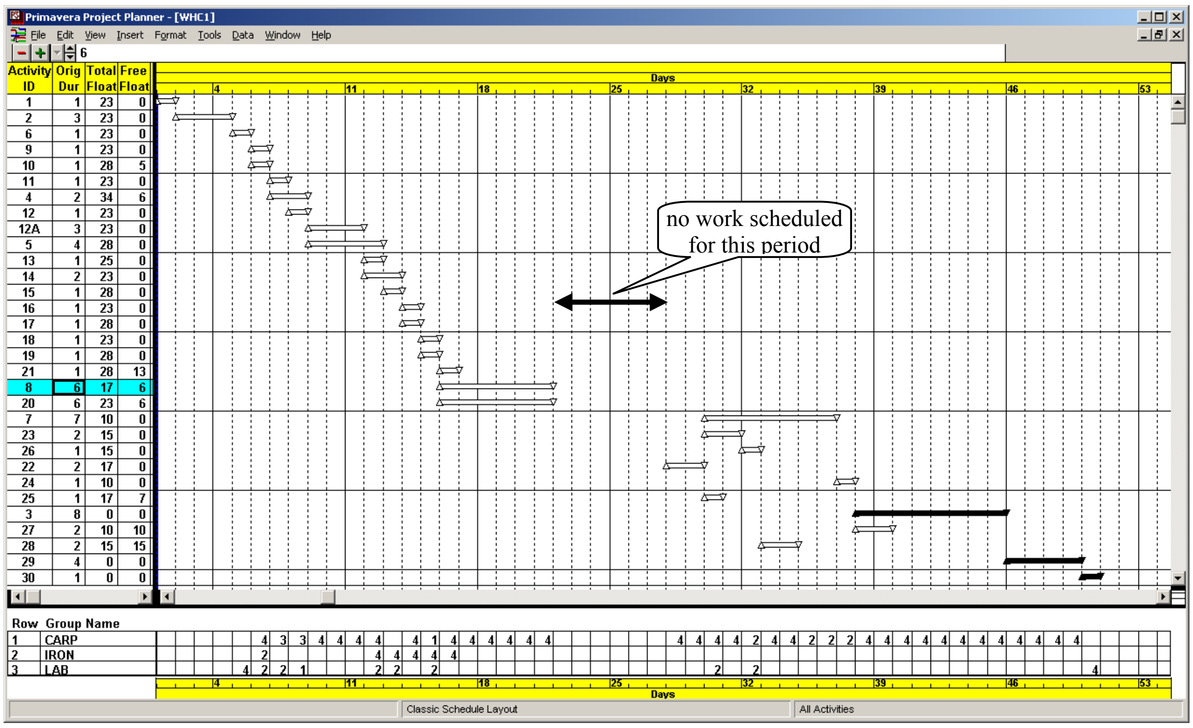

Primavera Project Planner - [WHC1]
File Edit View Insert Format Tools Data Window Help
1
Days
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float	Free Float
1	1	23	0
2	3	23	0
6	1	23	0
9	1	23	0
10	1	28	5
11	1	23	0
4	2	34	6
12	1	23	0
12A	3	23	0
5	4	28	0
13	1	25	0
14	2	23	0
15	1	28	0
16	1	23	0
17	1	28	0
18	1	23	0
19	1	28	0
21	1	28	13
8	12	17	0
20	6	23	6
7	7	10	0
23	2	15	0
26	1	15	0
22	2	17	0
24	1	10	0
25	1	17	7
3	8	0	0
27	2	10	10
28	2	15	15
29	4	0	0
30	1	0	0

Row	Group Name				4							11							18							25		
1	CARP						4	3	3	4	4	4	4		4	1	4	4	4	4	4	4	4	4	4	4	4	4
2	IRON						2						4	4	4	4	4											
3	LAB					4	2	2	1				2	2		2												

Row	Group Name					32							39							46							53	
1	CARP	4	4	4	4	2	4	4	2	2	2	4	4	4	4	4	4	4	4	4	4	4	4					
2	IRON																											
3	LAB				2	2																		4				

Figure 7.10 Rescheduled Output of RCS only Method (1)

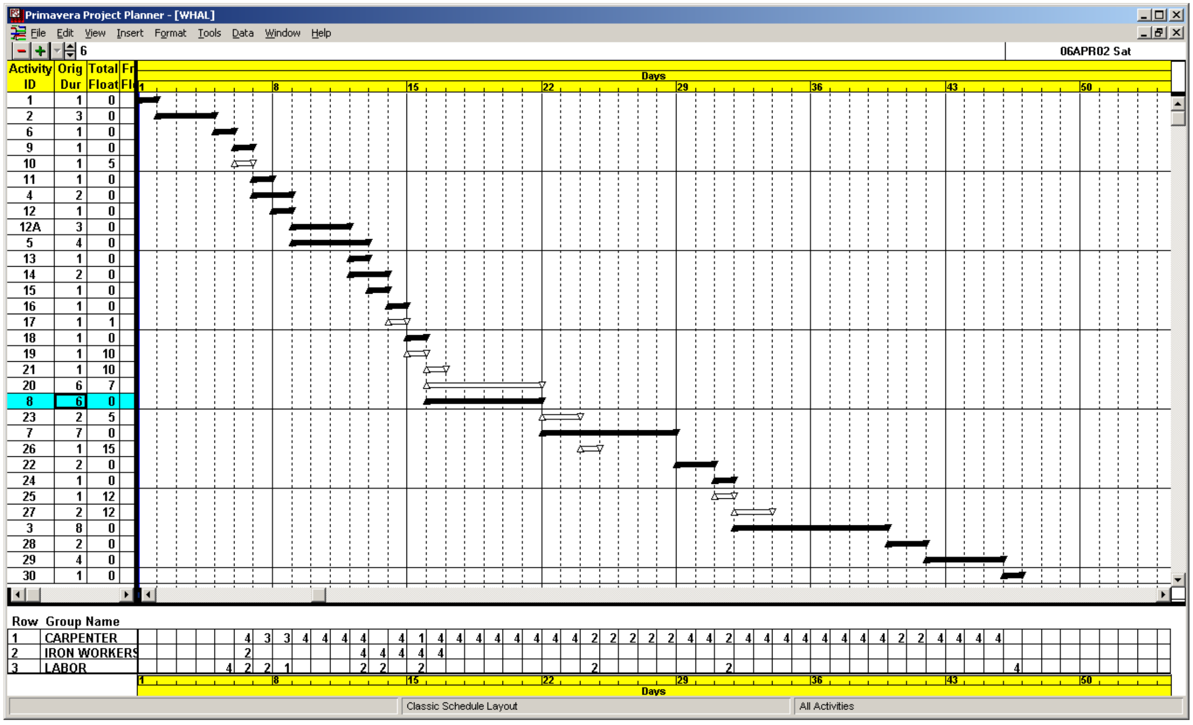

Primavera Project Planner - [WHC1]
File Edit View Insert Format Tools Data Window Help
6
Days
no work scheduled for this period
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float	Free Float
1	1	23	0
2	3	23	0
6	1	23	0
9	1	23	0
10	1	28	5
11	1	23	0
4	2	34	6
12	1	23	0
12A	3	23	0
5	4	28	0
13	1	25	0
14	2	23	0
15	1	28	0
16	1	23	0
17	1	28	0
18	1	23	0
19	1	28	0
21	1	28	13
8	6	17	6
20	6	23	6
7	7	10	0
23	2	15	0
26	1	15	0
22	2	17	0
24	1	10	0
25	1	17	7
3	8	0	0
27	2	10	10
28	2	15	15
29	4	0	0
30	1	0	0

Row	Group Name				4							11							18							25		
1	CARP						4	3	3	4	4	4	4		4	1	4	4	4	4	4	4						
2	IRON						2						4	4	4	4	4											
3	LAB					4	2	2	1				2	2		2												

Row	Group Name					32							39							46							53	
1	CARP	4	4	4	4	2	4	4	2	2	2	4	4	4	4	4	4	4	4	4	4	4	4					
2	IRON																											
3	LAB				2	2																		4				

Figure 7.11 Rescheduled Output of RCS only Method (2)


Primavera Project Planner - [WHAL]
File Edit View Insert Format Tools Data Window Help
6
06APR02 Sat
Days
1
8
15
22
29
36
43
50
Classic Schedule Layout
All Activities

Activity ID	Orig Dur	Total Float
1	1	0
2	3	0
6	1	0
9	1	0
10	1	5
11	1	0
4	2	0
12	1	0
12A	3	0
5	4	0
13	1	0
14	2	0
15	1	0
16	1	0
17	1	1
18	1	0
19	1	10
21	1	10
20	6	7
8	6	0
23	2	5
7	7	0
26	1	15
22	2	0
24	1	0
25	1	12
27	2	12
3	8	0
28	2	0
29	4	0
30	1	0

Row	Group Name	1							8							15							22							29							36							43							50			
1	CARPENTER						4	3	3	4	4	4	4		4	1	4	4	4	4	4	4	4	4	2	2	2	2	2	4	4	2	4	4	4	4	4	4	4	4	2	2	4	4	4	4								
2	IRON WORKERS						2						4	4	4	4	4																																					
3	LABOR					4	2	2	1				2	2		2									2							2															4							

Figure 7.12 Rescheduled Output of RCPM

7.1.4 RCPM after Rescheduling

RCPM should be run again after updating the schedule. Changes in the schedule could change resource usage condition so that additional resource links may be required to reflect the new condition. It is also possible that some resource links identified from the previous schedule are not necessary for the updated schedule. Currently the RCPM system does not provide functionality to remove resource links identified from the last schedule, but not required for the updated schedule. For this Case-1 schedule, the RCPM system has found no additional resource link required.

7.2 Multiple Calendar Schedule Example

7.2.1 Original Schedule

This example schedule includes all four PDM relationships and two calendars. Table 7.4 shows schedule data including calendar ID and relationship type and lag, and Figure 7.13 shows AON network of the schedule. Due to the discrepancy between P3 and RCPM as discussed in Chapter 4, negative lags of all four relationships and the start-to-finish relationship with zero lag are not included in this schedule.

Table 7.4 Case-II Schedule Activity Data

ID	Cal.	Dur.	Resources		Successors
			A	B	
1	1	1	0	0	2: FS(2) 3: FS(0)
2	1	3	2	2	4: FS(1) 5: SF(1)
3	2	3	3	2	6: FF(1)
4	1	1	0	3	12: FS(0)
5	1	2	1	2	7: FS(0) 8: SS(4)
6	2	4	0	3	8: SS(0)
7	2	3	1	0	9: FS(0)
8	2	1	2	0	11: FS(0)
9	1	3	2	0	10: SS(1) 11: FF(0)
10	1	3	0	0	12: FS(0) 13: FS(0)
11	2	3	1	2	16: FS(0)
12	1	3	0	1	14: FS(2)
13	2	3	0	0	15: SS(1)
14	1	2	1	3	17: SF(1)
15	1	1	0	3	16: FF(0)
16	2	2	2	4	17: SS(0)
17	2	3	0	0	
Max. Resource			3	6	

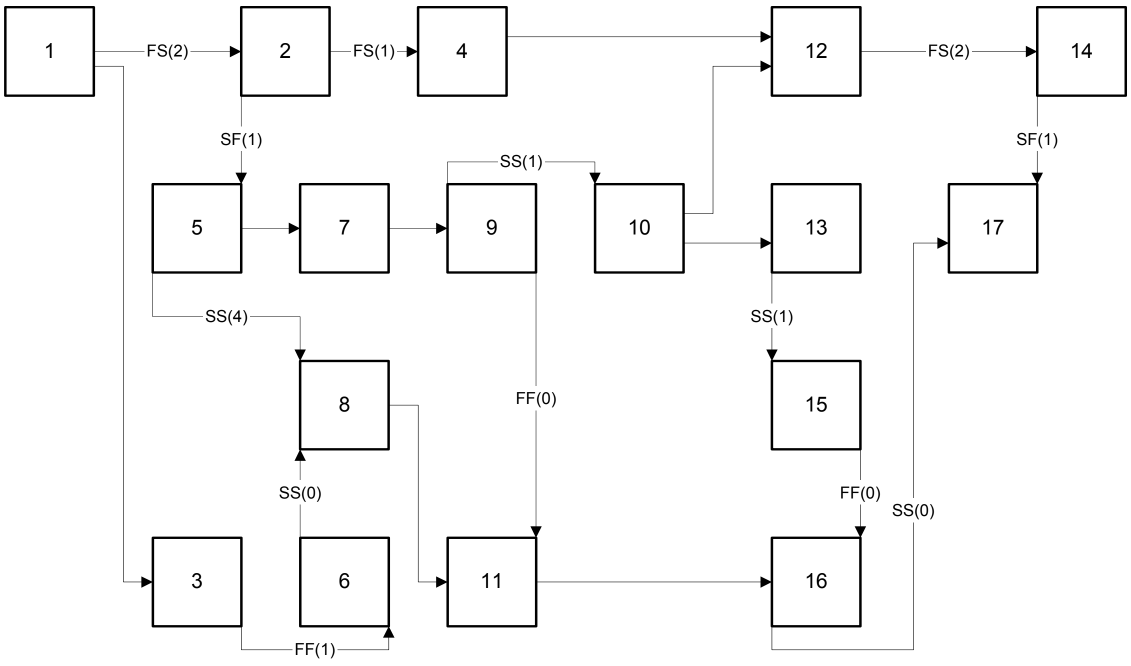


Figure 7.13 PDM Network of Case-II Schedule

July 2002						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4 ▨	5	6 ▨
7	8	9	10	11	12	13 ▨
14 ▨	15	16	17	18	19	20 ▨
21 ▨	22	23	24 ▨	25	26	27
28	29	30	31			

August 2002						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3 ▨
4 ▨	5	6	7	8	9	10 ▨
11 ▨	12	13	14	15	16	17 ▨
18 ▨	19	20	21	22	23	24 ▨
25 ▨	26	27	28	29	30	31 ▨

(a) Calendar 1: 5 days/week

July 2002						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
	1	2	3	4 ▨	5 ▨	6 ▨
7	8	9	10	11 ▨	12 ▨	13 ▨
14 ▨	15	16	17	18 ▨	19 ▨	20 ▨
21 ▨	22	23	24	25 ▨	26 ▨	27 ▨
28 ▨	29	30	31			

August 2002						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1 ▨	2 ▨	3 ▨
4	5	6 ▨	7	8 ▨	9 ▨	10 ▨
11 ▨	12	13	14	15 ▨	16 ▨	17 ▨
18 ▨	19	20	21	22 ▨	23 ▨	24 ▨
25 ▨	26	27	28	29 ▨	30 ▨	31 ▨

(b) Calendar 2: 3 days/week

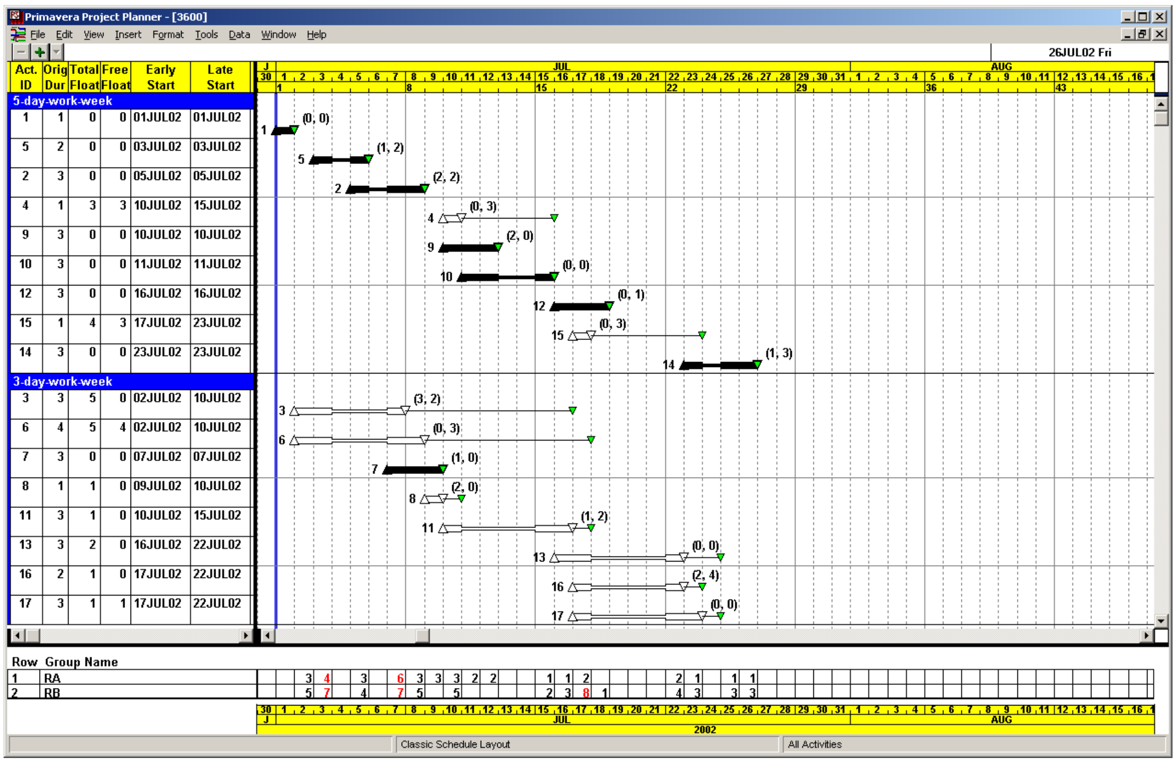
Figure 7.14 Calendars in Case-II Schedule

Two calendars used are shown in Figure 7.14. A P3 project can contain up to 31 different calendars, and each calendar can have a unique pattern of standard workdays per week, holidays, and exceptions. P3 also provides a *Global Calendar* of which holidays and exceptions are applied to all other calendars. In this schedule, *Calendar 1* has 5 standard workdays per week, and *Calendar 2* has 3 workdays per week. Holidays and exceptions shown in Table 7.5 are also included as shown in Figure 7.14. An exception date is a working day that is initially a holiday or a standard non-workday of the week.

Table 7.5 Holidays and Exceptions

	Global Calendar	Calendar 1	Calendar 2
Holidays	July 4	July 24	August 6
Exceptions	July 7	July 27 ~ July 28	August 4

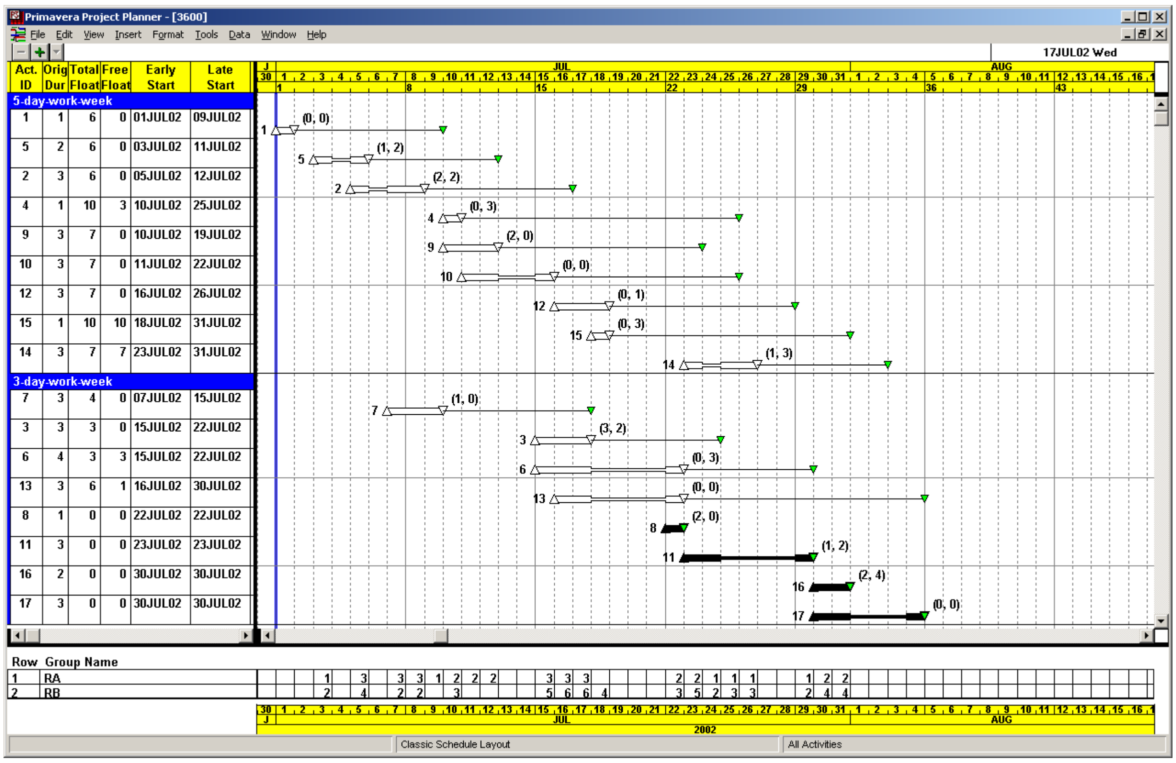
The P3 CPM schedule is shown in Figure 7.15, and the RCS schedule is in Figure 7.16 in which most activities contain phantom floats as expected. The resource links identified by the RCPM system is illustrated by curved line in Figure 7.17. They are added into the original P3 schedule, and then the P3 schedule becomes an RCPM schedule as shown in Figure 7.18. In addition, *activity 15*, as shown in Figure 7.19, can be scheduled any time between 27 and 29 July, which are beyond its LFT as marked in Figure 7.18.



Act. ID	Orig Dur	Total Float	Free Float	Early Start	Late Start
5-day-work-week					
1	1	0	0	01JUL02	01JUL02
5	2	0	0	03JUL02	03JUL02
2	3	0	0	05JUL02	05JUL02
4	1	3	3	10JUL02	15JUL02
9	3	0	0	10JUL02	10JUL02
10	3	0	0	11JUL02	11JUL02
12	3	0	0	16JUL02	16JUL02
15	1	4	3	17JUL02	23JUL02
14	3	0	0	23JUL02	23JUL02
3-day-work-week					
3	3	5	0	02JUL02	10JUL02
6	4	5	4	02JUL02	10JUL02
7	3	0	0	07JUL02	07JUL02
8	1	1	0	09JUL02	10JUL02
11	3	1	0	10JUL02	15JUL02
13	3	2	0	16JUL02	22JUL02
16	2	1	0	17JUL02	22JUL02
17	3	1	1	17JUL02	22JUL02

Row	Group Name	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	RA			3	4		3		6	3	3	3	2	2			1	1	2					2	1		1	1					
2	RB			5	7		4		7	5		5					2	3	8	1				4	3		3	3					

Figure 7.15 CPM Output of Case-II Schedule in P3



Act. ID	Orig Dur	Total Float	Free Float	Early Start	Late Start
5-day-work-week					
1	1	6	0	01JUL02	09JUL02
5	2	6	0	03JUL02	11JUL02
2	3	6	0	05JUL02	12JUL02
4	1	10	3	10JUL02	25JUL02
9	3	7	0	10JUL02	19JUL02
10	3	7	0	11JUL02	22JUL02
12	3	7	0	16JUL02	26JUL02
15	1	10	10	18JUL02	31JUL02
14	3	7	7	23JUL02	31JUL02
3-day-work-week					
7	3	4	0	07JUL02	15JUL02
3	3	3	0	15JUL02	22JUL02
6	4	3	3	15JUL02	22JUL02
13	3	6	1	16JUL02	30JUL02
8	1	0	0	22JUL02	22JUL02
11	3	0	0	23JUL02	23JUL02
16	2	0	0	30JUL02	30JUL02
17	3	0	0	30JUL02	30JUL02

Row	Group Name	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	RA				1		3		3	3	1	2	2	2			3	3	3					2	2	1	1	1			1	2	2
2	RB				2		4		2	2		3					5	6	6	4				3	5	2	3	3			2	4	4

Figure 7.16 RCS Output of Case-II Schedule in P3

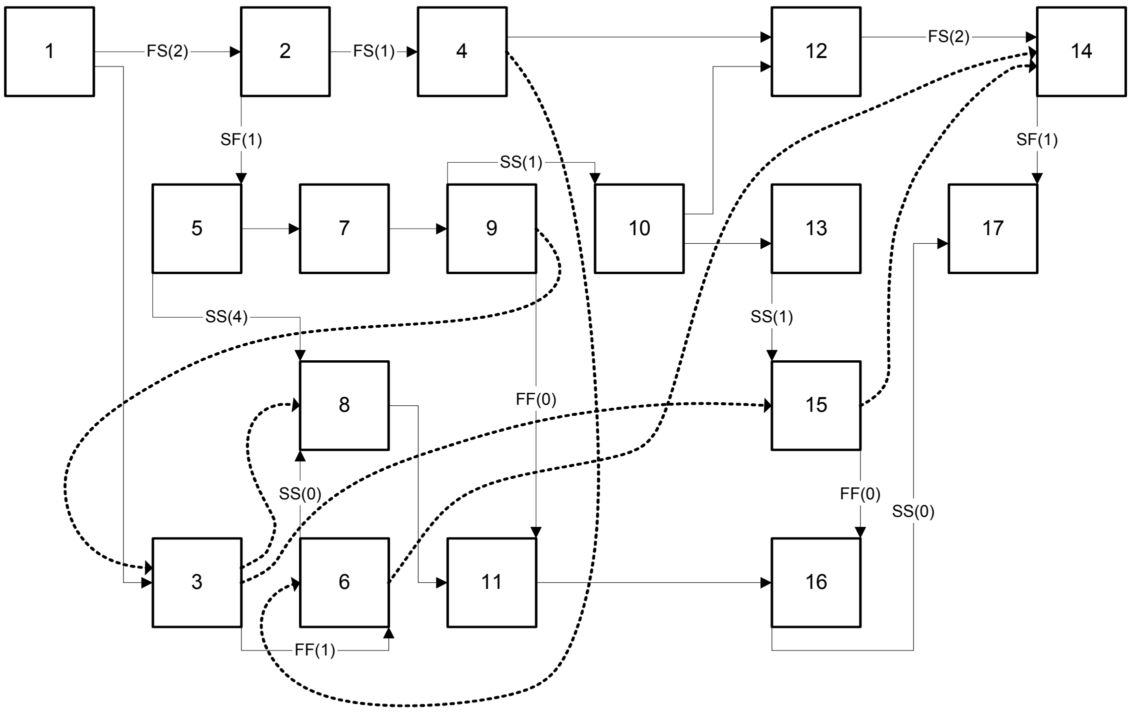
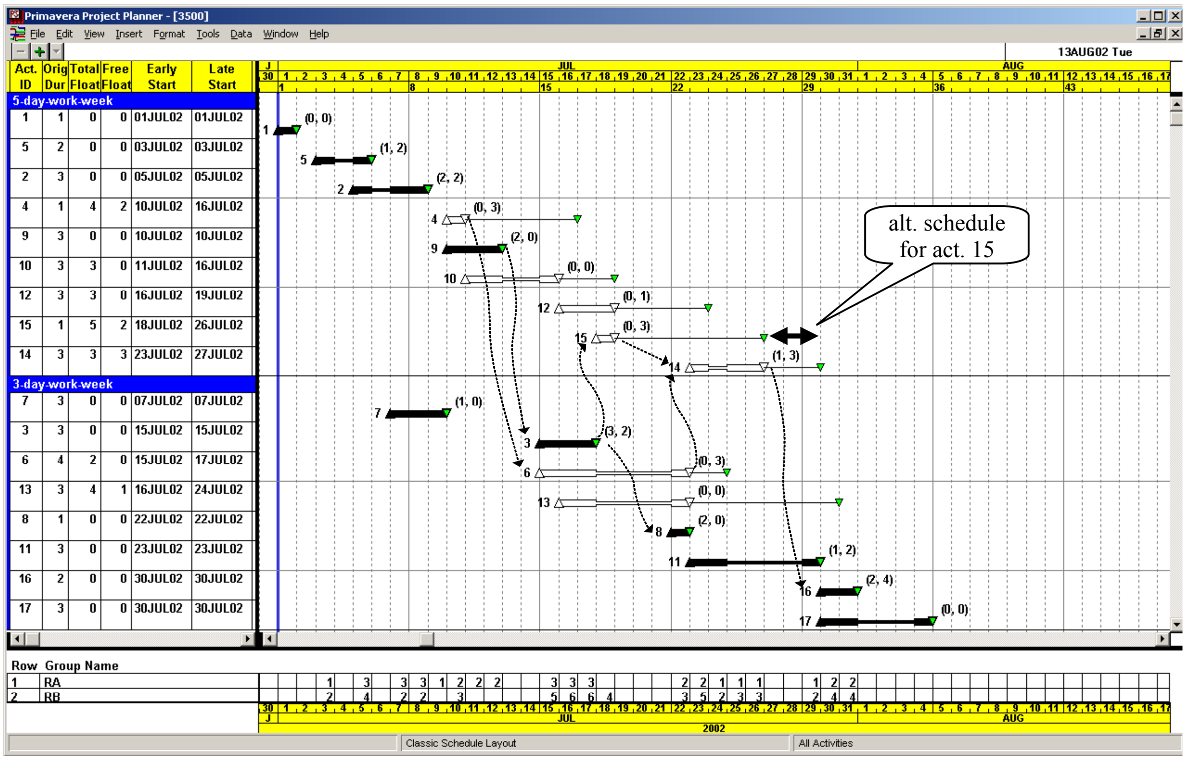


Figure 7.17 RCPM Network of Case-II Schedule

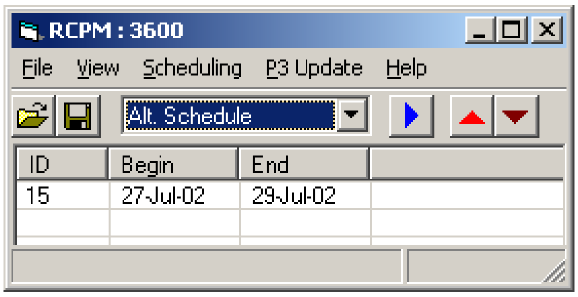


Act. ID	Orig Dur	Total Float	Free Float	Early Start	Late Start
5-day-work-week					
1	1	0	0	01JUL02	01JUL02
5	2	0	0	03JUL02	03JUL02
2	3	0	0	05JUL02	05JUL02
4	1	4	2	10JUL02	16JUL02
9	3	0	0	10JUL02	10JUL02
10	3	3	0	11JUL02	16JUL02
12	3	3	0	16JUL02	19JUL02
15	1	5	2	18JUL02	26JUL02
14	3	3	3	23JUL02	27JUL02
3-day-work-week					
7	3	0	0	07JUL02	07JUL02
3	3	0	0	15JUL02	15JUL02
6	4	2	0	15JUL02	17JUL02
13	3	4	1	16JUL02	24JUL02
8	1	0	0	22JUL02	22JUL02
11	3	0	0	23JUL02	23JUL02
16	2	0	0	30JUL02	30JUL02
17	3	0	0	30JUL02	30JUL02

Row	Group Name	3	5	7	8	9	10	11	12	15	16	17	18	22	23	24	25	26	29	30	31
1	RA	1	3	3	3	1	2	2	2	3	3	3		2	2	1	1	1	1	2	2
2	RB	2	4	2	2		3			5	6	6	4	3	5	2	3	3	2	4	4

2002

Figure 7.18 RCPM Output of Case-II Schedule in P3



ID	Begin	End
15	27-Jul-02	29-Jul-02

Figure 7.19 Alternative Schedules in Case-II Schedule from RCPM

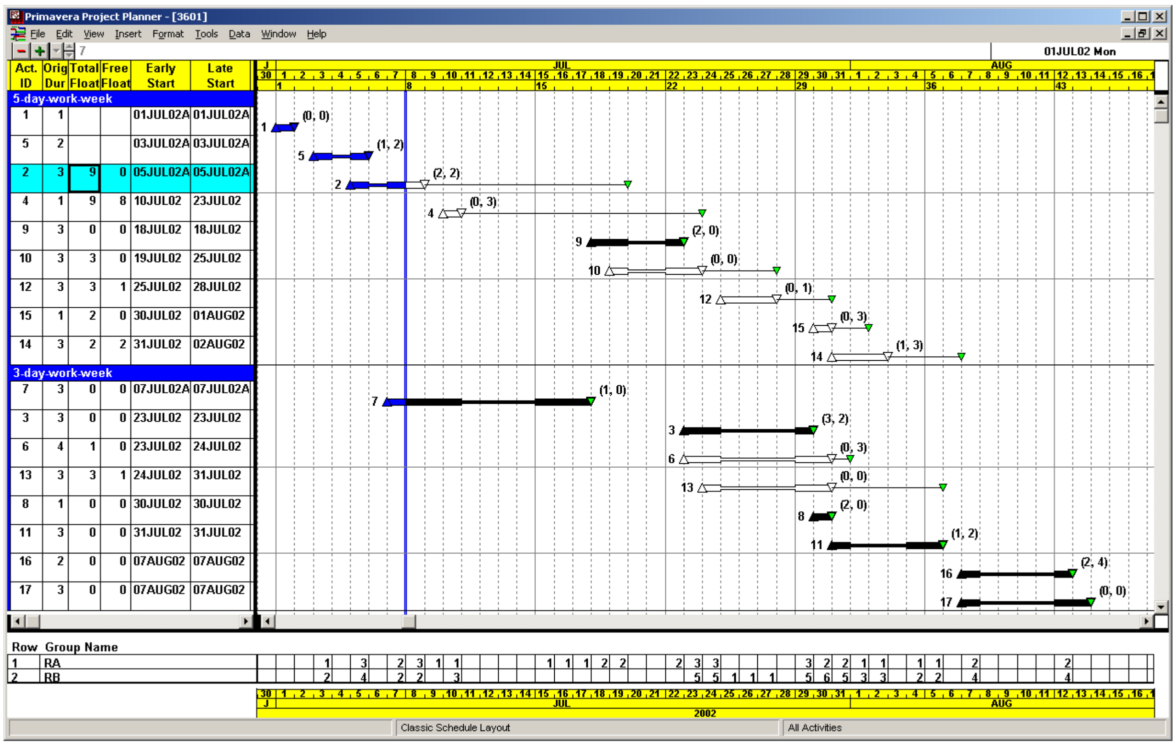
7.2.2 Progressed Schedule

At the end of the first week after the project started, the schedule has been updated based on the actual progress shown in Table 7.6. *Activity 1, 2,* and *5* are on schedule, but *Activity 7* requires 4 more days with the same daily resource requirement to finish. Figure 7.20 shows the updated RCPM schedule. The finish time delay of *Activity 7* has delayed *Activity 9* and its subsequent activities based on the technological and resource links.

RCPM is reapplied since the schedule is updated, and then a new resource link between *Activity 2* and *9* is identified. *Activity 2* had a zero total float due to the SF (1) relationship to *Activity 5* in the previous schedule (see Figure 7.18). Now in the progressed schedule, this relationship is no longer required since *Activity 5* was started and thus a 9-day total float is available for *Activity 2* (see Figure 7.20). This 9-day total float of *Activity 2* is not fully available because of *Resource A* and *Activity 9*s, since both activities require 2 of *Resource A*, but the limit of it is 3. Figure 7.21 shows the RCPM schedule with the additional resource link between *Activity 2* and *9*, which produces a 7-day total float for *Activity 2* showing that 2 days of 9 days total floats are phantom floats.

Table 7.6 Progresses as of the End of July 7

Act. ID	Actual Start Time	Actual Finish Time	Remaining Duration
1	July 1	July 1	0
2	July 5	-	1
5	July 3	July 5	0
7	July 7	-	6


Primavera Project Planner - [3601]

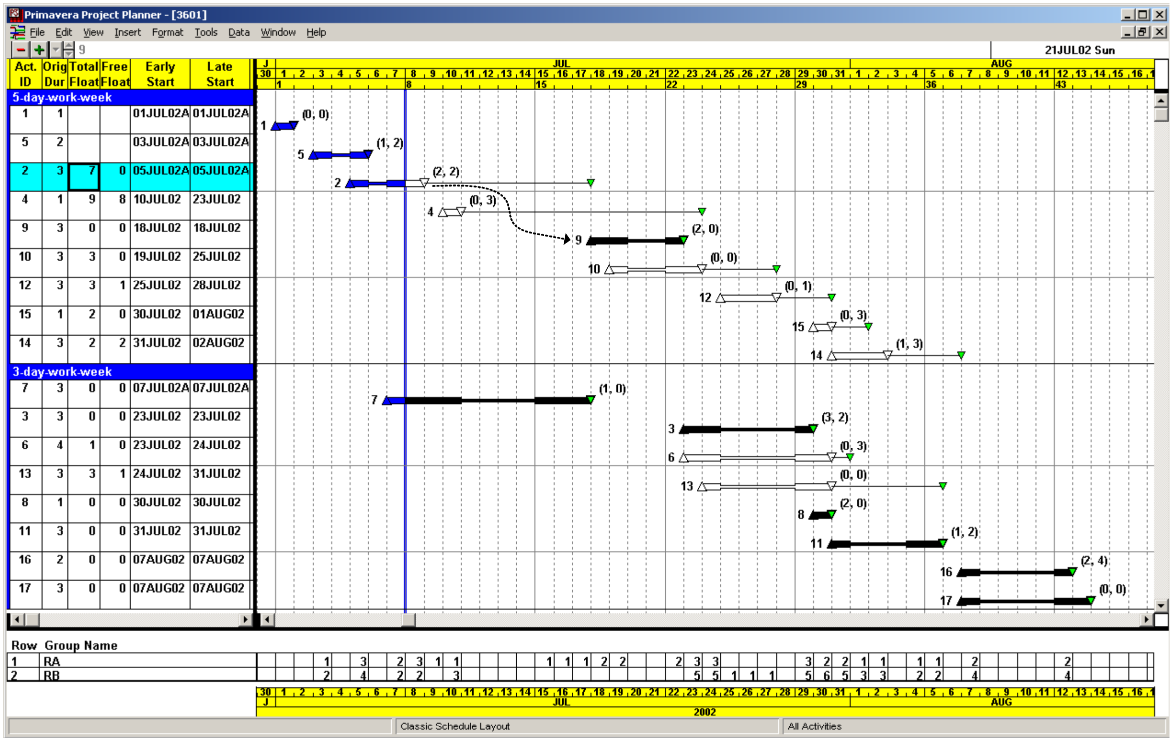
01JUL02 Mon

Act. ID	Orig Dur	Total Float	Free Float	Early Start	Late Start
5-day-work-week					
1	1			01JUL02A	01JUL02A
5	2			03JUL02A	03JUL02A
2	3	9	0	05JUL02A	05JUL02A
4	1	9	8	10JUL02	23JUL02
9	3	0	0	18JUL02	18JUL02
10	3	3	0	19JUL02	25JUL02
12	3	3	1	25JUL02	28JUL02
15	1	2	0	30JUL02	01AUG02
14	3	2	2	31JUL02	02AUG02
3-day-work-week					
7	3	0	0	07JUL02A	07JUL02A
3	3	0	0	23JUL02	23JUL02
6	4	1	0	23JUL02	24JUL02
13	3	3	1	24JUL02	31JUL02
8	1	0	0	30JUL02	30JUL02
11	3	0	0	31JUL02	31JUL02
16	2	0	0	07AUG02	07AUG02
17	3	0	0	07AUG02	07AUG02

Row	Group Name
1	RA
2	RB

Classic Schedule Layout
All Activities

Figure 7.20 Progressed RCPM Schedule


Primavera Project Planner - [3601]

21JUL02 Sun

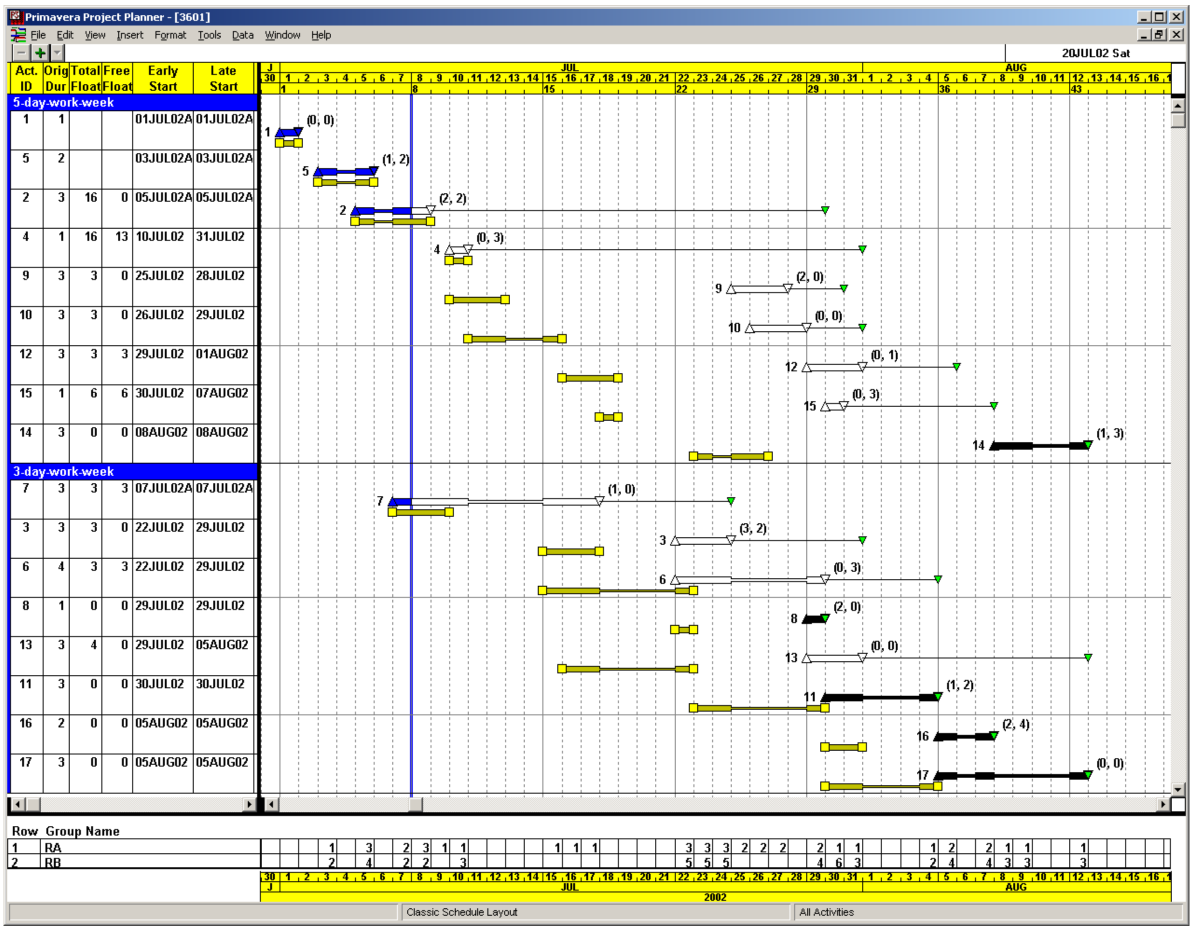
Act. ID	Orig Dur	Total Float	Free Float	Early Start	Late Start
5-day-work-week					
1	1			01JUL02A	01JUL02A
5	2			03JUL02A	03JUL02A
2	3	7	0	05JUL02A	05JUL02A
4	1	9	8	10JUL02	23JUL02
9	3	0	0	18JUL02	18JUL02
10	3	3	0	19JUL02	25JUL02
12	3	3	1	25JUL02	28JUL02
15	1	2	0	30JUL02	01AUG02
14	3	2	2	31JUL02	02AUG02
3-day-work-week					
7	3	0	0	07JUL02A	07JUL02A
3	3	0	0	23JUL02	23JUL02
6	4	1	0	23JUL02	24JUL02
13	3	3	1	24JUL02	31JUL02
8	1	0	0	30JUL02	30JUL02
11	3	0	0	31JUL02	31JUL02
16	2	0	0	07AUG02	07AUG02
17	3	0	0	07AUG02	07AUG02

Row	Group Name
1	RA
2	RB

Classic Schedule Layout
All Activities

Figure 7.21 Progressed RCPM Schedule with Additional Resource Link

On the other hand, Figure 7.22 shows the rescheduled RCS output after the update from the RCS schedule (Figure 7.16). In this chart, each activity has an early bar on top and a target bar (the initial RCS output in Figure 7.16) below. As expected, Phantom Floats are present since no resource links are incorporated, and the work sequence of several activities is changed compared to the target schedule.

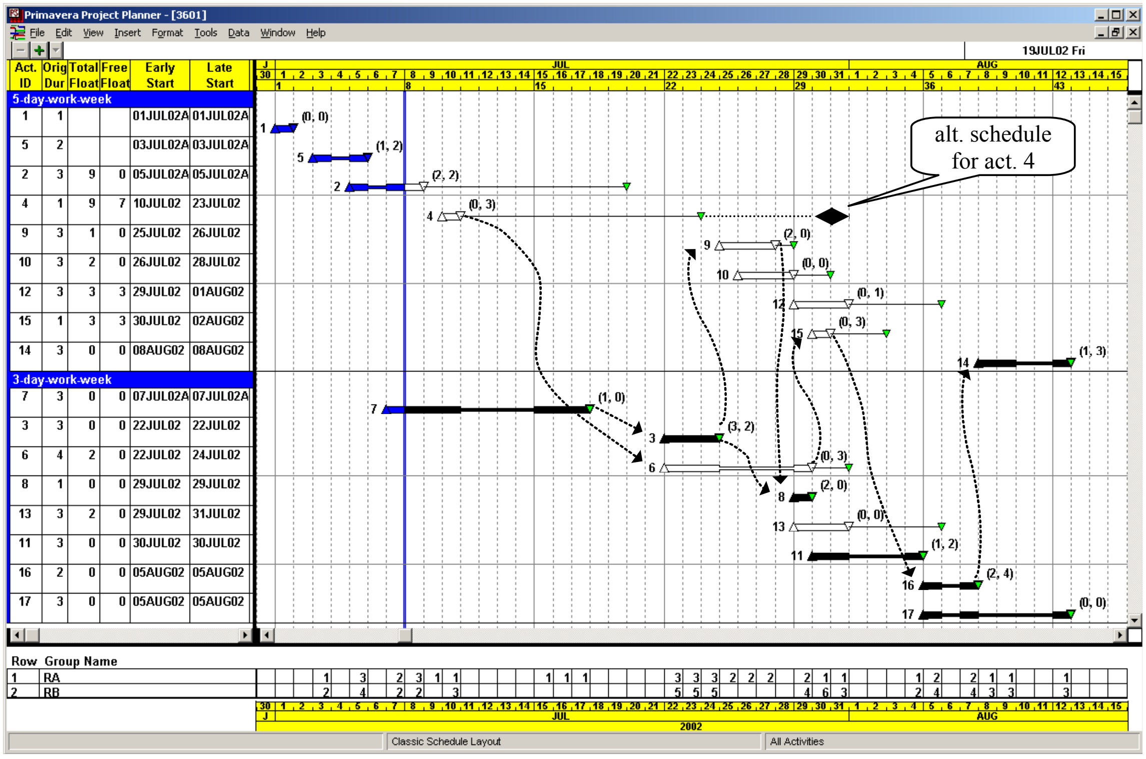


Primavera Project Planner - [3601]
File Edit View Insert Format Tools Data Window Help
20JUL02 Sat
JUL
AUG
2002
Row Group Name
1 RA
2 RB
Classic Schedule Layout
All Activities

Act. ID	Orig Dur	Total Float	Free Float	Early Start	Late Start
5-day-work-week					
1	1			01JUL02A	01JUL02A
5	2			03JUL02A	03JUL02A
2	3	16	0	05JUL02A	05JUL02A
4	1	16	13	10JUL02	31JUL02
9	3	3	0	25JUL02	28JUL02
10	3	3	0	26JUL02	29JUL02
12	3	3	3	29JUL02	01AUG02
15	1	6	6	30JUL02	07AUG02
14	3	0	0	08AUG02	08AUG02
3-day-work-week					
7	3	3	3	07JUL02A	07JUL02A
3	3	3	0	22JUL02	29JUL02
6	4	3	3	22JUL02	29JUL02
8	1	0	0	29JUL02	29JUL02
13	3	4	0	29JUL02	05AUG02
11	3	0	0	30JUL02	30JUL02
16	2	0	0	05AUG02	05AUG02
17	3	0	0	05AUG02	05AUG02

Figure 7.22 Progressed P3 RCS Schedule

RCPM is applied to the Figure 7.22 schedule in order to test how the RCPM system works with this progressed schedule (no resource link identified or included in this schedule) and to compare the result with the progressed RCPM schedule in Figure 7.21. RCPM performs well with this schedule, creating a different set of resource links compared to the original and the progressed RCPM schedules (Figure 7.18 and 7.21). Figure 7.23 shows the result after all resource links are incorporated; here, the ESTs of all activities are identical to the RCS schedule (Figure 7.22). In this schedule, *Activity 4* can be scheduled for 30 ~31 July, which is beyond its LFT, as marked in Figure 7.23. If *Activity 4* is scheduled on 30 July, *Activity 15* should be one day delayed to avoid resource overuse on that date.


Primavera Project Planner - [3601]
File Edit View Insert Format Tools Data Window Help
19JUL02 Fri
alt. schedule for act. 4
Classic Schedule Layout
All Activities

Act. ID	Orig Dur	Total Float	Free Float	Early Start	Late Start
5-day-work-week					
1	1			01JUL02A	01JUL02A
5	2			03JUL02A	03JUL02A
2	3	9	0	05JUL02A	05JUL02A
4	1	9	7	10JUL02	23JUL02
9	3	1	0	25JUL02	26JUL02
10	3	2	0	26JUL02	28JUL02
12	3	3	3	29JUL02	01AUG02
15	1	3	3	30JUL02	02AUG02
14	3	0	0	08AUG02	08AUG02
3-day-work-week					
7	3	0	0	07JUL02A	07JUL02A
3	3	0	0	22JUL02	22JUL02
6	4	2	0	22JUL02	24JUL02
8	1	0	0	29JUL02	29JUL02
13	3	2	0	29JUL02	31JUL02
11	3	0	0	30JUL02	30JUL02
16	2	0	0	05AUG02	05AUG02
17	3	0	0	05AUG02	05AUG02

Row	Group Name
1	RA
2	RB

Figure 7.23 RCPM Schedule with Progressed Schedule

Chapter 8

Summary, Future Studies, and Conclusion

8.1 Summary

The proposed Resource-constrained CPM (RCPM) is a heuristic algorithm, which establishes a CPM-like resource-constrained schedule by identified resource links. Alternative activity schedules are also available for the period beyond the resource-linked successors. RCPM takes advantages of both the CPM and the RCS techniques so it provides real total and free floats and the critical path, all of which are prerequisite for successful project planning and control. In addition, the RCPM technique provides a certain level of stability that a traditional resource-constrained schedule often does not.

RCPM performs based on multiple calendars. Scheduling with multiple calendars considering weekend, holidays and exceptions is more realistic than scheduling with continuous working days. Although P3 and MS-Project employs multiple calendars, the basic theories have not been disclosed to public. This study provides how multiple calendars are handled in scheduling through the examination on the calendar operations of P3. From this reverse engineering, it has been also noticed that the P3 operations with two calendars may generate a wrong answer for start-to-finish with zero lag and contradictory results in all negative lags when nonworking days are involved.

Based on the RCPM algorithm and the calendar handling operation, a prototype RCPM system is developed. This system is integrated with P3 so that a user can directly read a P3 project from the RCPM system and execute the RCPM procedures. Identified resource links can also be automatically incorporated into the P3 project from the RCPM system. Once resource links are inserted, the P3 schedule becomes an RCPM schedule. This RCPM system runs not only initial schedules but also progressed schedules.

The RCPM algorithm outperforms previous algorithms in handling multiple resources and parallel activities. Two case studies also show the benefit of RCPM over the traditional RCS. Based on these benefits, project mangers will be encouraged to use a resource-constrained schedule more actively.

8.2 Contributions to the Body of Knowledge

This study has contributed to the body of knowledge by improving traditional scheduling techniques and other related previous works in finding resource dependencies between activities in a resource-constrained schedule. Major contributions are as follows.

1) The proposed RCPM technique provides more reliable scheduling data than other technologies. An RCPM schedule can be more realistic than a CPM schedule because RCPM considers resource availability. An RCPM schedule can also be more practical than a schedule generated by traditional RCS technologies, because RCPM removes Phantom Floats and provides a continuous critical path. Hence, the quality of the schedule has been improved.
2) Stable work sequence with schedule updates is available because an RCPM schedule is a CPM-like schedule, in which there is no priority rule to sort and to schedule activities reflecting resource availability.
3) An RCPM schedule could be better in analyzing delay impacts than other schedules. A CPM schedule is not appropriate unless all resource constraints are reflected as technological relationships. A schedule generated by RCS is also not appropriate for a delay impact analysis due to phantom floats and unstable work sequence after schedule changes.
4) Applying RCPM could reduce the number of time-related claims. RCPM provides a more accurate schedule because of loaded resource constraints, correct float times, and stable work sequence, probably resulting in less disputes than with CPM or RCS.

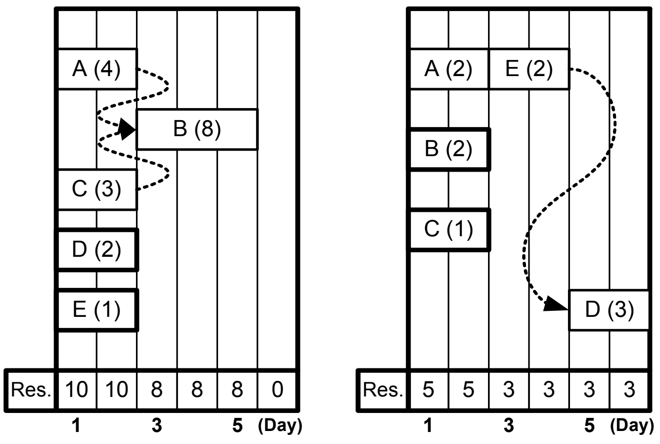
5) Alternative schedules of certain activities are additional benefits of RCPM to make the schedule more flexible against unanticipated circumstances. Most alternative schedules are available in the RCPM schedule unless they are overlapped with each other competing the same resources beyond their availability.
6) RCPM performs well in identifying resource dependencies between activities compared to other previous algorithms. First, identified resource links show how resource dependencies are present in the scheduling network and how a change in an activity affects other subsequent activities. Second, there is no resource limit violation for the total float period in a RCPM schedule. Third, RCPM does not create unnecessary resource links, which can cause incorrect float data, because RCPM considers not only resource dependencies but also technological relationships. Finally, RCPM provides a systematic procedure to handle multiple resources in multiple parallel activities.
7) RCPM can control not only an initial schedule but also a progressed schedule thanks to its functionality to handle calendars. Hence, RCPM can be adopted at any phase of a project. In addition, RCPM is ready for application to a real project that contains multiple calendars.
8) The scheduling process with multiple calendars can help project participants understand their schedule (with multiple calendars) generated by other project management software.
9) The prototype RCPM system enhances a practical use of the RCPM algorithm and gives more opportunities to encounter further requirements of the algorithm in a future RCPM system.

8.3 Future Studies

Followings are recommended future studies to enhance the capability of the current RCPM technique and system and to make them more practical in real construction projects.

8.3.1 Float Sharing

In a CPM schedule, activities are sharing TF only if they are on the same chain, but in the RCPM schedule, this basic concept may not hold true. As shown in Figure 8.1, due to sharing of resources, some parallel activities are sharing floats while not in the same chain. For example, in Figure 8.1(a), either *activity D* or *E* can be delayed up to 4 days, but both cannot be delayed without delaying *activity B*. A similar case happens for *activity B* and *C* in Figure 8.1(b). Shared floats can be consumed by any activity that needs it first, and then an additional resource link is required to reflect the condition. For instance, if the finish time of *activity D* in Figure 8.1(a) needs to be delayed, then *activity E* will have a resource link with *activity B*.



(a) resource limit: 10 **(b) resource limit: 5**

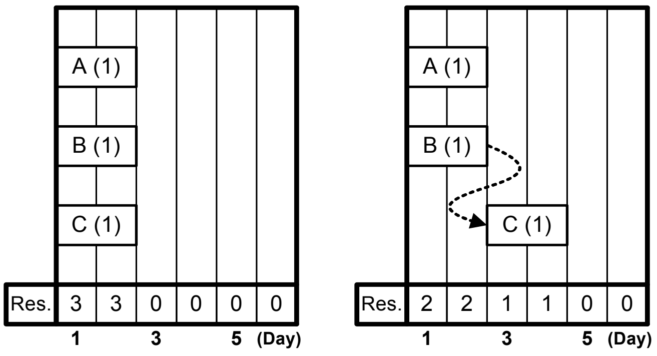
Figure 8.1 Float Sharing among Parallel Activities

The RCPM algorithm or system is currently not able to detect these shared floats without updating the schedule. If this sharing information is available at the beginning, the RCPM schedule can be more practical and realistic against unexpected events such as delays or change orders.

8.3.2 Alternative Resource Link

In step 2, RCPM makes at least one resource link when an activity is delayed due to resource availability. In this way, the EST of the delayed activity can be correctly described in the RCPM schedule with P3; otherwise, the same situation will be encountered as in the special case with multiple calendars (Figure 6.8). However, this “at least one resource link” may cause another problem.

Figure 8.2 illustrates an example. With a limit of two resources, *activity C* is delayed, then a search is performed, and *activity A* and *B* are found for possible predecessors by resource link. But, a time extension of either *activity A* and *B* does not directly affect the start time of *activity C*, and the resource requirement of both activities are the same. This tie can be resolved by the reverse order (*A, B, C*) of the sorted activity list for RCS, and thus a resource link is created between *activity B* and *C*.



(a) initial schedule **(b)RCPM schedule**

Figure 8.2 Dynamic Resource Links Example

This resource link, however, does not correctly represent current resource dependency. When *activity B* is delayed, no delay is required for *activity C*. In this case, a resource link between *activity A* and *C* can be used instead of *activity B* and *C*. Identifying these alternative links is available in the RCPM algorithm, but how to apply them dynamically with a schedule update has not been established yet. More research is required on this aspect.

8.3.3 Resource Links with Rescheduling

In the rescheduling process of the case studies in Chapter 7, all resource links identified at the initial schedule are fixed and treated as technological relationships. However, RCPM requires one more day than the RCS in P3 for both cases. This happens because the RCPM schedule has more activity relationships and so less flexibility to decrease project duration effectively. Assuming a shorter duration is desirable, a trade-off is required to handle the resource links prior to rescheduling with a schedule update. Three possible scenarios are available.

The first method, as in the case studies, is to fix all resource links like technological relationships, if there is no significant extension in project completion. Then, the schedule in this method becomes a CPM-like schedule that has stable work sequence, but also considering resource limits. The second method is to fix the resource links on certain period (one week, one month, one quarter, etc. based on project condition) or on certain criteria based on users demand, and neglect all other resource links. In this way, work sequence in the given period or in the given criteria will not be changed, but others will be changed by resource constrained scheduling, possibly giving a shorter duration. The last method is to neglect all resource links. In this case, a stable schedule is not available, but the project completion time may be the earliest possible time with the given RCS.

The selection of a particular method is fully dependent on the project conditions. The contract type, relationship between project participants, progress status, etc. could affect the decision. More research is required to investigate these scenarios. Especially, the second method will require more effort than the other methods to separate resource links from technological links and then to implement a reliable, systematic procedure. In addition, the resource usage condition can be changed in the updated schedule. Thus, it is also possible that some resource links identified from the previous schedule are not necessary for the updated schedule in the first and the second scenarios. Currently RCPM does not provide functionality to remove resource links, identified from the previous schedule, but not required for the updated schedule during the rescheduling process.

8.3.4 Alternative Schedules

Dependency

Each alternative schedule of an activity does not consider the other alternative schedules in current RCPM. Thus, if one alternative is selected, then some other alternative schedules may not be available. It will be more helpful if the RCPM system provides the dependency among alternative schedules.

Resource Links

Although a resource link is created in the alternative schedule of the SEMP in Chapter 3, the RCPM system does not provide this information. In addition, as discussed in the case study-I, some activity should be delayed consuming its TF period for another activity's alternative schedules. This information is not available from the RCPM system too. The only way to apply an alternative schedule currently is to manually exchange resource links based on the given RCPM output and the observation of the current schedule. It is desirable to automatically apply alternative schedules upon users' demand directly from the RCPM system.

Lifetime

Alternative schedules are available only in the RCPM system output after one cycle of RCPM is applied. If the same schedule is applied another time, the same alternative schedules will be identified. However, once resource links are added to the original P3 project, the same alternative schedule will be no longer available from the schedule. More work is required to store alternative schedule information and to handle them with updated progressed schedules.

8.3.5 Unidentified Resource Links in Multiple Calendars

When an activity is delayed and scheduled for an interrupted period (non-working days) of the delayed caused activity, RCPM does not create a resource link as discussed in the test schedule D of Chapter 6 (Figure 6.8). The RCPM system does provide correct early times of the delayed activity in its output. However, once all resource links are added into the P3 project, the P3 schedule does not represent the correct RCPM condition resulting in a resource limit violation, since this special case or an incoming resource link for the delayed activity is not present. Adding a dummy activity or applying a start-to-start relationship with lag could be a solution for this case. More research is required to evaluate these approaches for both initial and progressed schedules.

8.3.6 P3 Options or Constraints

It has been recognized that real construction projects tested with the RCPM system contains various P3 options or constraints. Followings are options that do not violate the Precedence Diagramming Method (PDM) network logic so that the RCPM system is encouraged to implement them.

- variable maximum resource amounts for several periods
- selected priorities rather than the fixed "LST, remaining duration, TF, and Activity ID" order in RCPM
- hammock activity
- repetitive holidays
- automatic holiday adjusting option when a holiday is on a weekend

In contrast, following options most likely violate the network logic so that it is recommended not to use them in scheduling, but the RCPM system is still required to accommodate them since many real projects do contain for a convenience in scheduling.

- activity time constrains such as early start/finish, late start/finish, start on, mandatory start/finish, expected finish, etc.
- immediate priority option in the leveling type for each activity

8.3.7 Parallel RCS Method

Although RCPM employs the serial RCS technique, which is the P3 algorithm, the parallel RCS technique is another well-known scheduling method to reflect resource constraints. If RCPM applies the parallel method, it may generate a different project completion time and different resource links in most projects compared to when the serial method is adopted. If the RCPM system employs both serial and parallel methods creating two different schedules, the better schedule can be selected depending on the project conditions. More research is required to implement the parallel method, to compare outputs between two methods, and to incorporate the selected schedule into P3.

8.3.8 Delay Impact Analysis with RCPM

More investigation is required to apply the RCPM concept or the system to delay impact analyses. Since RCPM is developed focusing on the scheduling purpose and the split option (common in a delayed activity) in an activity is not allowed, RCPM may require some additional features for delay impact analyses.

8.3.9 Technological vs. Resource-constrained Link

Many CPM construction schedules often include resource links in addition to genuine technological activity relationships. The resource links can be eliminated if more resources are provided, but the technological links cannot be removed, even with unlimited resource supplies. The RCPM technique assumes that the CPM schedule contains only the genuine technological links and creates resource links in a systematic way using the computer power. Under this circumstance, the following questions are asked.

What will happen if the CPM schedule already contains resource links? Should those initial resource links be removed for a better schedule? Can the RCPM-generated schedule without the initial resource links be more effective for the project than the RCPM-generated schedule with the initial resource links?

8.3.10 Calendar for Lag Time

From the basic rule in scheduling with multiple calendars, the lag time of an activity relationship belongs to the predecessor's calendar. It has been noticed that it will be more efficient if the lag time in some cases can have its own calendar that is independent of any activity. For example, the activity placing-concrete can be on a 5-day calendar, but curing needs to be on a 7-day calendar. Curing can be handled with a finish-to-start lag with its own 7-day calendar, rather than creating an activity called curing with its own 7-day calendar. More research is required to examine assigning an independent calendar to the lag time.

8.4 Conclusion

CPM has significantly contributed to construction scheduling processes, providing important time information such as total/free floats and the critical path. In addition, the CPM schedule is a vital source for a delay impact analysis in time related claims. However, the CPM does not consider constrained resources.

In most construction projects, certain resources are highly restricted, so a CPM schedule is unrealistic if appropriate resource allocation is not employed. Hence, a number of resource constrained scheduling (RCS) technologies have been developed to consider these limited resources in a scheduling process. However, these RCS technologies do not provide correct total/free floats and critical path, which the CPM schedule provides. The current RCS technologies are not equipped to calculate late start/finish times that arise not only by technological activity relationships but also by resource constrained relationships (resource links).

Due to the absence of resource links, the floats created in the current RCS technologies may have considerable non-existing floats. The non-existent floats in the current RCS technologies are referred to as Phantom Floats in this study. Leading project management software such as Primavera Project Planner (P3) and MS-Project still create phantom floats in a resource-constrained schedule because no practical way to identify resource links is currently available. Since real floats are unknown in a resource-constrained schedule, every activity has to be treated as a critical activity resulting in a serious burden for the project manger.

Another shortcoming of the current RCS technologies is unexpected work sequence changes after a schedule update. This problem results from the changed activity priority orders in the traditional RCS technologies. The activity order can be changed anytime with a progress update, resulting in a different work sequence. Additional time and cost will be necessary to reorganize the changed work sequence.

Fondahl (1991) pointed out that many disputes among project participants are triggered by incorrect scheduling information, and those disputes are often also resolved by incorrect information. John Fondahl was referring to the presence of Phantom Float being totally unnoticed. US courts, in recent times however, have realized that a CPM schedule without resource constraints is not realistic, expecting a resource-loaded schedule with the critical path for a time impact analysis [Wickwire 2001]. In the view of this fact, in addition to planning and controlling benefits, tools and technologies like RCPM will become more valuable and their usage will be increased as their utilization is required by courts of law. Continued research in this area will enhance the capability of the current RCPM technique, add new knowledge, and provide valuable assistance for construction project management.

REFERENCE

Ahuja, H. N. (1976). *Construction performance control by networks*, John Wiley & Sons, New York, NY.

Ahuja, H. N. and Nandakumar, V. (1985). "Calendar date scheduling on microcomputers" *Cost engineering*, AACE, 27(2), 21-27.

Badiru, A. B. (1993). "Activity-resource assignment using critical resource diagramming" *Project management journal*, PMI, 24(3), 15-21.

Bartholomew S. H. (1998). *Construction contracting: Business and legal principles*, Prentice Hall, Upper Saddle River, NJ.

Battersby, A. (1967). *Network analysis for planning and scheduling*, St Martin's Press, New York, NY.

Boctor, F. F. (1990). "Some efficient multi-heuristic procedures for resource-constrained project scheduling" *European journal of operational research*, 49, 3-13.

Bowers, J. A. (1995). "Criticality in resource-constrained networks" *Journal of the operational research society*, www.palgrave-journals.com/jors, 46(1), 80-91.

Bowers, J. A. (2000). "Multiple schedules and measures of resource-constrained float" *Journal of the operational research society*, www.palgrave-journals.com/jors, 51(7), 855-862.

Callahan, M. T., Quackenbush, D. G., and Rowings, J. E. (1992). *Construction project scheduling*, McGraw-Hill, New York, NY.

Clough, R. H., Sears, G. A. and Sears S. K., (2000). *Construction project management 4th Ed*, John Wiley & Sons, New York, NY.

David, E. W., and Patterson, J. H. (1975). “A comparison of heuristic and optimum solutions in resource-constrained project scheduling” *Management science*, 21(8), 944-955.

de la Garza, J. M. and Vorster, M. C. (1991). “Total float traded as commodity” *Journal of construction engineering and management*, ASCE, 117(4), 716-727.

Feibus, A (1998). “Project leaders” *Informationweek*, Manhasset, NY., August, 1A-10A.

Fondahl, J. W. (1991). “ The development of the construction engineer: Past progress and future problems.” *Journal of construction engineering and management*, ASCE, 117(3), 380-392.

Hegazy, T. (1999). “Optimization of resource allocation and leveling using genetic algorithms” *Journal of construction engineering and management*, ASCE, 125(3), 167-175.

Hernandez T. J. (1999). “Software solutions” *Building design & construction*, November, 38-40.

Just, M. R., and Murphy J. P. (1994). “The effect of resource constraints on project schedules*” AACE Transaction*s, DCL.2.

Kelley, J. E. Jr. (1963). “The critical-path method : resource planning and scheduling” *Industrial scheduling*, Prentice Hall, Englewood Cliffs, NJ, 347-365.

Korman, R., and Daniels, S. H. (2003). “Critics can’t find the logic in many of today’s CPM schedules” *Engineering News-Record*, 250(20), 30-33

Knutson, J. (2001). *Project management for business professionals: a comprehensive guide*, John Wiley & Sons, New York, NY.

Leu, S., Chen, A., and Yang, C. (1999). “Fuzzy optimal model for resource-constrained construction scheduling” *Journal of construction engineering and management*, ASCE, 125(3), 207-216.

Li, R. K, and Willis, R. J. (1993). “Resource constrained scheduling within fixed project durations” *Journal of the operational research society*, www.palgrave-journals.com/jors, 44(1), 71-80.

Liberatore, M. J., Pollack-Johnson, B., and Smith, C. A. (2001). “Project management in construction: software use and research directions” *Journal of construction engineering and management*, ASCE, 127(2), 101-107.

Lock, D. (2000). *Project management: seventh edition*, Gower, Burlington, VT.

Melin, J. W., and Whiteaker, B. (1981). “Fencing a bar chart” *Journal of construction division*, ASCE, 107(3), 497-507.

Melin, J. W. (1995). *CE 316 Class Notes*, Department of Civil Engineering, University of Illinois at Urbana-Champaign.

Moder, J. J. and Phillips, C. R (1964). *Project management with CPM, PERT*, Reinhold Publishing Corporation, New York, NY.

Moder, J. J., Phillips, C. R, and Davis, E. W (1983). *Project management with CPM, PERT, and precedence diagramming, 3rd Edition*, Van Nostrand Reinhold Company, New York, NY.

Primavera (1998). *RA help system*, Primavera Systems, Inc., Bala Cynwyd, PA.

Primavera (1999a). *Reference manual: Primavera Project Planner Ver. 3.0*, Primavera Systems, Inc., Bala Cynwyd, PA.

Primavera (1999b). *P3 help system Ver. 3.1*, Primavera Systems, Inc., Bala Cynwyd, PA.

Ritche, E. (1985). “Network based planning techniques: a critical review of published developments” *Further developments in operational research*, Pergamon Press Inc. New York, NY.

Wallwork, J. W. (2002). “Standards for a critical path method schedule*” Cost engineering*, AACE, 44(6), 13.

Wickwire, J. M., Driscoll, T. J., and Hurlbut, S. B (2001). *Construction scheduling: preparation, liability, and claims, 2001 cumulative supplement*, Aspen Law & Bussiness, New York, NY.

Wickwire, J. M., Driscoll, T. J., Hurlbut, S. B (2003), and Hillman, S. B. *Construction Scheduling: preparation, liability, and claims, 2nd Ed.*, Aspen Publishers, New York, NY.

Wiest, J. D. (1964). “Some properties of schedules for large projects with limited resources” *Operations research*, 12, 395-418.

Wiest, J. D. (1967). “A heuristic model for scheduling large projects with limited resources” *Management science*, 13(6), B359-B377.

Willis, R. J. (1985). “Critical path analysis and resource constrained project scheduling – Theory and practice” *European journal of operational research*, 21, 149-155.

Willis, E. M. (1986). *Scheduling construction projects*, John Wiley & Sons, Inc., New York, NY.

Woodworth, B. M., and Shanahan, S. (1988). “Identifying the critical sequence in a resource constrained project” *International journal of project management*, 6(2), 89-96.

Woodworth, B. M. (1989). “Is resource-constrained project management software reliable*?” Cost engineering*, 31(7), 7-11.

BIBLIOGRAPHY

Ahuja, H. N., and Arunachalam, V. (1984). "Risk evaluation in resource allocation" *Journal of construction engineering and management*, ASCE, 110(3), 324-336.

Appleman, D. (1999). *Visual Basic programmer's guide to the Win32 API, SAMS*, Indianapolis, IN.

Ballard, G., and Howell. G. (1998). "Shielding production: essential step in production control" *Journal of construction engineering and management*, ASCE, 124(1), 11-17.

Basu, A. (1990). "A mechanical model for CPM scheduling calculations" AACE Transactions, AACE, H.5.1-H.5.6.

Bordoli, D. W., and Baldwin, A. N. (1998). "A methodology for assessing construction project delays" *Construction management and economics*, E & FN Spon, 16, 327-337.

Bubshait, A. A., and Cunningham, M. J. (1998). "Comparison of delay analysis methodologies" *Journal of construction engineering and management*, ASCE, 124(4), 315-322.

Christofides, N., Alvarez-Valdes, R., and Tamarit, J. M. (1987). "Project scheduling with resource constraints: a branch and bound approach" *European journal of operational research*, 29, 262-273.

Davis, E. W. (1966). "Resource allocation in project network models – a survey" The j*ournal of industrial engineering*, 17(4), 177-188.

Coleman, D., Arnold, P., Bodoff, S., Dollin, C., Gilchrist, H, Hayes, F., and Jeremaes, P. (1994). *Object-Oriented development: the fusion method*, Prentice Hall, Englewood Cliffs, NJ.

El-Bibany, H. (1997). "Parametric constraint management in planning and scheduling: computational basis" *Journal of construction engineering and management*, ASCE, 123(3), 348-353.

Faniran, O. O., Love, P. E., and Li, H. (1999). "Optimal allocation of construction planning resources" *Journal of construction engineering and management*, ASCE, 125(5), 311-319.

Fischer, M., and Aalami, F. (1996). "Scheduling with computer-interpretable construction method models" *Journal of construction engineering and management*, ASCE, 122(4), 337-347.

Fischer, M., and Tatum, C.B. (1997). "Characteristics of design-relevant constructability knowledge" *Journal of construction engineering and management*, ASCE, 123(3), 253-260.

Fredlund, D. J., and de Leon, G. P. (1990). "Delay evaluation using record schedules" AACE Transactions, AACE, R.2.1-R.2.7.

Gemmill, D. D., and Edwards, M. L. (1999). "Improving resource-constrained project schedules with look-ahead techniques" *Project management journal*, PMI, 30(3), 44-55.

Hegazy, T., and Ersahin, T. (2001). "Simplified spreadsheet solutions. I: subcontractor information system" *Journal of construction engineering and management*, ASCE, 127(6), 461-468.

Hegazy, T., and Ersahin, T. (2001). "Simplified spreadsheet solutions. II: overall schedule optimization" *Journal of construction engineering and management*, ASCE, 127(6), 469-475.

Holzner, S. (1998). *Visual Basic 6 black book*, Coriolis, Scottsdale, AZ.

Jacobson, I. (1992). *Object-oriented software engineering: a use case driven approach*, Addison Wesley, Reading, MA.

Jingsheng, S., and Arditi, D. (2001). “Construction delay computation method” *Journal of construction engineering and management*, ASCE, 127(1), 60-65.

Karaa, F. A., and Nasr, A. Y. (1986). “Resource management in construction” *Journal of construction engineering and management*, ASCE, 112(3), 346-357.

Kartam, S. (1999). “Generic methodology for analyzing delay claims” *Journal of construction engineering and management*, ASCE, 125(6), 409-419.

Kähkönen, K. E. E. (1997). “Interactive decision support system for building construction scheduling” *Journal of computing in civil engineering*, ASCE, 8(4), 519-535.

Kang, M. (1991). *Network and algorithm*, HongReung Pub., Seoul, Korea.

Kang, M. (1994). *Data structure: algorithm, object-oriented, and C++*, HongReung Pub., Seoul, Korea.

Larman, C. (1998). *Applying UML and patterns: an introduction to object-oriented analysis and design*, Prentice Hall, Upper Saddle River, NJ.

Leu, S., and Yang, C. (1999). “GA-based multicriteria optimal model for construction scheduling” *Journal of construction engineering and management*, ASCE, 125(6), 420-427.

Lomax, P. (1998). *VB & VBA in a nutshell: the language*, O’Reilly, Sebastopol, CA.

Manber, U. (1989). *Introduction to algorithms: a creative approach*, Addison Wesley, Reading, MA.

Main, M., and Savitch, W. (1998). *Data structures & other object using C++,* Addison Wesley, Reading, MA.

McCullough, R. B. (1989). "CPM schedules in construction claims*" Cost engineering*, 31(5), 18-21.

Plemmons, J. K., and Bell, L. C. (1995). "Measuring effectiveness of materials management process" *Journal of management in engineering*, ASCE, 11(6), 26-32.

O'Brien, J. (1971). *CPM in construction management*, McGraw-Hill, New York, NY.

Özdama, L., and Ulusoy, G. (1995). "A survey on the resource-constrained scheduling problem" *IIE Transactions*, IIE, 27, 574-586.

Prata, S. (1991). *C++ primer plus*, Waite Group, Carte Madera, CA.

Rao, G. N., Grobler, F., and Ganeshan, R. (1997). "Interconnected component application for AEC software development" *Journal of computing in civil engineering*, ASCE, 11(3), 154-164.

Raz, T, and Marshall, B. (1996). "Effect of resource constraints on float calculations in project networks" *International journal of project management*, 14(4), 241-248.

Russell, J. S., Jaselskis, E., and Lawrence, S. P. (1997). "Continuous assessment of project performance" *Journal of construction engineering and management*, ASCE, 123(1), 64-71.

Shanmuganayagam, V. (1989). "Current float techniques for resources scheduling" *Journal of construction engineering and management*, ASCE, 115(3), 401-411.

Son, J., and Skibniewski, M. J. (1999). "Multiheuristic approach for resource leveling problem in construction engineering: hybrid approach" *Journal of construction engineering and management*, ASCE, 125(1), 23-31.

Suhail, S., and Neale, R. H. (1994). "CPM/LOB: new methodology to integrate CPM and Line Of Balance" *Journal of construction engineering and management*, ASCE, 120(3), 667-684.

Talbot, F. B. (1982). “Resource-constrained project scheduling with time-resource tradeoffs: the non-preemptive case” *Management science*, 28(10), 1197-1210.

Tavakoli, A., and Roger, R. (1990). “CPM use in ENR top 400 contractors” *Journal of management in engineering*, ASCE, 6(3), 282-295.

Thomasen, O. B., and Butterfield, L. (1993). “Combining risk management and resource optimization in project management sortware” *Cost engineering*, 35(8), 19-24.

Tommelein, I. D., Riley, D. R., and Howell, G. A. (1999). “Parade game: impact of work flow variability on trade performance” *Journal of construction engineering and management*, ASCE, 125(5), 304-310.

Winston, P. H. (1995). *On to C++*, Addison-Wesley, New York, NY.

Wright, P. (1998). *Beginning Visual Basic 6*, Wrox, Chicago, IL.

Zouein, P. P., and Tommelein, I. D. (2001). “Improvement algorithm for limited space scheduling” *Journal of construction engineering and management*, ASCE, 127(2), 116-124.

VITA

Kyunghwan Kim was born in Taegu, Korea. He received a Bachelor's degree in 1993 and a Master's degree in Construction Management in 1996 from the Department of Architectural Engineering at Hanyang University, Seoul. He earned his second Master's degree in Computer Aided Engineering and Construction from the Department of Civil and Environmental Engineering at Carnegie Mellon University in 1998. In 999, he enrolled the Department of Civil and Environmental Engineering at Virginia Polytechnic Institute and State University, to pursue his Ph.D. degree. His doctoral study focused on the improvement of scheduling techniques.