

Construction Scheduling Using Critical Path Analysis with Separate Time Segments

by

Wail Menesi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Civil Engineering

Waterloo, Ontario, Canada, 2010

© Wail Menesi 2010

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Project managers today rely on scheduling tools based on the Critical Path Method (CPM) to determine the overall project duration and the activities' float times. Such data provide important information about the degree of flexibility with respect to the project schedule as well as the critical and noncritical activities, which leads to greater efficiency in planning and control of projects.

While CPM has been useful for scheduling construction projects, years of practice and research have highlighted a number of serious drawbacks that limit its use as a decision support tool. The traditional representation of CPM lacks the ability to clearly record and represent detailed as-built information such as slow/fast progress and complete representation of work interruptions caused by the various parties involved. In addition, CPM is based on two unrealistic assumptions: that the project deadline is not restricted and that resources are unlimited. With CPM, therefore, the most cost-effective corrective actions needed in order to recover delays and overruns cannot be determined. This research is based on the view that many of the drawbacks of CPM stem from the rough level of detail at which progress data is represented and analyzed, where activities' durations are considered as continuous blocks of time.

To overcome CPM drawbacks, this research presents a new Critical Path Segments (CPS) mechanism, with its mathematical formulation, that offers a finer level of granularity by decomposing the duration of each activity into separate time segments. The CPS mechanism addresses the problems with CPM in three innovative ways: (1) the duration of an activity is represented as a series of separate time segments; (2) the representation

of the progress of an activity is enhanced; and (3) an optimization mechanism to incorporate project constraints into the CPS analysis. To demonstrate the ability of the CPS to provide better analysis than the traditional CPM, a number of case studies are used to show its ability to (1) simplify network relationships and accurately calculate floats and critical path(s); (2) achieve better resource allocation and facilitate accurate delay analysis; and (3) overcome problems associated with the use of multiple resource calendars.

This research represents a change from well-known CPM techniques and has the potential to revolutionize and simplify the analysis of ongoing and as-built schedules. The developed CPS technique is expected to help project managers achieve a better level of control over projects and their corrective actions because it offers better visualization, optimization, and decision support for meeting project goals within the specified constraints.

Acknowledgements

First and foremost, I thank God for giving me the knowledge, strength, and ability to complete this work.

Along the way, several people have contributed in a variety of ways to this research. I would like first to express my sincere appreciation and gratitude to my supervisor, Dr. Tarek Hegazy, for his invaluable guidance and for enriching my research and teaching skills over the last several years. I am deeply impressed with his wealth of knowledge, excellence in teaching, and dedication to academic research. It has been a great honor to work with him and to learn from his experience.

I am very grateful to the Libyan Ministry of Education for providing the financial support throughout my PhD program.

I would like to thank my committee members Prof. Jesús de la Garza, Prof. Carl Haas, Prof. Susan Tighe, and Prof. Otman Basir for their insightful comments.

I would also like to thank my colleagues in the construction research group for their priceless friendship, for all the great times we shared, and for providing the friendly environment that I had the pleasure to work in.

I am as ever, especially indebted to my parents, whose love and support made this work possible.

I dedicate this thesis to

My Parents

Table of Contents

Author's Declaration	ii
Abstract	iii
Acknowledgements	v
Dedication	vi
Table of Contents	vii
List of Figures	x
List of Tables.....	xiii
Chapter 1 Introduction.....	1
1.1 Background	1
1.2 Research Motivation.....	2
1.3 Research Scope and Objectives.....	5
1.4 Research Methodology	6
1.5 Thesis Organization.....	7
Chapter 2 Literature Review	9
2.1 Construction Scheduling	9
2.1.1 Bar Charts.....	10
2.1.2 Development of the Critical Path Method.....	10
2.1.3 Growth of CPM Usage.....	12
2.2 The CPM Mechanism.....	13
2.3 Resolving Practical Constraints in CPM	17
2.3.1 Resource Leveling/Allocation	17
2.3.2 Time-Cost Trade-Off Analysis.....	21
2.4 Delay Analysis	24
2.5 Drawbacks of the CPM and Existing Software.....	29
2.6 Recent Efforts to Enhance CPM.....	32
2.7 Conclusions	35
Chapter 3 Critical Path Segments (CPS) Scheduling Technique	36
3.1 Introduction	36
3.2 Areas Requiring Enhancement.....	36
3.2.1 Improving Network Representation to Avoid Complexity	36
3.2.2 Improving Representation of Activity Progress	39

3.2.3 Incorporating Constraints into the CPM.....	40
3.2.4 Enhancing Project Control and Schedule Analysis	42
3.3 Critical Path Segments (CPS).....	43
3.3.1 New Network Representation.....	43
3.3.2 New Representation of Activity Progress	46
3.4 Proof of Concept	48
3.5 Summary and Conclusions	53
Chapter 4 Mathematical Formulation of the Critical Path Segments Method.....	54
4.1 Introduction	54
4.2 New Schedule Calculation without a Backward Pass	54
4.3 CPS Calculation Procedure	57
4.3.1 Step 1: User Input of Schedule Data	57
4.3.2 Step 2: Translation of Relationships into the CPS Format	59
4.3.3 Step 3: CPS Forward Pass Scheduling Calculations	67
4.3.4 Step 4a: Float Calculation in the Case of No Resource Constraints.....	75
4.3.5 Step 4b: Float Calculation with Consideration of Resource Constraints	79
4.3.6 Step 4c: Float Refinement	81
4.4 Generalized CPS Formulation with Consideration of Progress Information	84
4.4.1 Baseline Schedule.....	84
4.4.2 Current Schedule	85
4.5 Detailed Scheduling Mechanism.....	87
4.6 Summary and Conclusions	89
Chapter 5 Resolving Project Constraints in CPS.....	90
5.1 Introduction	90
5.2 Resolving Constraints in CPM (Activity Level)	90
5.3 Resolving Constraints in CPS (Time-Segment Level).....	93
5.4 Schedule Optimization for Project Control (Activity Level)	94
5.4.1 Resolving Resource Constraints.....	94
5.4.2 Meeting Deadline Duration	96
5.5 Generalized Optimization Formulation for CPS	97
5.6 Evolutionary Optimization	102
5.6.1 Solution Structure (Activity Level).....	104
5.6.2 CPS Solution Structure (Time-Segment Level)	104

5.6.3 Generating an Initial Population.....	105
5.6.4 Evaluation Criterion	105
5.7 Example of Schedule Optimization.....	106
5.8 Summary and Conclusions	113
Chapter 6 Validation: Case Studies	115
6.1 Introduction	115
6.2 Better Resolution of Constraints	115
6.3 Efficient Project Control and Delay Analysis	118
6.4 Summary and Conclusions	120
Chapter 7 Conclusions and Future Research.....	121
7.1 Summary	121
7.2 Discussion of the CPS Approach	123
7.2.1 Issues Related to Planning.....	124
7.2.2 Issues Related to Project Control.....	125
7.3 Contributions	126
7.4 Future Research.....	128
Appendix A	130
Evolutionary Algorithms	130
A.1 Genetic Algorithms (GAs).....	131
A.1.1 Fitness Normalization.....	133
A.1.2 Selection	134
A.1.3 Crossover	135
A.1.4 Mutation	136
A.1.5 Elitism.....	137
References	138

List of Figures

Figure 2.1: Types of activity relationships	11
Figure 2.2: Forward pass calculations in a CPM network.....	14
Figure 2.3: Backward pass calculations and the critical path of the network.....	15
Figure 2.4: Early bar chart for the project	16
Figure 2.5: Effect of resource allocation	20
Figure 2.6: Project schedule after the duration of activity E is increased by two days	21
Figure 2.7: Project schedule after resource over-allocation is resolved	21
Figure 2.8: Project time-cost relationship.....	22
Figure 2.9: Bar charts for a small example of a daily windows analysis	26
Figure 2.10: Daily windows analysis showing the window for day 3.....	27
Figure 2.11: Daily windows analysis showing the window for day 4.....	28
Figure 2.12: Daily windows analysis showing the window for day 5.....	28
Figure 2.13: Example of a project network based on Herold's (2004) approach	33
Figure 2.14: Sample schedule using LDM notation (Ponce de Leon 2008).....	34
Figure 3.1: Case study showing that complex relationships can create partially critical activities	38
Figure 3.2: Two methods for calculating the remaining duration of an activity that are not clearly indicated in existing tools.....	40
Figure 3.3: New representation of activities as separate time segments to avoid complex relationships and lag times	44
Figure 3.4: CPS representation accurately identifies the critical segments of the activities	45
Figure 3.5: Time-based and production-based options in CPS representation.....	46
Figure 3.6: CPS representation of progress events.....	47
Figure 3.7: Traditional MS Project Schedule for Case 1.....	49
Figure 3.8: Simulated schedule with separate time segments	49
Figure 3.9: Possible early start times under multiple calendars	51
Figure 3.10: Simulating CPS for Case 2	52
Figure 4.1: A simple case study illustrating the new approach for total float calculation.....	55
Figure 4.2: Flowchart for the CPS model.....	58
Figure 4.3: CPS representation of an FS relationship with a zero lag	60
Figure 4.4: CPS representation of an FS relationship with a lag time.....	60
Figure 4.5: CPS representation of an SS relationship with a zero lag	61

Figure 4.6: CPS representation of an SS relationship with a lag time.....	61
Figure 4.7: CPS representation of FF relationships.....	62
Figure 4.8: CPS representation of an SF relationship with a zero lag.....	63
Figure 4.9: CPS representation of a SF relationship with a lag time.....	64
Figure 4.10: The purpose of hard links when FF or SF relationships are converted.....	65
Figure 4.11: Converting SS and FF relationships to a production-based relationship	66
Figure 4.12: AON network and its CPS equivalent for the case study.....	71
Figure 4.13: Resource allocation for the case study using the GRD rule.....	74
Figure 4.14: Example of the CPS total float determination process.....	76
Figure 4.15: Calculation of path floats after resource allocation.....	79
Figure 4.16: CPS total float refinement with consideration of resource constraints	81
Figure 4.17: Total float calculation in the case of resource constraints	82
Figure 4.18: Simulation of the float refinement process	83
Figure 4.19: Different types of schedules.....	86
Figure 4.20: Portion of the schedule that represents flexibility for corrective action	87
Figure 5.1: Case study activities and their optional estimates.....	91
Figure 5.2: Details of a solution that meets all constraints.....	92
Figure 5.3: CPS representation of resource leveling and TCT decisions	93
Figure 5.4: Optimization process using evolutionary algorithms.....	103
Figure 5.5: Solution structure for activity-level optimization	104
Figure 5.6: Solution structure	105
Figure 5.7: Project network of a case study example	106
Figure 5.8: General data for the sample project	107
Figure 5.9: Schedule and resource utilization profiles for the case study	108
Figure 5.10: MS Project solution to the over-allocations of R1 and R3.....	109
Figure 5.11: EasyPlan solution to resolve project constraints.....	111
Figure 5.12: Simulated CPS schedule on EasyPlan	112
Figure 5.13: EasyPlan solution using simulated CPS schedule.....	113
Figure 6.1: Traditional schedule and resource profile before resource leveling	115
Figure 6.2: MS Project solution to resource over-allocation (without splitting).....	116
Figure 6.3: MS Project solution to resource over-allocation when splitting is allowed	116
Figure 6.4: Project schedule and resource profile when activities are segmented	117
Figure 6.5: As-planned schedule for a case study (13 days)	119

Figure 6.6: Simulated owner delay on day 3 causing resource over-allocation on day 7	119
Figure 6.7: Resolving the resource over-allocation results in a one-day delay	119
Figure 6.8: Resolving the resource over-allocation using CPS	120
Figure A. 1: Population, gene and chromosome in GA.....	132
Figure A.2: Genetic algorithm process (Lin and Lee 1996).....	133
Figure A.3: Weighted roulette wheel (Lin and Lee 1996).....	134
Figure A.4: Crossover operator to generate offspring genes.....	136

List of Tables

Table 2.1: Existing techniques for Time-Cost Trade-Off analysis.....	23
Table 4.1: Example of activity data in the proposed model	58
Table 4.2: CPS resource allocation for the case study	72
Table 4.3: Calculation of path float	77
Table 4.4: Activity total floats based on the path floats	78
Table 4.5: Activity total float values after resource allocation.....	80
Table 4.6: Refined total float values after resource-related constraints are considered	83
Table 5.1: Activities data for the example.....	107

Chapter 1

Introduction

1.1 Background

Since its introduction in the late 1950s, the Critical Path Method (CPM) has proven to be a useful tool for planning and controlling construction projects. CPM enables project managers to evaluate the early and late times at which activities can start and finish, to calculate activity float (slack), to define critical activities, and to evaluate the impact of changes in duration and logical relations on the overall project duration.

Because of its benefits and the significant advancements that have been made in both computer hardware and scheduling software, the use of the CPM and its variation, the precedence diagram method (PDM), in all industries, including construction, has dramatically increased in the last three decades (Liberatore et al. 2001). For the purposes of this research, CPM is used to indicate both CPM and PDM. In construction projects, CPM is very important because it enables the contractor to determine when and how many resources are needed, vendors to determine when to deliver materials, and subcontractors to determine when they can perform their work. However, CPM has serious limitations that have yet to be overcome. The analytical capabilities and computing efficiency of CPM also need to be enhanced in order to meet the changing requirements of the construction industry (Ahuja and Thiruvengadam 2004).

Construction involves unique environments, challenges, and project management needs, not found in other industries. While the industry includes many large companies, statistics

indicate that more than two-thirds of construction firms have fewer than five employees (Halpin and Woodhead 1998). The majority of these small firms are specialist subcontractors working with a general contractor. This category of firms experiences the highest level of business failures, as reported in a survey by Russell and Radtke (1991). The survey identified the factors that contribute to failure, including underbidding, insufficient cash flow, external difficulties, and lack of experience in estimating and monitoring costs. These factors, in essence, indicate lack of efficient project management, which is in part due to the drawbacks associated with CPM, particularly the lack of direct mathematical formulation for satisfying project constraints such as deadlines and resource limits. Despite the many practical insights provided by professional organizations and commercial software, to many construction professionals, particularly trades and small contractors, the use of CPM and project management tools does not extend beyond creating a schedule with a neat appearance in order to satisfy contract requirements (Baweja 2006).

1.2 Research Motivation

Despite the long history and expanding use of CPM, the literature indicates that CPM has a number of drawbacks that raise concerns about its use in the construction industry (Galloway 2006). The reasons for the lack of faith in CPM, which also represent the motivation for this research, can be described as follows.

Inadequate Planning before Construction: In the planning stage before construction, the CPM algorithm is based on two unrealistic assumptions: that the project deadline is not restricted and that resources are unlimited. To account for practical considerations, a project

manager must apply a variety of techniques, which are discussed in chapter 2, such as time-cost trade-off analysis, resource allocation, resource leveling, and cash-flow management. These techniques, however, deal with distinct sub-problems and thus can be applied to a project only one after another, rather than simultaneously (Hegazy 2002). While these techniques can improve CPM scheduling, they also render the process longer and less comprehensible. In addition, most construction projects are subject to multiple constraints, and it is therefore often difficult to produce a realistic schedule because a solution to one constraint (e.g., resource limits) may interfere with the solution to another (e.g., deadline). This difficulty adds to the perception that CPM and existing software are useful only for presentation purposes. Because of the lack of adequate procedures and models for resolving all constraints combined, existing software tools focus instead on enhancing technology-related aspects, such as web collaboration, rather than on features that address basic engineering and project management problems.

Inadequate Decision Support during Construction: Once a project has begun, the schedule becomes essential to the successful coordination of day-to-day activities and acts as a baseline for measuring progress. When accurate site events are recorded and entered into the schedule, CPM analysis can help project managers anticipate problems that may occur in the future (Gould 2005). Using the difference between actual and planned progress, management can initiate appropriate corrective actions, such as replanning, rescheduling, or increasing the level of resources. This dynamic cycle of reviewing the current status and forecasting future requirements is one of the primary purposes of project control (Ahuja and Thiruvengadam 2004).

Although the CPM might be useful for updating activity status, actual progress data are recorded on the schedule mainly in the form of a cumulative percentage complete for activities, without much detail about interruptions caused by any party or about other slow or fast progress times. These drawbacks prevent the use of CPM for accurately analyzing project delays. In addition, CPM incorporates no mechanism to support decision making related to determining the most cost-effective corrective actions for recovering delays and overruns. Project managers currently must employ iterative cycles of changes in order to manually alter the schedule from different angles, which is a slow trial-and-error process that does not guarantee good solutions.

Inadequate Analysis after Construction: In addition to being an essential tool for project scheduling, CPM analysis also plays an important role in the analysis of final as-built schedules so that the responsibility of each party for any delays experienced during the project can be determined. The boards of contract appeals and the courts have shown their willingness to utilize CPM network analysis as a mean of identifying the source of any delays in construction projects (Ostrowski 2006). However, the complex features of commercial scheduling software, such as multiple resource calendars, make CPM schedules difficult to analyze. Among the well-documented factors that contribute to the difficulty of analyzing CPM schedules are the complex relationships among activities, the use of lead and lag times in relationships (Herlod 2004), and the inadequate representation of site events in CPM scheduling software. Accurate analysis of construction schedules is also often challenging because of the lack of a detailed delay analysis mechanism that considers

multiple baselines and daily events caused by the different parties (Hegazy and Menesi 2008b).

1.3 Research Scope and Objectives

The goal of this research was to develop an innovative scheduling model that overcomes the current CPM drawbacks. The model will better handle schedule constraints, such as project deadline and resource limits; facilitate corrective actions during construction; and produce accurate schedule analysis during and after construction. The following were the detailed objectives:

- Identify the practical areas of potential improvement that can enhance the representation and formulation of critical path analysis.
- Develop a new critical path analysis model that is based on segmented activity durations and examine the ability of the new critical path segments (CPS) mechanism to
 1. Provide a better representation of mid-activity events,
 2. Better identify critical path fluctuations,
 3. Represent the various activity relationships more simply, and
 4. Enhance the resolution of project constraints.
- Based on the new CPS model, formulate a schedule optimization procedure to serve as a decision support system that considers all project constraints collectively.
- Simulate the CPS on existing scheduling software tools and experiment with a number of case studies in order to validate the CPS and demonstrate its benefits.

1.4 Research Methodology

The methodology for achieving the above objectives was as follows:

- **Extensive Review of CPM Drawbacks and Suggested Improvements:** An extensive survey of the literature was carried out in order to examine existing CPM procedures and to identify the limitations that prevent CPM from satisfying the changing requirements of the construction industry.
- **Development of New Representation to Avoid Complexity in Schedules:** Based on the literature review of potential improvements, a new representation of the relationships among the activities was developed so that a schedule becomes simple enough for field personnel to use and easier for project parties to understand.
- **Enhancement of Project Control and Schedule Analysis:** To make the schedule a useful tool for suggesting corrective actions and identifying delays and accelerations, a new representation of project activities was introduced. This representation is capable of more accurately determining the critical path, thus facilitating better schedules and more effective corrective actions.
- **Handling Project Constraints:** A generic representation of project decisions was formulated considering the new representation of project activities and their relationships. Based on this new representation of decisions, an optimization model for schedule optimization was developed.

- **Validation:** Once the mathematical formulation for the new critical path analysis was completed, simple case studies were used in order to validate the CPS and demonstrate its functionality and usefulness:
 - A case study that showed the ability of CPS to accurately define the critical path
 - A case study that proved the benefits of CPS in avoiding errors when multiple calendars are used
 - A case study that demonstrated the ability of CPS to provide a better resolution of project constraints than traditional CPM
 - A case study that proved the benefits of CPS in project control and schedule analysis

1.5 Thesis Organization

The remainder of the thesis is organized as follows:

Chapter 2 presents a detailed literature review of the use of CPM in the construction industry. Numerous researchers and practitioners have studied CPM and reported both benefits and criticisms. A list of the most important critical views of CPM and the pitfalls inherent in commercial software is presented along with a description of the recent efforts of researchers to enhance CPM.

Chapter 3 provides a detailed description of the main areas of necessary enhancements to current critical path analysis. A new critical path model that can address these needed

enhancements in a collective manner is then presented. The critical path segments (CPS) representation of the project network and activity progress is described. Two case studies are then used to show the ability of CPS to accurately define the critical path and avoid errors when multiple calendars are used.

Chapter 4 presents the detailed mathematical formulation of the CPS mechanism and comments about its divergence from traditional CPM. The CPS calculation process is described, along with illustrative examples. The CPS approach for calculating accurate total float values in the case of resource-constraint scheduling is also introduced, and the detailed CPS formulation for progress analysis is then described.

Chapter 5 presents the development of an optimization model for CPS. A simple approach for facilitating the resolution of multiple constraints at the activity level is illustrated through an example, and the reformulation of this approach to suit CPS networks is then explained.

Chapter 6 focuses on the validation of the CPS. Two case studies are presented in order to demonstrate the usefulness of CPS. One case study shows that CPS can provide better resource allocation than traditional CPM, while the other shows the ability of CPS to analyze project delays accurately.

Chapter 7 summarizes the research, highlights its contributions, and lists recommendations for future research.

Chapter 2

Literature Review

2.1 Construction Scheduling

Schedules are key documents in the management of construction projects. A project schedule establishes the start date, duration, completion date, and resource needs for each activity in the project. Mistakes in the schedule may cause the project team to allocate resources to the wrong place at the wrong time or may prevent the parties from accurately assessing whether the project is ahead of or behind schedule (Ackley et al. 2007). Knowing precisely when an activity is going to begin also has substantial cost implications. For example, rental of a large crane can cost more than \$5,000 per week, so if the duration of a project is miscalculated, contractors can quickly consume in rental charges any profit they might hope to earn from a job. In addition, the contractor's overhead is dependent on how long the project is expected to take (Gould 2005).

Scheduling the construction process is essential not only so that projects can be completed profitably and on time, but also so that any delays can be evaluated in order to prove entitlement to time and cost compensation. As problems are encountered, the schedule helps project managers rearrange project tasks and resources so that they can meet the primary objectives of time, cost, and quality under limited resource and budget constraints. Although bar charts (Gantt charts) have been used as a simple scheduling method, network schedules that employ the critical path method (CPM) are now much more widely used. This is because of the fact that network analysis can show which activities are critical and which are not.

Almost all project management software is therefore based on critical path analysis, which is the focus of this research.

2.1.1 Bar Charts

The bar or Gantt chart was introduced in the 1920s by industrialist and management consultant Henry Laurence Gantt. Since then, bar charts have been used extensively for planning and monitoring construction projects, such as the construction of the Hoover Dam (Lowsley and Linnett 2006). The usual format for a bar chart is for the activities to be listed in a vertical column on the left-hand side of the chart, with a horizontal bar for each activity plotted against a timescale to mark the start and finish times of the activities. Although this format is simple and effectively communicates the necessary information, the use of such bar charts has limitations with respect to updating the schedule as the project progresses because no consideration is given to the logical relationships among the activities. This major drawback prevents simple bar charts from reacting dynamically when changes are made to the schedule. Modern scheduling software enables critical path analysis to be displayed in a linked bar chart format that overcomes some of the problems associated with simple bar charts.

2.1.2 Development of the Critical Path Method

The critical path method (CPM) was developed in the late 1950s by researchers at the E. I. Du Pont de Nemours Company. When first developed, the traditional form of CPM networks was termed an AOA or "activity on arrow" diagram, which allows only Finish-to-Start relationships among the activities. This means that activities can not be overlapped and that all preceding activities must be completed before a current activity can start.

With the introduction of the Precedence Diagram Method (PDM), more flexibility regarding activity relationships has been added while the schedule calculations still utilize CPM analysis. In precedence networks, an activity can be connected from either its start or its finish, which in addition to the traditional Finish-to-Start relationship, allows the use of three additional relationships between project activities: Start-to-Start, Finish-to-Finish, and Start-to-Finish (Figure 2.1).

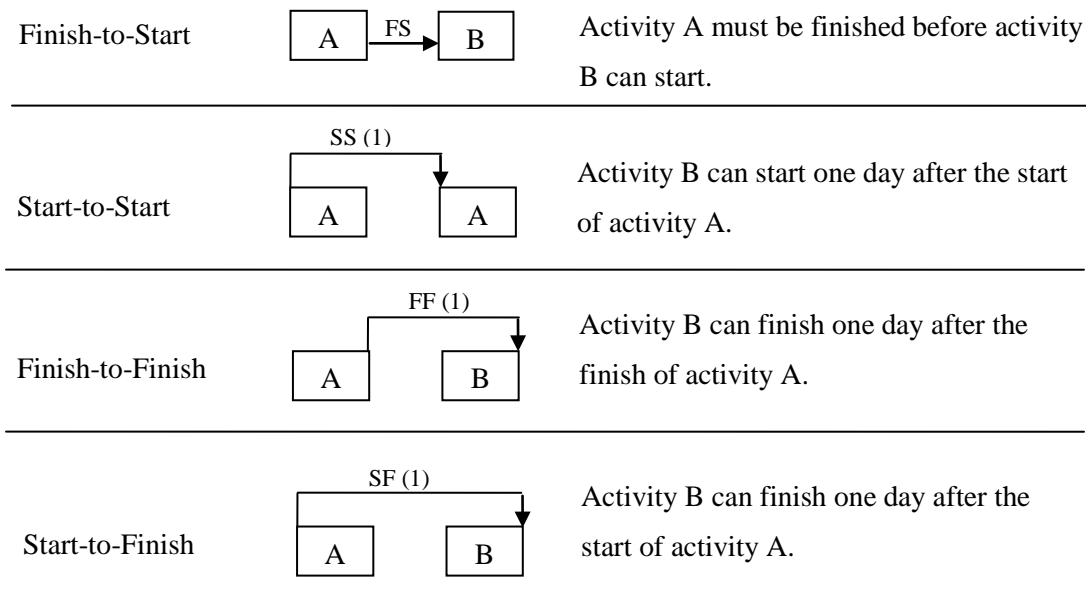


Figure 2.1: Types of activity relationships

Another characteristic of PDM diagrams is that periods of time can be assigned between the start and/or finish of one activity and the start and/or finish of a succeeding activity. These periods of time between the activities are referred to as leads and lags. A lead is the amount of time by which an activity precedes the start of its successor(s), and a lag is the amount of time delay between the completion of one task and the start and/or finish of its successor.

Most commercial software, such as Primavera Project Planner and Microsoft Project allow the use of non-traditional relationships with lags.

2.1.3 Growth of CPM Usage

Several surveys have demonstrated that, over the years, CPM use has been growing in the construction industry. Kelleher (2004) analyzed the data from three surveys, conducted in 1974, 1990, and 2003, that investigated how Engineering News Record's (ENR) top 400 contractors use CPM. The study revealed a growing CPM use that reached 98% of respondents in 2003. Given the mix of both large and small contractors, a recent survey by Hawkins (2007) proved that it is not only the large ENR 400 firms who utilize CPM to manage their projects, but also small and mid-size construction firms. All respondents indicated that they used CPM scheduling at least some of the time, with 45% reporting they used it all of the time and another 40% reporting they used it most of the time.

The main uses of CPM were reported as planning before construction, control during construction, and claim analysis. The disadvantages of CPM were also reported as logic abuse, too much dependency on specialists, implementation requiring excessive work, and not lack of responsiveness to the needs of field personnel. These findings agree with the arguments presented in Chapter 1 and also with the results of a more recent survey (Galloway 2006): CPM has not gained the trust of the industry as a project control tool. This statement is true because in spite of the fact that contractors can report that they used CPM for project control, as indicated in Kelleher's (2004) survey. They may find it useful for

updating activity data and analyzing progress status but not as beneficial in supporting other important aspects, such as corrective actions and recovering execution problems.

2.2 The CPM Mechanism

To perform critical path analysis, the activities that make up the project are first identified. A project network is then used to represent the precedence relationships, according to which each activity can have a group of predecessors and can be followed by a group of successors. Once the project network is drawn, as shown in the example in Figure 2.2, the following steps are performed (Hegazy 2002):

- A forward pass to determine the early start times of the activities
- A backward pass to determine the late finish times of the activities
- Float calculations
- Identification of critical activities

The forward pass calculations start at the beginning of the project and move to the end of the project, or from left to right. As shown in Figure 2.2, the early start time (ES) is noted in the upper left corner and the early finish time (EF) is noted in the upper right corner of the node that represents an activity. The calculations begin from the left-most node in the network, which is assigned an early start time of zero. Since all activity times use an end-of-day notation, the early start of "site preparation" being zero means that the activity starts at the end of day zero or the beginning of day 1. Adding the duration of the first activity (3 days) to its ES (end of day 0) results in the early finish of that activity (end of day 3). The network shows that the excavation of both trench 1 and trench 2 can start as soon as the site

preparation has finished; therefore, the next step is to transfer the EF of the predecessor activity to the ES of the successor activity. Adding the duration of each activity to its ES gives an EF day for excavating trench 1 at the end of day 9 and for excavating trench 2, the end of day 7, as shown in Figure 2.2.

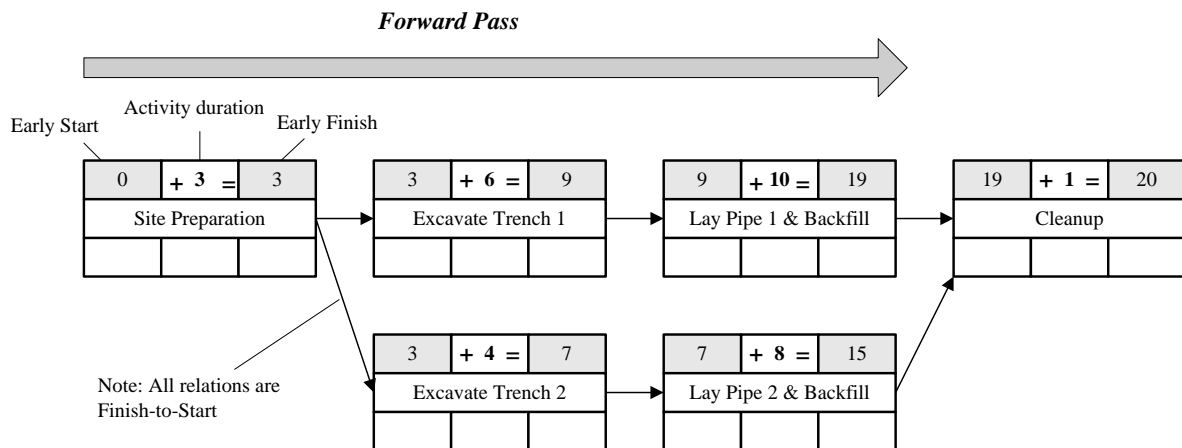


Figure 2.2: Forward pass calculations in a CPM network

If two or more activities are predecessors to a single activity, the one with the largest EF is chosen to insert into the successor activity. As shown in Figure 2.2, the ES of the "Cleanup" activity is the largest EF of the predecessor activities: 19, not 15. Adding the duration of this activity to this ES (19) then results in an EF of 20 for the final activity. The project is therefore scheduled to finish at the end of day 20.

The backward pass calculations start at the end of the project and move to the beginning of the project, or from right to left. As shown in Figure 2.3, the late start time (LS) is noted in the lower left corner and the late finish time (LF) is noted in the lower right corner. Starting from the last activity (right-most node) in the network, the EF of the last activity "cleanup" is

transferred to become that activity's LF. Subtracting the activity's own duration, the LS is then calculated as 19 and entered in the bottom left cell of the activity box. Moving backward to the predecessors of the last activity, the LS time of a successor is copied to the LF of its predecessors. The LS of each activity is then calculated as shown in Figure 2.3.

During the backward pass, when two or more successor activities back into a single activity, the one with the shortest LS time is chosen to become the LF of the predecessor activity. For example, both "Excavate Trench 1", which has an LS of 3, and "Excavate Trench 2", which has an LS of 7, back into "Site Preparation". In this case, the LF of "Site Preparation" becomes 3, the smallest. When the backward pass is complete, the ES and the LS of the first activity should both be zero, as shown in Figure 2.3.

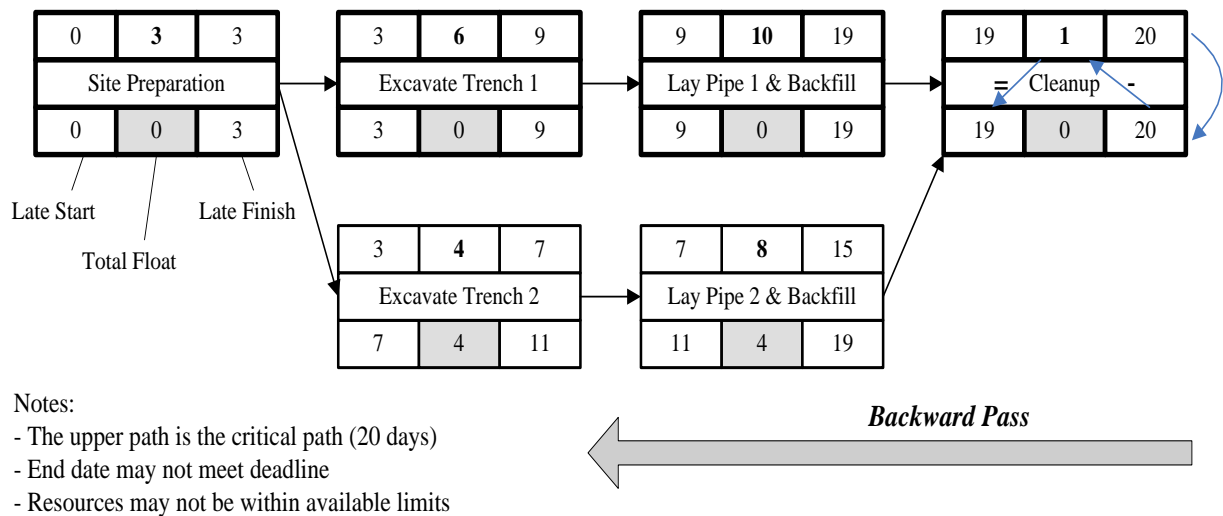


Figure 2.3: Backward pass calculations and the critical path of the network

Once the forward pass and backward pass are finished, the total float (TF) of each activity can be calculated as the difference between the early start and the early finish or the difference between the late start and the late finish time. The mathematical formula for the total float is

$$Total\ Float = LS - ES = LF - EF \quad (2.1)$$

The total float is calculated for each activity and entered in the bottom center cell of the activity box, as shown in Figure 2.3. The calculation of total float, also called the total slack, is important because it determines the flexibility of an activity: how much it can be delayed. For example, the activity "Excavate Trench 2" has an ES of 3 and an LS of 7, indicating a four-day total float, which means that "Excavate Trench 2" can be delayed by up to four days without delaying the completion of the project. On the other hand, activities with zero floats, such as "Excavate Trench 1", are called critical activities because any delay in these activities causes a delay in the project duration. Critical activities form a continuous path that spans the network from beginning to end. This path is the longest path in the network and is called the critical path of the project. Figure 2.3 identifies the critical path activities with bolded activity boxes, and Figure 2.4 shows the early bar chart for the project.

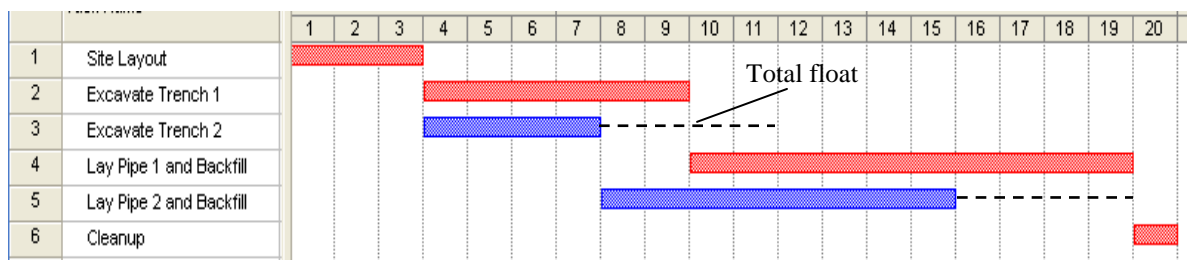


Figure 2.4: Early bar chart for the project

2.3 Resolving Practical Constraints in CPM

As illustrated in the previous section, consideration of resources is not incorporated in the formulation of the forward and backward pass calculations. In addition, CPM formulation does not incorporate a deadline that will constrain the project duration (Hegazy 2002). Thus, other techniques must be applied separately after the analysis in order to deal with deadlines and limited resources. However, applying such techniques poses several problems, as discussed in the following subsections. In addition, the solution to one constraint, e.g., resource limits, may violate the solution to another, e.g., the deadline.

2.3.1 Resource Leveling/Allocation

CPM assumes that the resources required for activities are unlimited, while in most practical situations, resources are available only in limited amounts, particularly when resources are used for multiple activities or even for multiple projects (Lu and Li 2003). The problem is that once resources are considered in the scheduling process, the accuracy of the total float calculation is lost (Bowers 1995, Fondahl 1991, Kim and de la Garza 2003). As previously mentioned, total float is the amount of time an activity can be delayed without affecting the project completion date. In traditional CPM analysis, activities with a total float of zero are identified as critical, and these activities form the critical path of the project schedule. Total float is important in construction scheduling and control because it directs the contractor to pay more attention to critical activities whose total float is zero. In addition, the total float of an activity is very important in delay analysis that is undertaken in order to determine the impact on the project completion date of any delays or slow progress. For resource-constrained projects, the backward pass CPM calculation may produce incorrect total floats

because the sequence of some activities relies not only on the logical relationships but also on resource dependencies (Kim and de la Garza 2003). With the traditional critical path method, resource-critical activities can thus not be identified, and an incorrect critical path is therefore produced.

2.3.1.1 Challenges in Resource Allocation Algorithms

A number of studies have focused on the problem of identifying the actual total floats in resource-constrained projects (e.g., Bowers 1995, Kim and de la Garza 2003, Lu and Li 2003, Wiest 1964, Woodworth and Shanahan 1988). These studies adopted the approach of creating resource-constrained links between the activities in addition to the existing logical relationships in the original CPM schedule. However, the algorithms proposed in these studies do not provide dynamic features in resource links that can reflect schedule changes (Kim and de la Garza 2005a).

Lu and Lam (2008) investigated the resource scheduling functions of Primavera Project Planner (P3) software. P3's limitations with respect to the use of SF relationships under resource constraints were identified: overestimated total floats are produced, and incorrect dates are generated. In an attempt to overcome these limitations, Lu and Lam (2008) introduced a new approach for determining the total float for each activity by observing the effect of extending the duration of each activity on the project duration. However, this approach is also inaccurate because it accounts for only one type of activity delay. An activity can be delayed due to either inability to start or slow progress (increase in duration). Each of these reasons has a different impact on the schedule and on the total duration of the

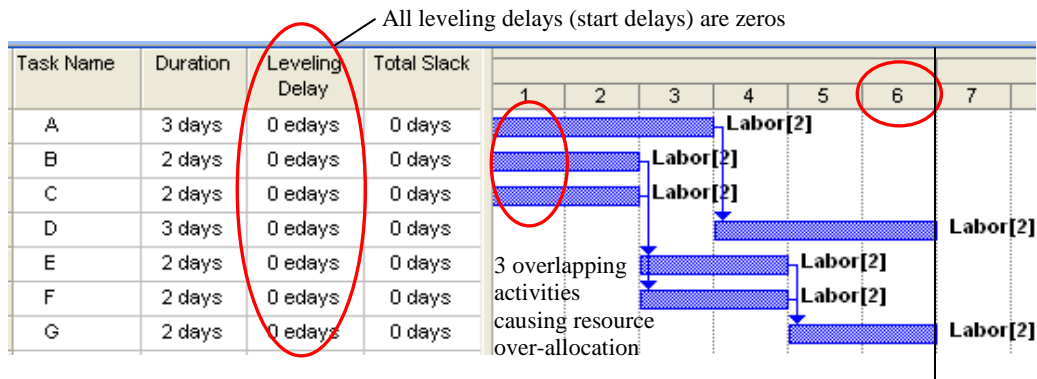
project. When the activity is delayed due to slow progress, e.g., low productivity, the resource would be in use by that activity, and the contractor would be unable to utilize the resource in another activity until the current activity is finished. On the other hand, delaying the start of an activity releases the scarce resource so that the contractor can use it to execute another activity in order to minimize the effect of that delay on the total project duration.

The Microsoft Project (MS Project) software package is one of the packages most commonly used by construction managers (Galloway 2006; Liberatore et al. 2001), so Microsoft Office Project Professional 2003 was used to highlight the problem of producing an incorrect critical path in resource-constrained projects.

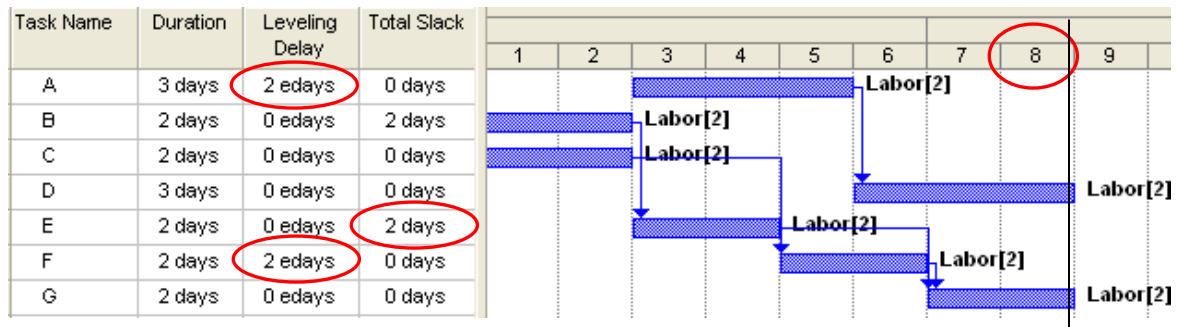
Figure 2.5a shows the as-planned schedule of a case study with a six-day duration. In this simple example, each activity needs two laborers per day (shown beside the activity bars), but the contractor has a limit of four laborers per day. The adjusted schedule (Figure 2.5b) shows how the contractor changed the start time of activities A and F (using leveling-delay values) to avoid resource over-allocation. This adjustment resulted in project duration being extended by eight days.

As shown in Figure 2.5b, the total float for activity E is determined by MS Project as two days. This total float value means that even if activity E is extended by two days, the project duration will not be affected. However, if the duration of activity E is increased by two days, as shown in Figure 2.6, the resource would then be over-allocated on days 5 and 6, and the project would have to be rescheduled in order to meet the resource limits. After rescheduling, the project duration would increase from eight to nine days, as shown in Figure 2.7. Activity

E should therefore not be considered to have 2 days of float because it is a resource-critical activity. These 2 days of float are referred to by Kim and de la Garza (2003) as Phantom Float.



a) Initial schedule that includes a resource problem



b) Adjusted schedule after resource limits are considered

Figure 2.5: Effect of resource allocation

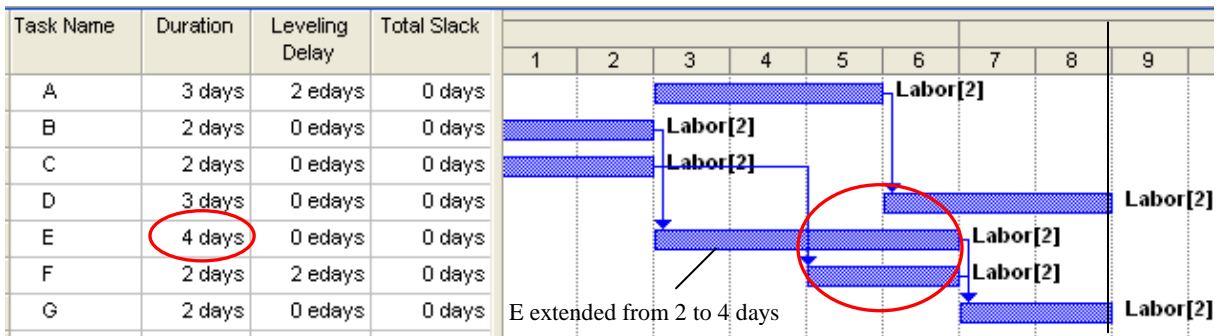


Figure 2.6: Project schedule after the duration of activity E is increased by two days

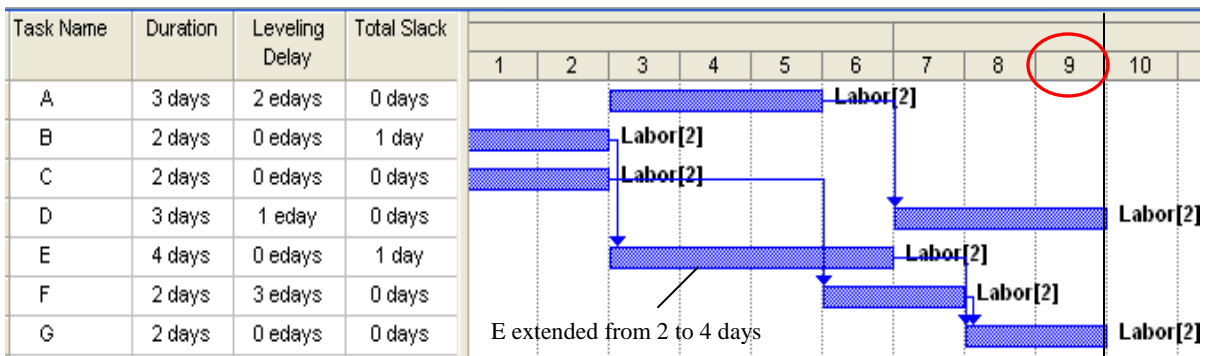


Figure 2.7: Project schedule after resource over-allocation is resolved

This case study shows that the total float determination in MS Project is based on precedence relationships among the activities without considering the resource dependencies between them, resulting in incorrect total floats and, consequently, an incorrect critical path.

2.3.2 Time-Cost Trade-Off Analysis

Time-Cost Trade-off (TCT) analysis is a technique used to overcome CPM's lack of ability to confine the schedule to a specified duration. The objective of the analysis is to reduce the original CPM duration of a project in order to meet a specific deadline with the minimum cost (Chassiakos and Sakellaropoulos 2005). TCT analysis is an important management tool because it can also be used to accelerate a project so that delays can be recovered and

liquidated damages avoided. The project can be accelerated through the addition of resources, e.g., people or equipment, or through the addition of work hours to crash critical activities. Reducing project duration therefore results in an increase in direct costs, e.g., the cost of materials, labor, and equipment. The increase in direct cost expenditures, however, can be justified if the indirect costs, e.g., expenditures for management, supervision, and inspection, are reduced or if a bonus is earned (Gould 2005).

TCT analysis involves selecting some of the critical activities in order to reduce their duration through the use of a faster construction method, even at an additional cost. Different combinations of construction methods for the activities can then be formed, each resulting in a specific project duration and direct cost. To determine the optimum TCT decision for the project, the direct cost and indirect cost curves are plotted individually so that the total cost curve can be developed from the addition of these two components, as shown in Figure 2.8. The minimum point on the total cost curve represents the set of optimum combination of construction methods for the activities. However, for projects that involve a large number of activities with varying construction options, finding optimal TCT decisions becomes difficult and time consuming (Zheng et al. 2004).

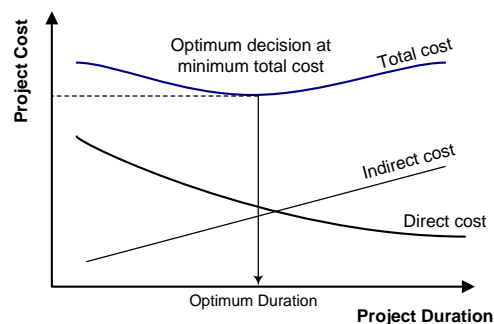


Figure 2.8: Project time-cost relationship

In the literature, three major approaches have been used for solving TCT problems: mathematical programming models, heuristic approaches, and genetic algorithms. Hegazy (2002) compared the advantages and drawbacks of these techniques, as shown in Table 2.1.

Table 2.1: Existing techniques for Time-Cost Trade-Off analysis

Techniques for Time-Cost Trade-Off Analysis		
Heuristic Methods	Mathematical Programming Models	Genetic Algorithms
Description: Simple rules of thumb	Linear Programming, Integer Programming, or Dynamic Programming	Optimization search procedures that mimic natural evolution and reproduction
Advantages: - Easy to understand - Provide good solutions - Used for large projects	- May provide optimal solutions	- Robust search algorithm - Can use discrete relationship between time and cost - Applicable to large problems
Drawbacks: - Lack mathematical rigor - Do not guarantee optimal solutions - Mostly assume linear, rather than discrete relationship between time and cost	- Difficult to formulate - Gradient-descent approach that often terminates in local minimum. - Applies to small problems only - Mostly assume linear, rather than discrete relationship between time and cost	- Random search that is time consuming - Cannot tell when or if an optimal solution is obtained
Examples: Prager 1963 Siemens 1971 Moselhi 1993	Kelly 1961 Liu et al. 1995 Chassiakos et al. 2000 Moussourakis and Haksever 2004 Chassiakos and Sakellaropoulos 2005	Feng et al. 1997 Li et al. 1999 Lu and Li 2003 Senouci and Eldin 2004 Zheng et al. 2004 Jaskowski and Sobotka 2006 Eshtehardian et al. 2008 Rogalska et al. 2008 Zahraie and Tavakolan 2009

2.4 Delay Analysis

Delays happen in most construction projects, both simple and complex. The causes of project delays include: design changes, poor weather conditions, labor actions, and mistiming of deliveries. To recover the damage caused by delays, both the delays and the parties responsible for them should be identified. However, delay situations are complex because multiple delays can occur concurrently and because they can be caused by more than one party or by none of the principal parties. As well, one delay may contribute to the formation of other delays (Arditi and Pattanakitchamroon 2006). The analysis of these delays involves not only the calculation of the delay time but also the identification of the root causes and the responsibility for the delays. Such an analysis then becomes a basis for the financial calculations that determine penalties or other damages to be assigned to the parties responsible for the delays.

Researchers and practitioners have used many techniques to assess project delays and apportion delay responsibility among the parties involved. However, different analysis techniques can provide different results for the same circumstances depending on the time and resources available for the analysis and the accessibility of project control documentation. The same technique may also yield inconsistent results when the points of view of different parties are considered (Hegazy and Zhang 2005).

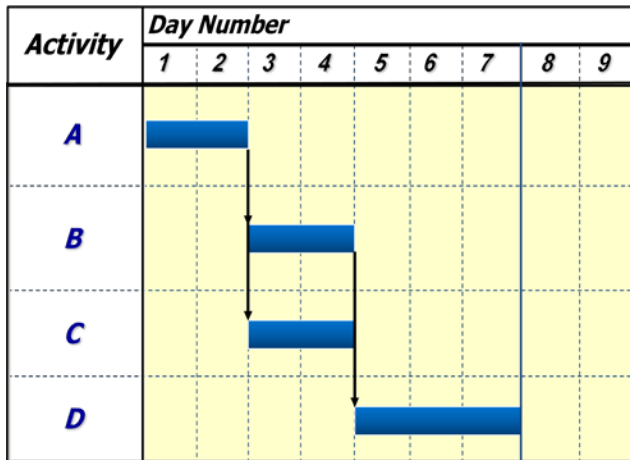
Of the methods available, windows delay analysis is recognized as the most credible method, and it is one of the few techniques much more likely to be accepted by the courts than any other method (Arditi and Pattanakitchamroon 2006, Finke 1999, Hegazy and Zhang 2005,

Kartam 1999, Stumpf 2000). Windows analysis breaks the project into a number of sequential periods, called windows, and analyzes successively the delays that occurred in each window. Despite its benefits, windows analysis can produce different results depending on the window size, it does not consider owner and contractor acceleration, it does not systematically consider the impact of several baseline updates due to changes in the duration and logical relationships of the activities, and it does not consider the impact of the progress of events on resource over-allocation and its consequent delays (Hegazy and Menesi 2008a).

Hegazy and Zhang (2005) introduced changes to the traditional windows analysis method in order to resolve some of its drawbacks. They proposed using a daily window size that would accurately take into consideration slowdowns, accelerations, work stoppages, and changes in the critical path(s). They utilized an intelligent bar chart (IBC) to represent information about progress and delays that occur as a project evolves.

Daily windows analysis can be demonstrated by an example reported in Hegazy and Zhang (2005). Figure 2.9 shows the as-planned and the as-built schedules of a simple 4-activity case study. According to the relationships shown, activities B and C both follow activity A and are then followed by activity D. The as-planned duration is seven days, while the as-built duration is nine days; the project delay is thus two days. Generally, the letters (o), (c), and (n) on an activity bar chart represent the responsibility of the party indicated (o = owner, c = contractor, n = neither) for work stoppages on a given day for a specific activity.

(a) As-planned bar chart



(b) As-built bar chart

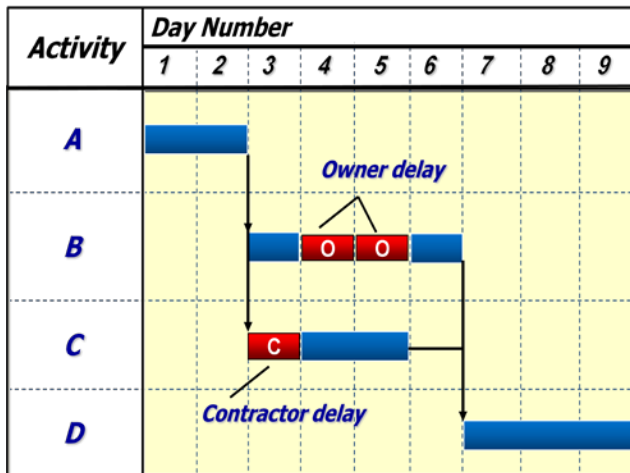


Figure 2.9: Bar charts for a small example of a daily windows analysis

Using the daily windows process in this example yields nine windows, which are analyzed as follows:

Days 1 and 2: The project did not experience any delays, so the project duration remains seven days.

Day 3: As shown in Figure 2.10, the critical path A-C-D exhibits a one-day contractor delay (c), which extends the project duration to eight days. Therefore, this window is one day longer than the previous window, indicating a project delay of one day. An examination of the critical path A-C-D reveals that this one-day project delay was caused by the contractor's (c) event. Accordingly, a contractor delay (C) is accumulated.

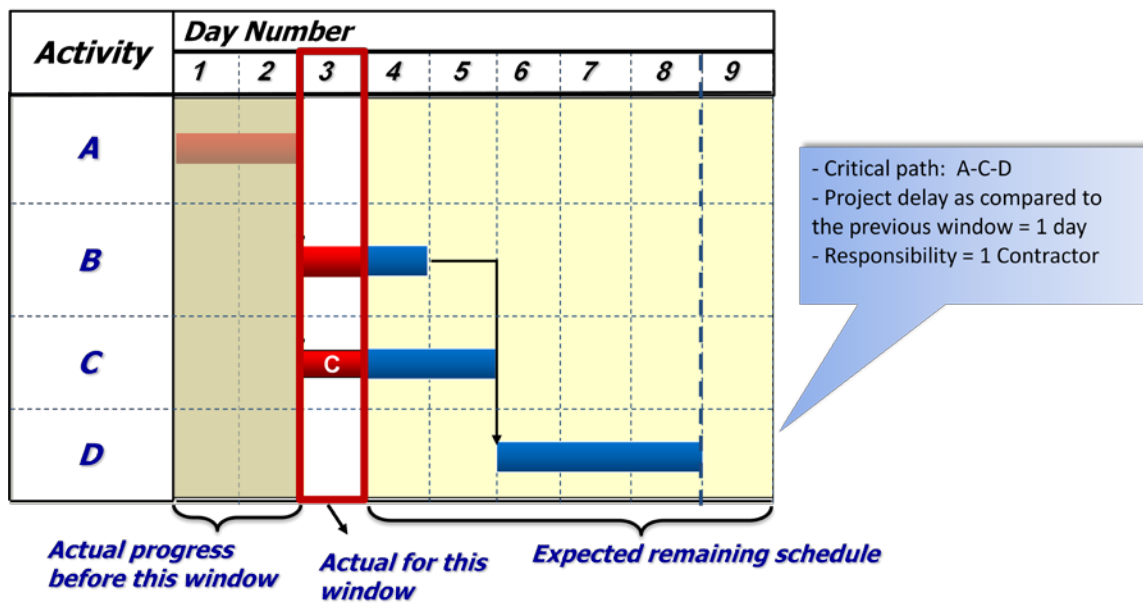


Figure 2.10: Daily windows analysis showing the window for day 3

Day 4: As shown in Figure 2.11, the window for the fourth day shows a one-day owner delay on the path A-B-D, but the project duration remains eight days, as in the previous window.

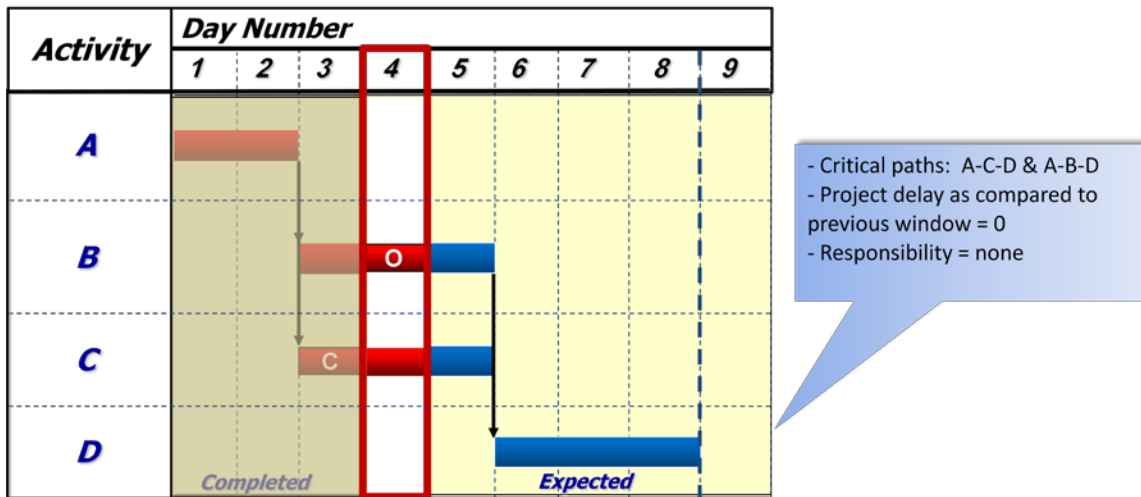


Figure 2.11: Daily windows analysis showing the window for day 4

Day 5: The project experiences a one-day delay due to the owner's delay on the critical path A-B-D, leading to the project duration becoming nine days (Figure 2.12).

Days 6 to 9: No additional delays occur, so the project duration remains at nine days.

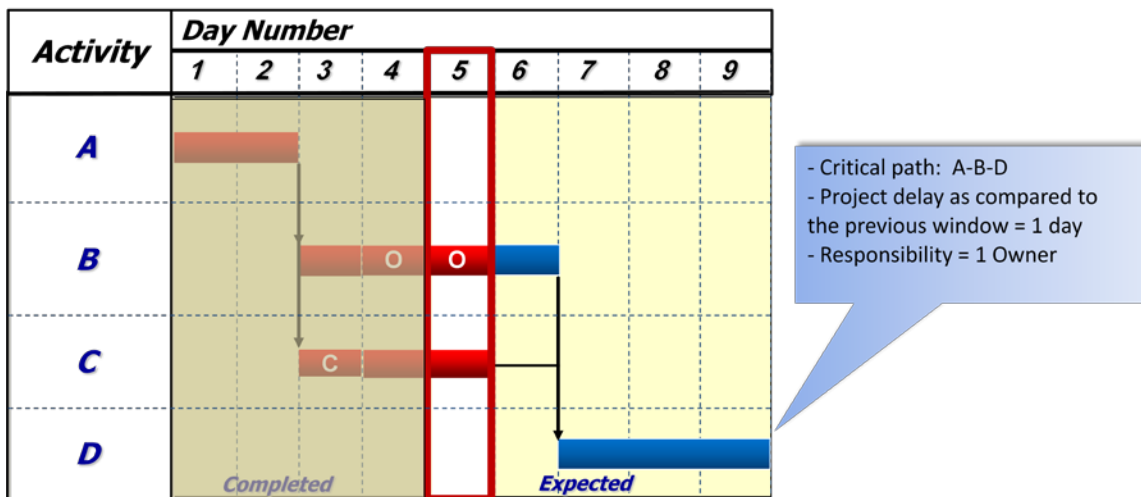


Figure 2.12: Daily windows analysis showing the window for day 5

The conclusions of the daily windows analysis are therefore a one-day contractor delay (1 C) and a one-day owner delay (1 O).

As demonstrated by this simple example, the daily windows analysis considers every change in the critical path(s). Some of these changes would be overlooked if traditional windows analysis were used to analyze the same case. However, daily windows analysis does not take into consideration other factors, such as multiple baselines and resource over-allocation. Later research by Hegazy and Menesi (2008b) introduced further improvements to the daily windows analysis. The resulting modified daily windows analysis considers multiple baseline updates due to changes in the durations of the activities and in the logical relationships among them, as well as the impact of resource over-allocation. This modified daily windows analysis is used in the developed model for schedule analysis.

2.5 Drawbacks of the CPM and Existing Software

Although owners and managers of contracting companies see the value in using CPM, field supervisors and subcontractors cannot use it effectively simply because CPM schedules do not reflect reality (Kuhn 2006). A number of researchers and practitioners have studied CPM and reported both benefits and criticisms. The following list includes the most important critical views of CPM and the pitfalls inherent in commercial software:

- Floats, the critical path, and the project status can be inaccurate due to the extensive use of leads and lags (Wickwire and Ockman 2000).
- Start-to-Start (SS) or Finish-to-Finish (FF) relationships have time dependence but not work-amount dependence (Lowsley and Linnett 2006).

- Date constraints ignore the network logic and the duration of the activity (Wickwire and Ockman 2000).
- Unrealistic activity durations can result from incorrect calculations of the remaining durations (Street 2000).
- The critical path may not always require the most attention (Street 2000).
- In both Primavera P3 and MS Project software systems, resource-constrained schedules produce inaccurate total float values under resource calendar constraints (Kim and de la Garza 2003).
- Logic abuses can result in confusion, delayed projects, and lawsuits (Korman and Daniels 2003).
- Multiple calendars diminish the ability to understand and analyze the critical path and total floats (Scavino 2003, O'Brien and Plotnick 2006, Kim and de la Garza 2005b).
- CPM schedules can be difficult to analyze due to out-of-sequence progress (Herold 2004).
- Primavera software can produce inaccurate dates when resource calendars that include nonworking days are used (Kim and de la Garza 2005b).
- Negative lags and Start-to-Finish (SF) relations with different calendars need to be avoided (Kim and de la Garza 2005b).
- Schedule results lack transparency for projects with a large number of activities (Sanders 2005).
- Delay analysis produces different results under different window sizes (Hegazy and Zhang 2005).

- Schedule analysis is not a straightforward task under multiple baseline updates and under resource allocation (Hegazy and Menesi 2008a).
- Networks with multiple relationships (FF and SS) are complex to analyze (Lu and Lam 2008).
- Schedule analysis is difficult, particularly when the contractor changes the logical relations to show fewer delays and does not notify the owner (Hegazy and Menesi 2008b, Livengood and Anderson 2006).
- CPM scheduling software that can constrain the finish of an activity also lacks the ability to determine where or when the activity may be interrupted (Winter 2003).
- CPM cannot quantify the effect that consuming the floats has on the project duration and cost (Sakka and El-Sayegh 2007).
- CPM analysis can be wrong if the level of detail used to prepare the analysis is inappropriate (Lowsley and Linnett 2006).
- When SS and FF relations are used, only a portion of an activity becomes critical. Available software is generally unable to portray this situation (Lowsley and Linnett 2006).
- Representing contractor and owner events on as-built schedules is difficult (Hegazy et al. 2005).
- Some of the ENR's Top 400 Contractors have commented about CPM (Kelleher 2004) as follows
 - "It does not always simulate actual conditions."
 - "PDM can be harder to follow and explain the logic because of the start-to-start, finish-to-finish logic relationships and the use of lag durations."

- "It is sometimes too cumbersome to convey exactly what we want to convey."
- "Changes in the field and deviations from the baseline often take a long time to be reflected on the schedule."
- "Cannot easily trace network logic graphically."
- "We need the detail to make the schedule easy to update and able to quantify impact properly and in a timely manner."

2.6 Recent Efforts to Enhance CPM

Herold (2004) pointed out that the construction industry and software vendors need to develop enhanced PDM scheduling software that can not only perform complex calculations, but that also has the ability to present schedule information clearly and concisely. In an attempt to improve schedule representation, Herold's key to improving the CPM is to change complex relations into FS only. His approach converts activity relationships into additional activities whose durations are equal to the relationship lags. The resulting project network, however, is then more complex and difficult to understand. An example is shown in Figure 2.13.

Plotnick (2006) focused on a better understanding of the relationships between activities. He discussed the confusion with respect to whether the lag duration is measuring the passage of time or actual progress. Most existing commercial software takes the approach of measuring the number of days from the reported start, regardless of actual progress. He then introduced a new system called the Relationship Diagramming Method (RDM), a variant of CPM, which records additional information about the relationships, such as the purpose of the relationship. He also introduced additional relationship types, such as Begin-to-Start,

Progressed-to-Start, Remaining-to-Start, End-to-Finish, Finish-to-Remainder, and Finish-to-Progressed. However, adding more types of relationships among the activities further complicates the project network.

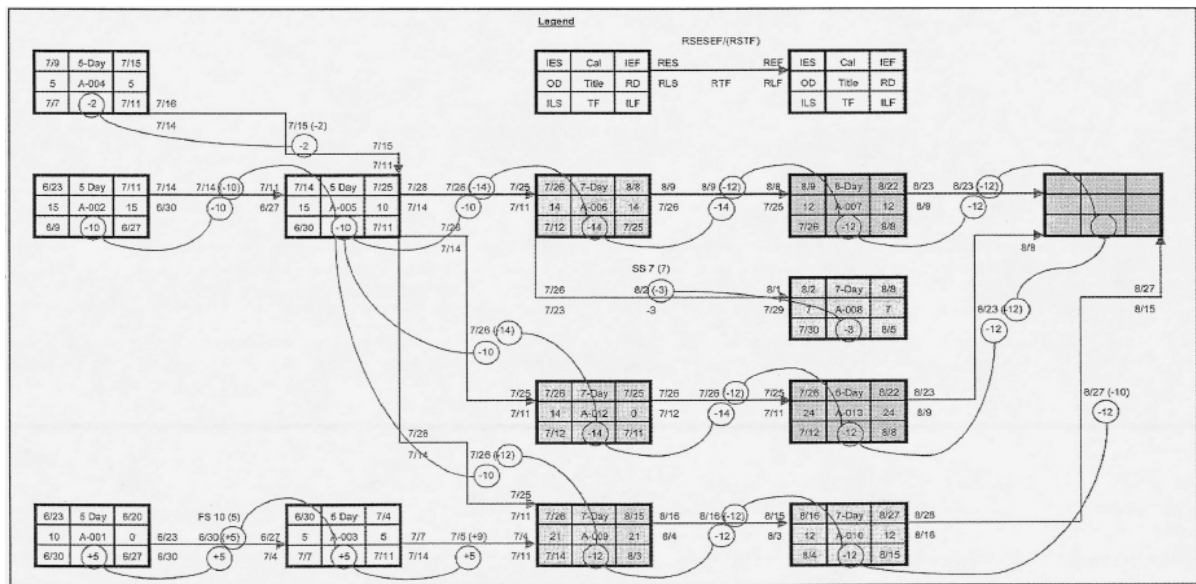


Figure 2.13: Example of a project network based on Herold's (2004) approach

Basu (2008) investigated how CPM scheduling software handles business rules such as resource allocation, cost tracking, or claim management. These rules, which have been embedded in scheduling software, are poorly documented, are poorly verified, and sometimes conflicting. Basu stressed the need to establish precise rules and a mathematical basis for schedule development and use. To that end, he suggested that software functionalities be validated so that the CPM calculations yield repeatable and consistent results, and he concluded that validation should cover not only the baseline schedule but also the various states of progress and the percentage of completion.

Many other studies reported in the literature have investigated ways to improve CPM scheduling and avoid common mistakes. In an effort to improve planning and avoid the problems created by complex relationships, Ponce de Leon (2008) presented a Logic Diagramming Method (LDM) that uses an activity notation that resembles arrow diagramming, albeit on a time scale. Activity relationships such as SS, FF, and SF, are permitted by inserting embedded nodes on, or between, the activity start and finish nodes. In LDM notation, relationships are viewed as connecting two nodes, an embedded node and a node, or two embedded nodes, as shown in Figure 2.14.

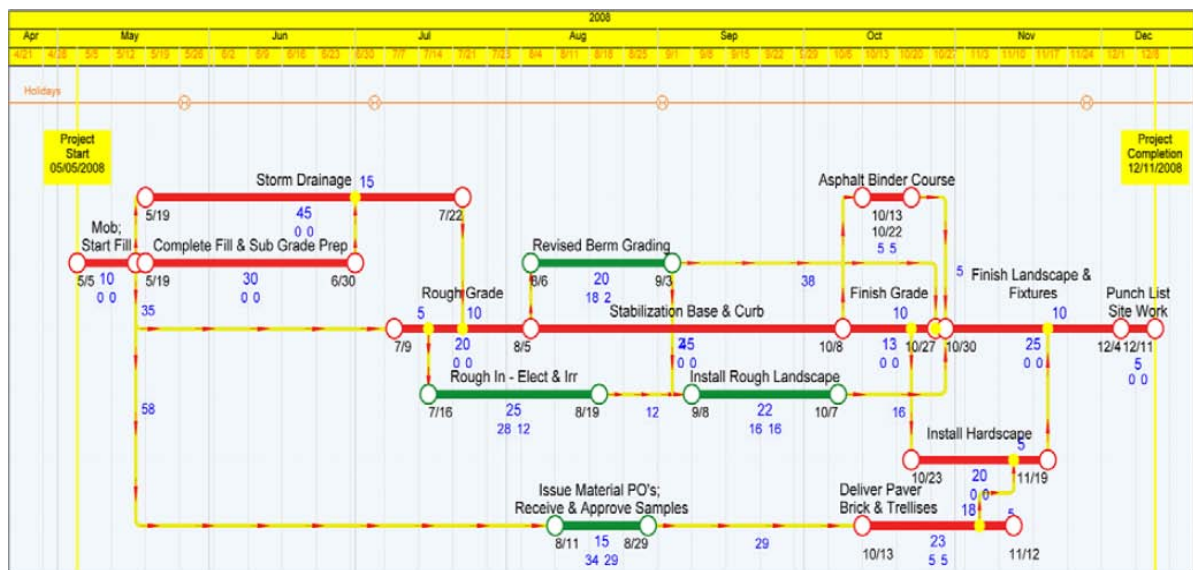


Figure 2.14: Sample schedule using LDM notation (Ponce de Leon 2008)

In another effort to avoid the difficulties associated with complex relationships in project networks, Lu and Lam (2009) proved through a PDM network example containing non-FS relationships that non-FS relationships complicate total float determination and interpretation. They then proposed generic transform schemes in order to transform non-FS

relationships in a project network into equivalent FS relationships. They also indicated that transforming non-FS relationships to FS relationships with zero lag provides a better understanding of the scheduling results and paves the way for conducting further sophisticated scheduling analysis.

2.7 Conclusions

This chapter has presented a review of the research that reveals the shortcomings of CPM with respect to satisfying the changing requirements of the construction industry. Several surveys have confirmed that CPM schedules are used for two primary purposes: project management and claim analysis. In project management, the main purpose of the schedule is to help prioritize daily activities so that the project is completed on time, within budget, and to the approved level of quality. Although CPM provides important information, such as total floats and the critical path, CPM forward pass and backward pass calculations incorporate neither resource limits nor a deadline for constraining the duration of the project. These drawbacks make CPM schedules neither responsive to the on-the-spot nature of daily situations that require schedule changes, nor reflective of how project managers react to challenges encountered during the course of a project.

In construction claims, CPM schedules are analyzed in order to allocate their responsibility to the appropriate parties for delays and accelerations. In the literature, many researchers and practitioners indicate that CPM schedules have become difficult to analyze for reasons such as the use of complex relationships among activities, the use of lead and lag times in relationships, and the inadequate representation of site events in CPM scheduling software.

Chapter 3

Critical Path Segments (CPS) Scheduling Technique

3.1 Introduction

Based on the literature review represented in Chapter 2, four areas in which the current CPM algorithm needs enhancement were identified (Menesi and Hegazy 2008):

1. Improving network representation to avoid complexity
2. Improving representation of activity progress
3. Incorporating constraints into CPM
4. Enhancing project control and schedule analysis

These areas correspond to the drawbacks highlighted in Chapter 2. It is important to note that resolving the problems in these areas requires careful consideration of all the areas combined and necessitates a departure from traditional critical path analysis, which is rigid with respect to the representation of mid-activity events. The following sections provide a description of these areas of enhancement, followed by the presentation of a new critical path model that can address these needed enhancements in a collective manner. How the model addresses the first two enhancements is explained in this chapter, while the details about addressing the third and fourth enhancements are described in succeeding chapters.

3.2 Areas Requiring Enhancement

3.2.1 Improving Network Representation to Avoid Complexity

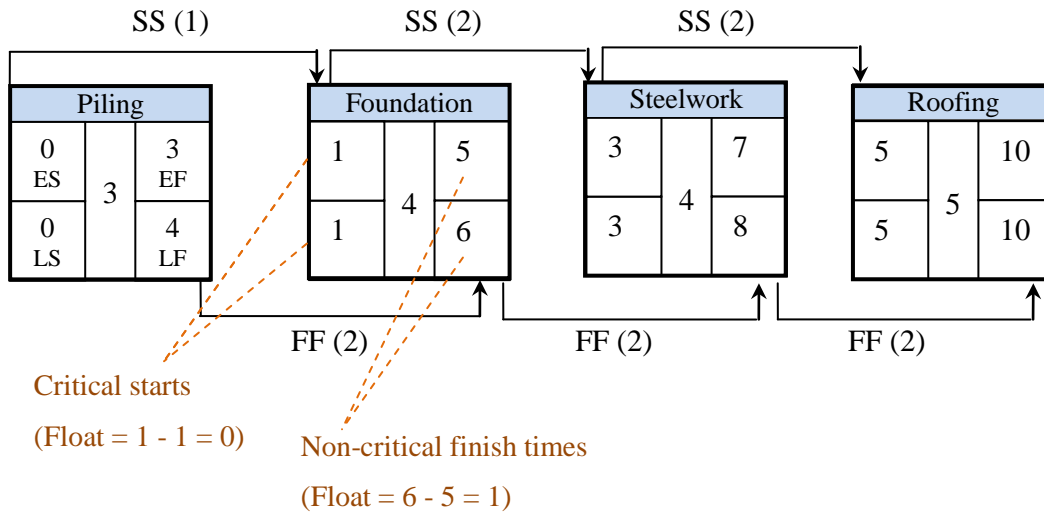
Often, construction projects involve situations that mandate the use of non-traditional relationships and lags to represent important interrelationships among the activities. For

example, a scheduler might need to indicate that mechanical work can start five days after the start of the electrical work. In this case, the scheduler needs to use a start-to-start (SS) relationship with a lag of five days between the electrical and the mechanical activities.

As explained in the literature, however, complex relationships such as finish-to-finish (FF), start-to-start (SS), and start-to-finish (SF) complicate the CPM network. More importantly, the use of such relationships can lead to situations in which the start dates of some critical activities might be critical but their finish dates are not (Lowsley and Linnett 2006, Moder et al. 1983). The CPM algorithm and existing software systems are generally unable to portray these activities as partly critical, mainly because of the assumption that each activity is a single undivided bar with a given duration.

Figure 3.1 illustrates a simple case study similar to the one reported in Lowsley and Linnett (2006). The figure shows a network in which each activity is linked by both an SS and an FF relationship. The network calculations in this case, as shown on the figure, reveal that the start dates are critical for all activities; however, because of the overlap created by the SS and FF relationships, the finish dates for the first three activities contain a float. Such a situation is complex to analyze in conjunction with the incorporation of practical aspects such as resource allocation, corrective actions, and schedule crashing, not to mention progress evaluation and delay analysis. Thus, a new representation is needed that can reduce network complexity yet enable the planner to specify practical relations and at the same time correctly define the critical path.

(a) How complex relationships create partially critical activities



(b) How existing software indicates all activities as critical, in contrast to the calculations in part a

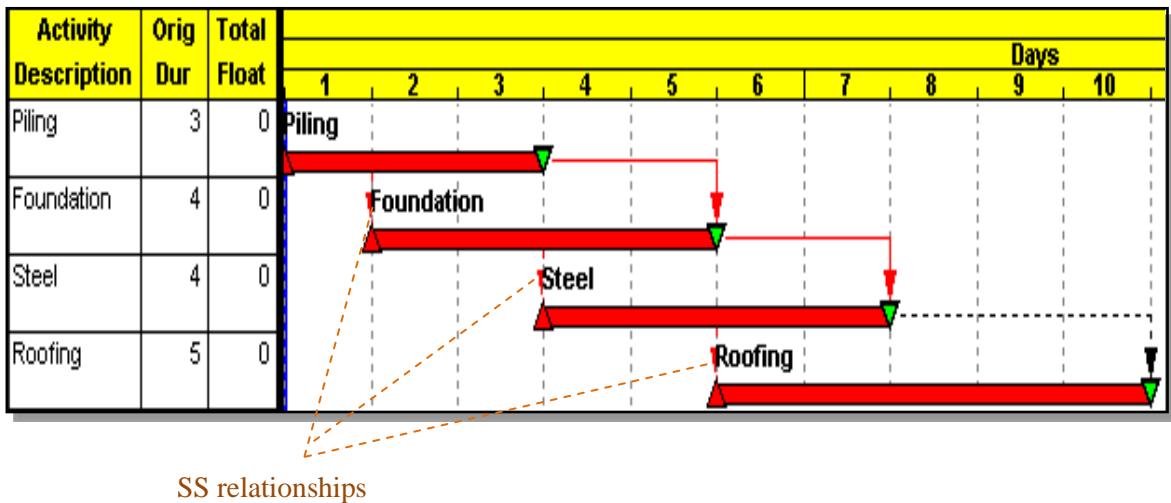


Figure 3.1: Case study showing that complex relationships can create partially critical activities

As shown in the bar chart in Figure 3.1, representing each activity as a single bar (as in all software systems) does not provide a clear picture of the critical path, which is essential for delay analysis and project updates.

3.2.2 Improving Representation of Activity Progress

The representation of activities and their durations are the basis of schedule calculations. An important consideration is therefore the effect of actual progress and site events on the representation of the activity and on the calculation of the remaining duration. When the schedule is updated with the actual progress, differing assumptions can be made when the remaining duration of activities is estimated (Figure 3.2). While some schedulers may assume that the rate of progress experienced to date will continue for the remaining work, others would estimate the remaining duration based on the original planned duration. Equations for calculating the remaining duration in either cases (or a weighted average of both) are presented in Hegazy and Petzold (2003). These assumptions may lead to disagreements among the project parties about the project completion date, and about the manner in which delay analysis results are produced.

The representation illustrated in Figure 3.2 shows a daily percentage of the activity and thus indicates clearly which calculation method is used for the remaining duration, as well as the slow and speedy progress dates. In contrast to this representation, existing software systems represent activities as solid bars that span the entire duration, without any indication of progress amounts or the method used to calculate the remaining duration. Because the software representation of progress data is not well suited for legible schedule analysis,

Hegazy et al. (2005) developed a bar chart in which the bar for each activity is composed of spreadsheet cells, with each cell representing one day or one week, or any other unit of time. The activities are thus represented not in solid bars, as in commercial software, but as a group of adjacent cells that make up the duration of the activity. Such a representation can better represent site events associated with the different parties.

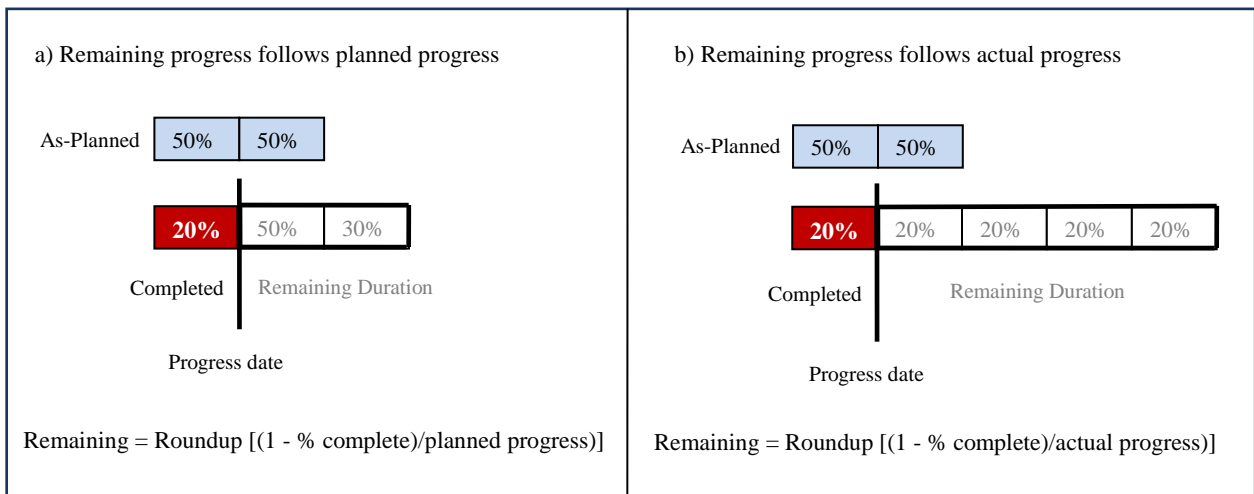


Figure 3.2: Two methods for calculating the remaining duration of an activity that are not clearly indicated in existing tools

3.2.3 Incorporating Constraints into the CPM

In addition to the effect of complex relationships, the critical path and float calculations are significantly affected by situations that involve resource limits and multiple calendars (Bowers 1995, Fondahl 1991, Kim and de la Garza 2003, Lu and Li 2003). For resource-constrained projects, the backward pass CPM calculation may produce incorrect total floats because the sequence of some activities relies not only on the logical relationships but also on resource dependencies (Kim and de la Garza 2003). One key problem with existing tools

is the fact that each activity is considered as a single bar, which does not represent the way contractors resolve resource constraints, particularly during actual progress. In existing software, for example, once an activity is started, no splits can be introduced in order to resolve resource over-allocation on a given day (Son and Mattila 2004).

Despite the advancements in CPM scheduling software during the last two decades, the resolution of resource over-allocation still needs to be included as part of the CPM formulation, not as an external improvement. Another consideration that should also be incorporated into the CPM calculation is the dynamic nature of the duration of the activities. Currently, durations are estimated and pre-fixed before the CPM calculation. CPM analysis should dynamically consider the duration of an activity as a function of the calendar date on which the activity is to start. For example, when the CPM forward pass starts an activity in a low-productivity season, the duration of the activity should be modified accordingly.

In addition to its limitations with respect to resource limits and dynamic activity duration, regular CPM analysis is not formulated to determine a schedule as a function of a given deadline. While the literature describes several techniques that have been developed in order to resolve these problems individually, i.e., time-cost trade-off analysis and limited-resource allocation, little effort has been devoted to considering them simultaneously, largely because of the inherent complexity of projects and the difficulties associated with modeling the combination of all aspects.

3.2.4 Enhancing Project Control and Schedule Analysis

Because it is rare that a construction project proceeds exactly as scheduled, construction progress needs to be continually monitored and documented, deviations must be identified, and optimum corrective actions suggested.

Daily site events are usually recorded in a variety of media, including daily site diaries, notes from progress meetings, daily weather records, photographs, and weekly progress reports. Although the daily site report is an important document for following the progress of an activity, it is often given the least attention (Pogorilich 1992). Few researchers have been interested in developing computerized systems for daily site reporting (e.g., Scott 1990, Hegazy et al. 2005).

Midway through the execution of a project, the parties may agree on a schedule update for reasons such as the contractor's corrective action to recover delays, e.g., acceleration and logic changes, responses to owner-requested changes; and responses to changed resource loads. The update becomes a new baseline for measuring progress. In such a case, the earlier portion of the project is measured against the first baseline, while the portion that occurs after the update is measured against the new baseline. Therefore, a better representation of the baseline data/decisions needs to be formulated. In addition, a systematic procedure for schedule analysis is needed in order to account for varying baselines, particularly when baseline updates involve changes to the duration of an activity and to logical relationships. Among recent developments for improving schedule analysis and project control is the effort

by Hegazy and Menesi (2008b) to develop a procedure for considering multiple baseline updates.

3.3 Critical Path Segments (CPS)

The following subsections describe the new Critical Path Segments CPS representation of a project network and activity progress, which addresses the areas of needed enhancement in a practical manner and within a unified framework.

3.3.1 New Network Representation

Since the representation of activities and their durations are the basis for schedule calculations, improving the representation of the activities would solve many of the problems mentioned above. As opposed to the traditional way of representing the activities as solid bars that span a given duration, CPS represents each activity as a number of separate, but connected time segments that add up to the total duration of the activity. For example, an activity with a duration of three days is represented by three time segments. This method permits the representation of any logical relationship (Start-to-Start, Finish-to-Finish, or Start-to-Finish) using only a Finish-to-Start (FS) relationship (Figure 3.3). As shown in the figure, SS and FF relations are easily converted to FS relations. In addition, lag times, which are the source of many calculation problems in traditional CPM, are not needed.

It should be noted that this modified CPM analysis uses physically separated activity portions not just as a matter of representation but that it also then applies schedule calculations, not at the whole activity level but at the individual segment level.

In essence, CPS uses a day-by-day rather than activity-by-activity CPM analysis. This method also facilitates the tracking of resources on a daily basis.

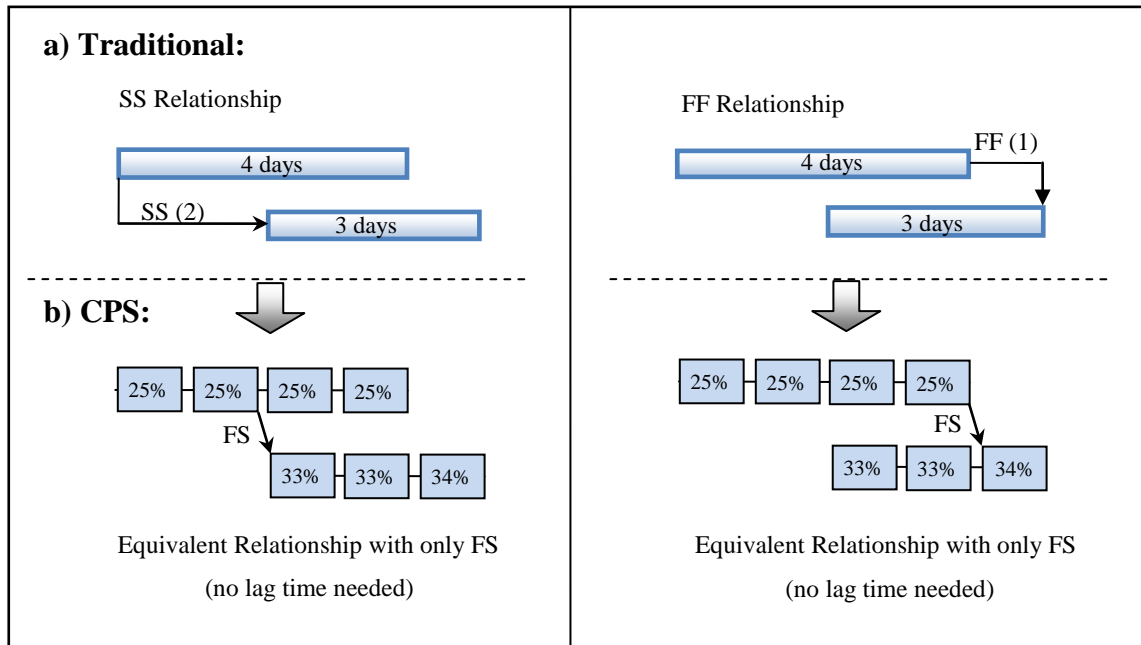


Figure 3.3: New representation of activities as separate time segments to avoid complex relationships and lag times

The improvements that CPS provides compared to CPM are due not only to the reduced complexity of the network but also to more accurate calculations. CPS can identify any partly critical activities and thus produce a more accurate critical path (Hegazy and Menesi 2010). As an example, the small CPM example in Figure 3.1 was recalculated using CPS, as shown in Figure 3.4. These two figures permit a comparison of the traditional representation of project activities shown in Figure 3.1 with the CPS representation illustrated in Figure 3.4. It can be seen that the CPS representation clearly displays the relationships between the

activities only as FS relationships without lags and that it also defines the critical parts of these activities, which current software tools cannot do.

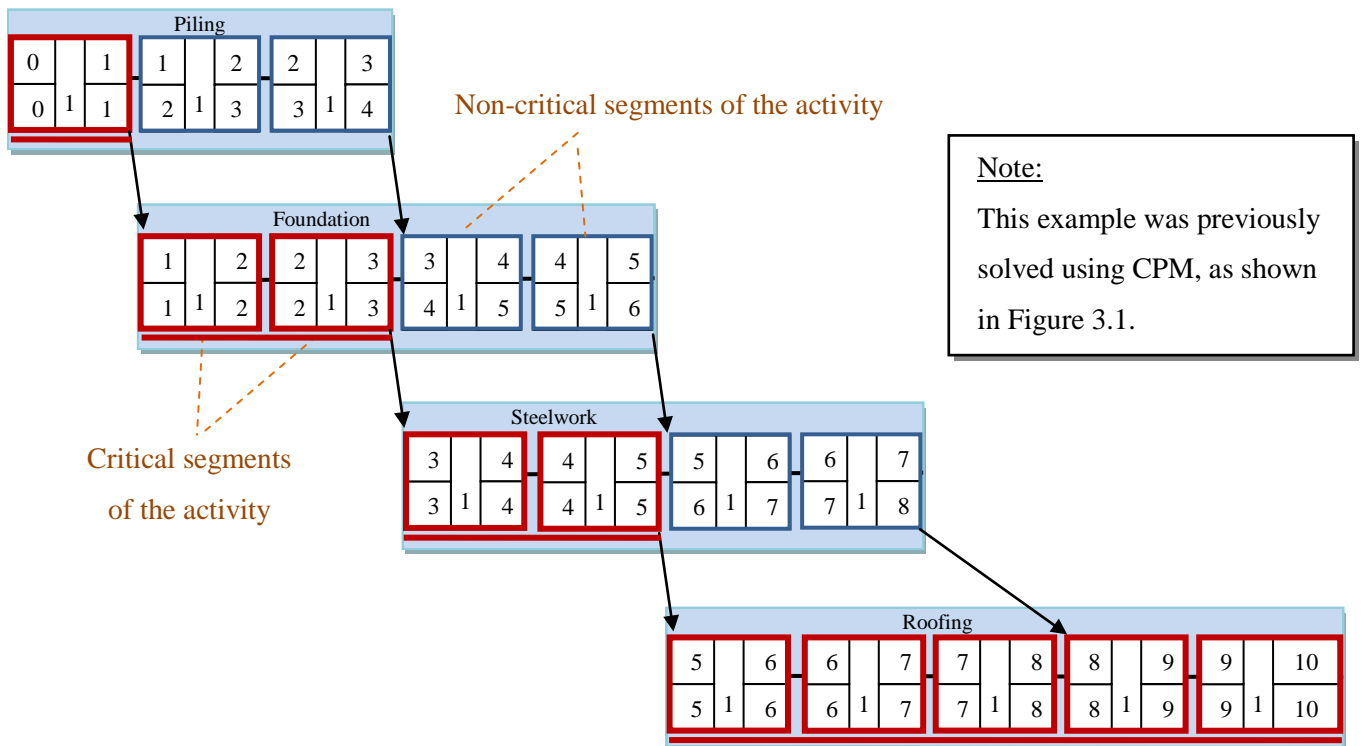


Figure 3.4: CPS representation accurately identifies the critical segments of the activities

An important additional benefit of the new network representation is the increased ability to represent the intent of the relationships between the activities. The CPS model can define the relationships between activities not only as time-based, but also as production-based. For instance, rather than indicating that steel reinforcement work can start two days after the formwork begins, CPS enables the project manager to specify that each 20% of the formwork completed is followed by 20% of the steel reinforcement. This kind of relationship is

illustrated in Figure 3.5. This representation of relationships thus not only communicates the reason for the relationship but also simplifies network calculations.

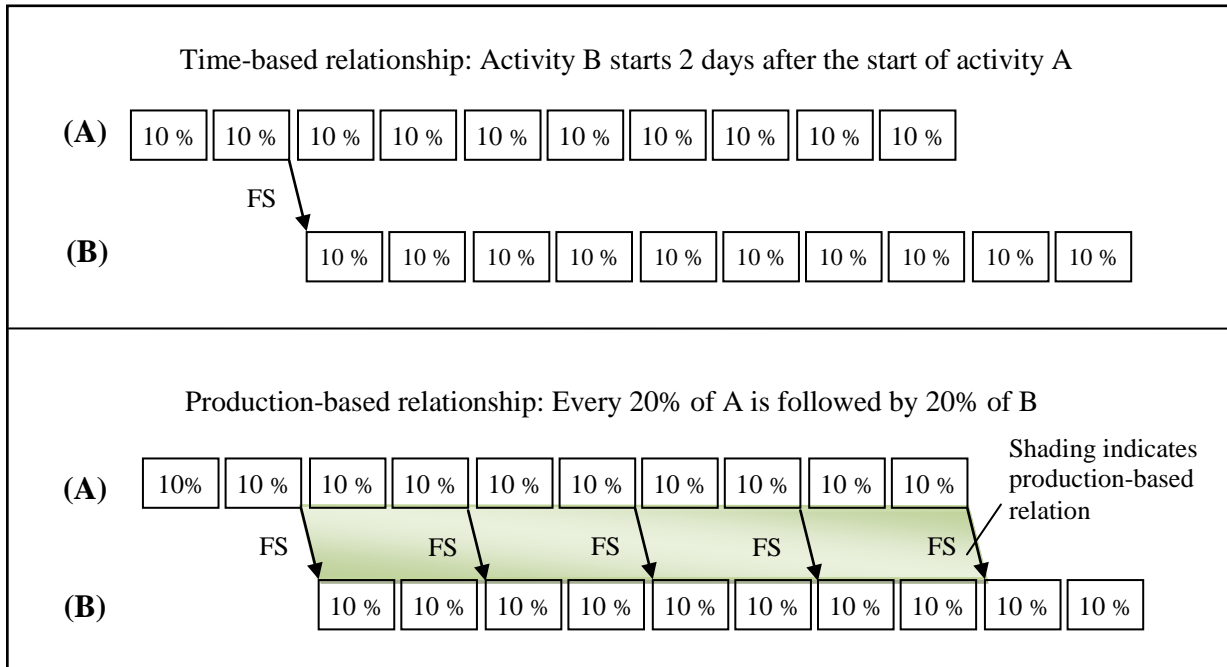


Figure 3.5: Time-based and production-based options in CPS representation

3.3.2 New Representation of Activity Progress

In CPS, progress is clearly represented so that schedule analysis can be carried out accurately with less disagreement among parties. With CPS representation, progress data is shown on the activity segments, and other events such as those involving the owner, the contractor, and the weather can be inserted as additional segments. Figure 3.6 shows the CPS representation of an activity with a five-day baseline duration. During the actual progress, the contractor used a one-day start delay to resolve a resource over-allocation. After the completion of 20%, the work was stopped for two days, with one due to owner interruption and one to bad

weather. The contractor thus had only one day in which to complete the activity as planned. Accordingly, the contractor decided to use a faster and more expensive method to accelerate the activity and finish the remaining work in two days, and each was then allocated 40% of the work.

Such a generic representation of the activities clearly shows the evolution of all activity events, including the effect of decisions such as acceleration and resource allocation. This representation is therefore general enough to permit the consideration of different relationships as well as the calculation of the float, the remaining duration, corrective actions, baselines, and schedule analysis.

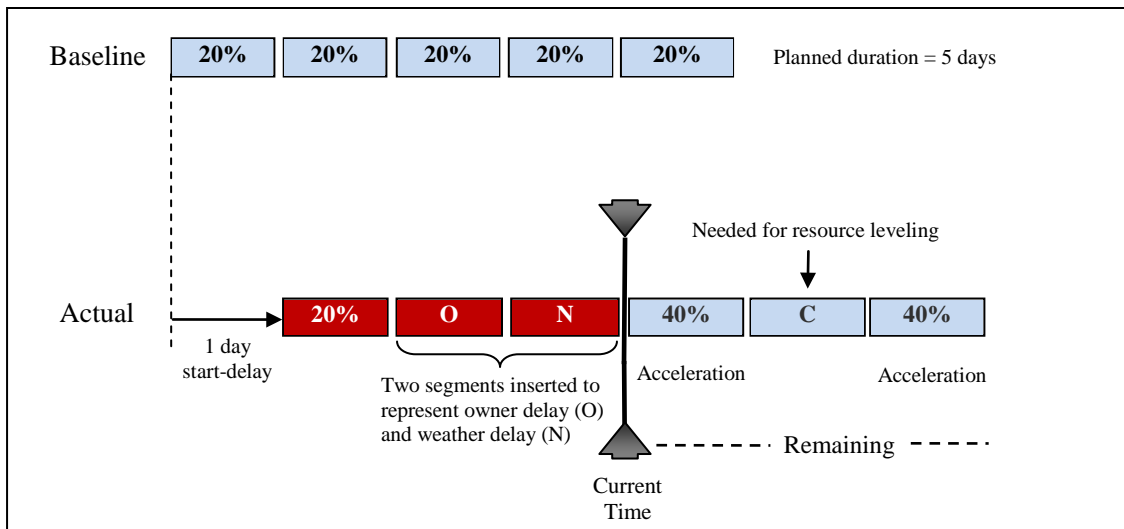


Figure 3.6: CPS representation of progress events

3.4 Proof of Concept

To demonstrate the ability of CPS to provide better representation and analysis than the traditional CPM, the following two simple case studies were used in order to show the ability of CPS to

1. Accurately define the critical path,
2. Avoid the errors caused by multiple calendars

Case 1: Ability to Accurately Define the Critical Path

Case 1 illustrates how the CPS can determine the critical path more accurately. Microsoft Office Project Professional 2007 (MS Project) has been used to develop the traditional schedule in this case (with continuous activities). As shown in Figure 3.7, the schedule produced by MS Project indicates two critical paths (red-colored activities): ABSDEF and AKLF. However, this information is neither accurate nor practical.

For comparison purposes, CPS representation has also been simulated on MS Project, with each time segment indicated by a separate MS Project activity, with a one-day duration, as shown in Figure 3.8. Although the software is not readily suited for CPS representation, Figure 3.8 clearly shows that for activities B and C only the first two days are critical, rather than the whole activity. Figure 3.8 also shows that CPS representation has enabled the work sequence to be modeled using only Finish-to-Start relationships with no need to use the Start-to-Start relationships and lags.

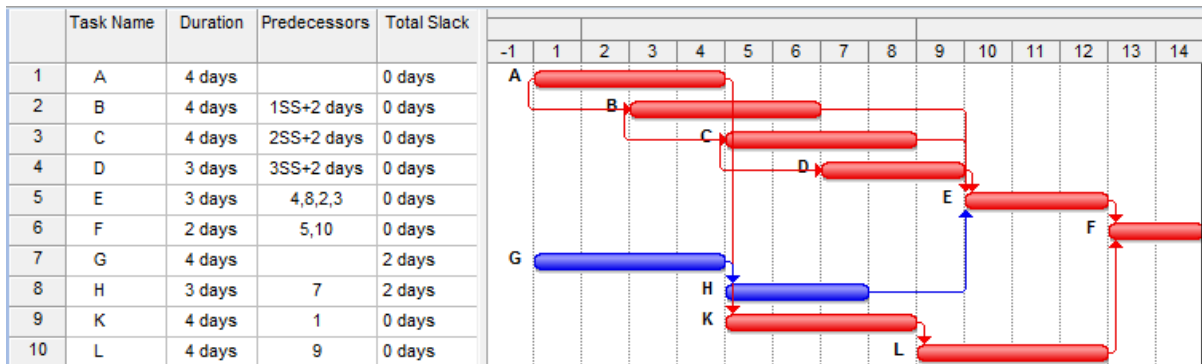


Figure 3.7: Traditional MS Project Schedule for Case 1

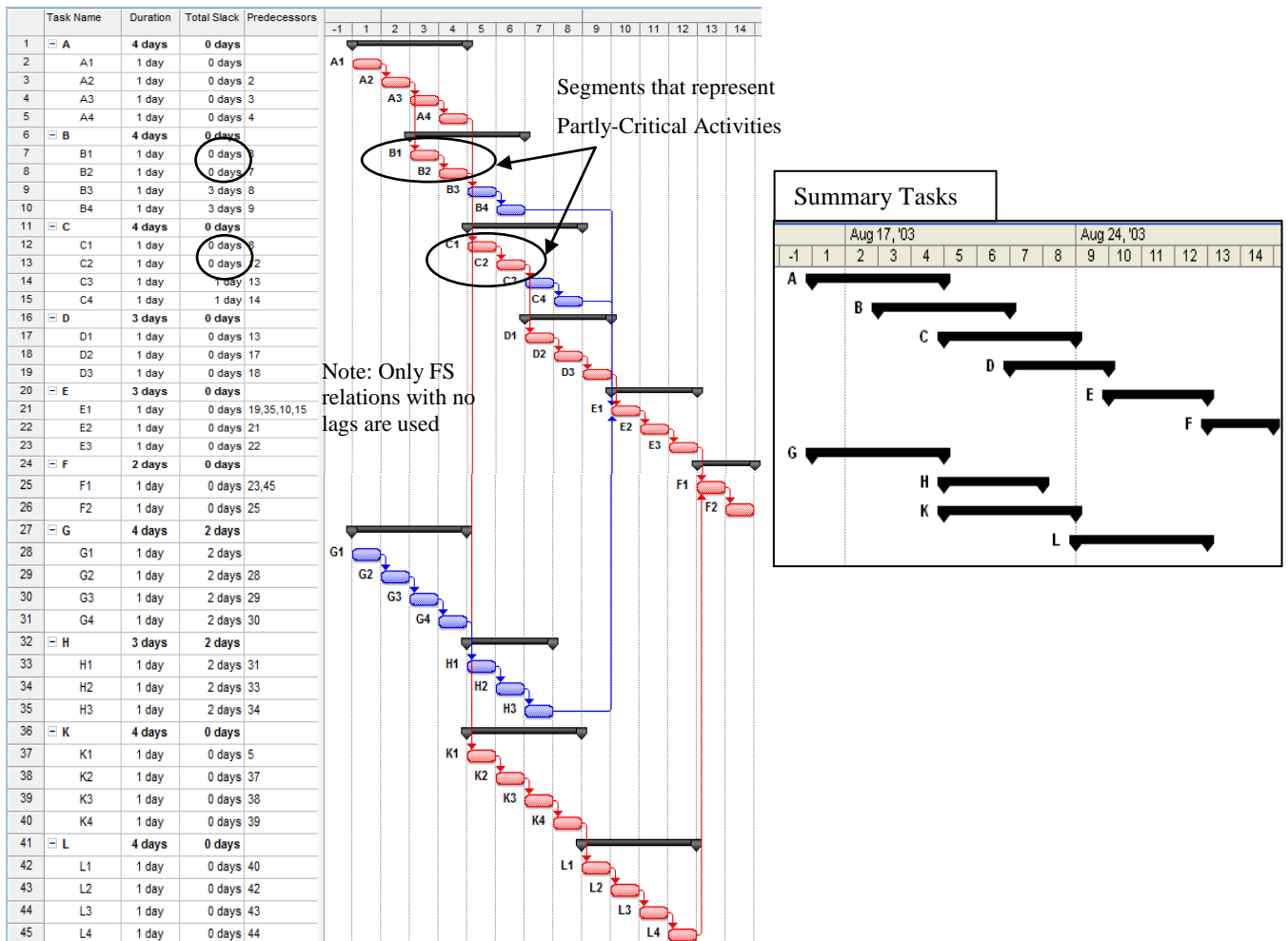


Figure 3.8: Simulated schedule with separate time segments

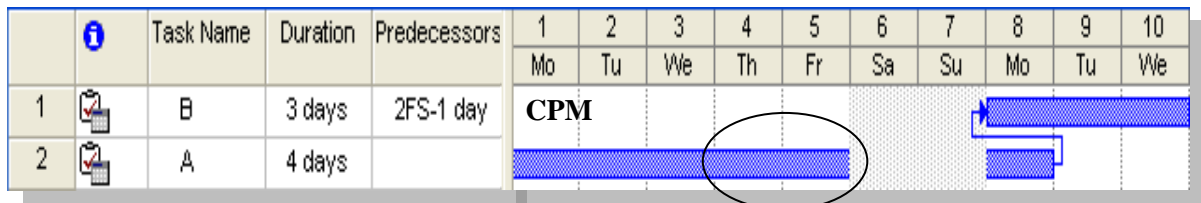
Case 2: Avoiding Errors Caused by Multiple Calendars

In Case 2, a small project involving the use of multiple calendars is examined. The example was used by Kim and de la Garza (2005b) to prove that when multiple calendars are used, Primavera P3 software generates incorrect dates for the activities. As shown in the top part of Figure 3.9, activities A and B have a finish-to-start (FS) relationship with a (-1) lag. Accordingly, the forward pass calculation in CPM determines that the early start time (EST) of B is day 8. However, since the FS relationship with a (-1) lag means that the successor can start whenever the remaining duration of the predecessor is 1 day, other options exist for the EST of activity B. Because of the difference in the calendars, the bottom part of Figure 3.9 shows that activity B can start on either day 4 or day 5. Day 4 is therefore the EST of Activity B, not day 8, which is not detectable using CPM calculations and existing software systems.

Microsoft Project software determines day 8 as the EST of activity B (Figure 3.10a), without taking advantage of the other possible EST times for activity B. Primavera P3 produces the same inaccurate results. Kim and de la Garza (2005b) therefore recommended that negative lags not be used with multiple calendars. It should be noted that because existing software shows only one calendar on the bar chart, Figure 3.10 (a) for example, wrongly shows that Activity A extends over the nonworking days of Calendar 2 (Th. and Fr.), which does not provide a correct indication of the activity duration.

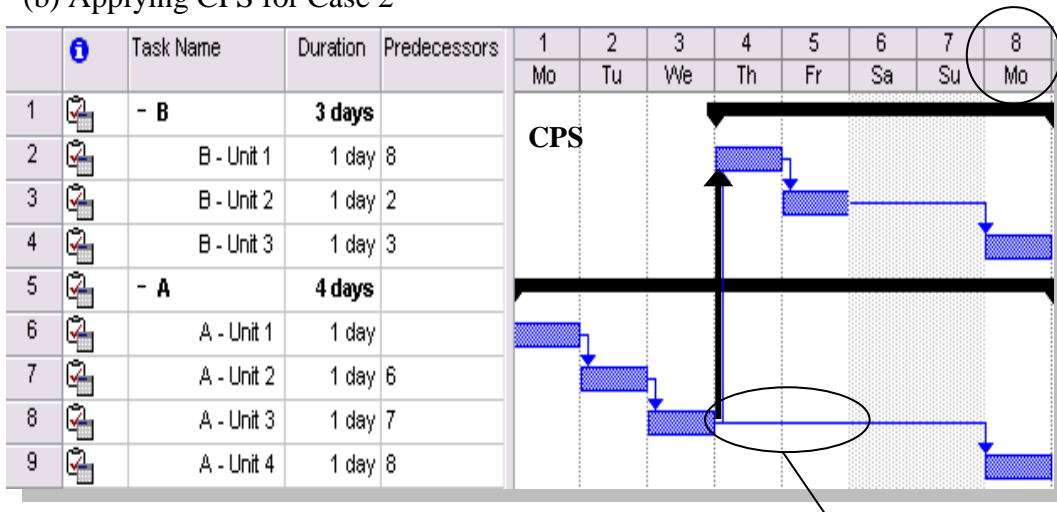
work will be performed for activity A on its nonworking days. More importantly, it illustrates how the CPS is capable of indicating possible earlier times that would result in the project being completed in 8 days, rather than 10.

(a) Microsoft Project output for Case 2



Nonworking days in Calendar 2 shown incorrectly by the CPM software

(b) Applying CPS for Case 2



CPS does not show incorrect duration

Figure 3.10: Simulating CPS for Case 2

3.5 Summary and Conclusions

This chapter has presented four areas of needed enhancements to traditional CPM analysis: (1) improving network representation to avoid complexity; (2) improving representation of activity progress; (3) incorporating constraints into CPM; and (4) enhancing project control and schedule analysis. To address these needed enhancements in a practical and collective manner, a new critical path analysis technique called critical path segments (CPS) is presented. The CPS representation of the project network and activity progress has also been described.

Two cases were used to demonstrate the benefits of using separate time segments in order to avoid complex network relationships, to accurately identify all critical path fluctuations, and to avoid multiple-calendar problems. The next chapters present an extended description of the CPS, including the full mathematical formulation for incorporating project constraints into the CPS formulation and for facilitating schedule analysis.

Chapter 4

Mathematical Formulation of the Critical Path Segments Method

4.1 Introduction

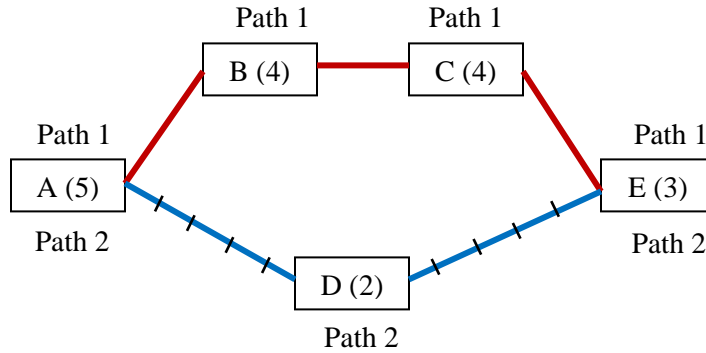
This chapter introduces the mathematical formulation underlying the CPS mechanism that makes it a generic tool for simplified planning, optimum schedule development, efficient project control, corrective action generation, and detailed delay analysis. The new Critical Path Segments (CPS) approach offers a finer level of granularity for a micro-level critical path analysis (Hegazy and Menesi 2010). It considers activity duration as a chain of separate time segments that correspond to a desired level of analytical detail. To save processing time, a new schedule calculation is introduced based on a forward pass only, which is then later applied to the CPS technique in order to eliminate the problems associated with backward pass and total float calculations.

This chapter first describes the new scheduling calculations based on a forward pass only, and then explains the details of the CPS algorithm.

4.2 New Schedule Calculation without a Backward Pass

Typically, forward pass calculations determine project duration and backward pass calculations determine the float times of activities. While scheduling calculations that do not include a backward pass substantially reduce the computational effort, a new approach for determining activity floats is then needed. To demonstrate the new calculation process, a simple case study was considered, the calculation process for which is shown in Figure 4.1.

a) Determining the paths



b) Path lengths and path floats

Path (1)	Path Duration (2)	Longest Path (3)	Path Float (4) = (3) - (2)
1. A-B-C-E	$5+4+4+3 = 16$	Max (2) = 16	$16-16 = 0$
2. A-D-E	$5+2+3 = 10$		$16-10 = 6$

c) Activity total floats

Activity	Paths	Path Total Float	Activity Total Float
A	1, 2	0 (path 1), 6 (path 2)	Min (0,6) = 0
B	1	0	0
C	1	0	0
D	2	6	6
E	1, 2	Path 1 = 0, path 2 = 6	Min (0,6) = 0

Figure 4.1: A simple case study illustrating the new approach for total float calculation

In Figure 4.1, all the paths in the project network are first identified (two paths in the presented case). In addition, each activity is marked by the path(s) in which it lies on. For example, activity A lies on both path 1 and path 2, as shown in Figure 4.1. The length of each path is then calculated as the sum of its activities' durations (column 2 in Figure 4.1b), which is a simple forward pass calculation. The project duration, which is also the critical path duration, then becomes the duration of the longest path (column 3 of Figure 4.1b). Non-critical paths will thus have floats that can be calculated as the difference between the duration of the longest path and the duration of each individual path (column 4 in Figure 4.1b).

Once the path floats are calculated, the total floats for the activities can be directly calculated as shown in Figure 4.1c. For each activity, the total float is calculated as the minimum of the path floats for all its paths. For example, activity A lies on both path 1 and path 2. Its total float is therefore the minimum of zero (float for path 1) and 6 (float for path 2): its total float is thus zero. In this process, no backward pass is needed, and activity total floats are basically calculated from the path floats.

Being based on forward pass processing only, this scheduling approach agrees with the findings of Woolf (2008) who indicated that the behavior of total float consumption is entirely path-centric. He concluded that activities acquire the criticality of the paths they inhabit because a change in the duration of any single activity changes the total float values for all activities on the path upon which the activity resides.

4.3 CPS Calculation Procedure

Figure 4.2 shows the flowchart for the CPS model. As shown in the figure, once the schedule data, including the activity data and project constraints, are defined, the CPS model formulates the activities and relationships in the new FS representation. The project duration and activity timings are then obtained by executing the CPS analysis, which considers both precedence relationships and resource limits. In this analysis, only a forward pass calculation is performed with the new method of calculating floats, as described in section 4.2. The resolution of any resource over-allocation is also incorporated into the forward pass, which substantially reduces computational time and eliminates errors in float calculations. The detailed steps of this CPS mechanism are described in the following subsections.

4.3.1 Step 1: User Input of Schedule Data

To avoid changing the conventional method schedulers are accustomed to using, the activity data do not differ from these of existing techniques. An example of the activity data is shown in Table 4.1. The data structure for each activity is as follows:

- Activity predecessors and relationship types (columns 2 and 3 of Table 4.1)
- Data for estimate 1:
 - Duration (days)
 - Cost (\$)
 - Resources used
 - Interruption (No = activity segments have hard links and cannot be separated or segments cannot have start-delay values, Yes = activity segments may be separated)
 - Interruption Cost (in case interruption of the activity is permitted)

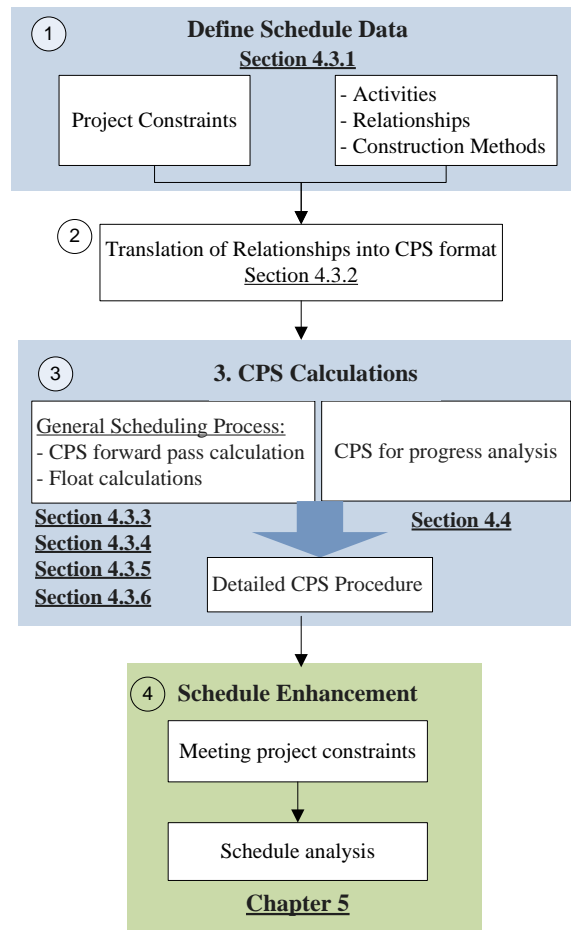


Figure 4.2: Flowchart for the CPS model

Table 4.1: Example of activity data in the proposed model

Activity	Activity relationships		Estimate 1					Estimate 2				
	Predecessors	Type of Relation	Duration	Cost	Resources	Interruptible	Interruption Cost	Duration	Cost	Resources	Interruptible	Interruption Cost
A	-	-	3	\$3,000	2R1+3R2	No	-	2	\$4,000	3R1+4R2	No	-
B	A (SS 2)	T*	7	\$7,000	2R3+2E1	Yes	\$300	5	\$8,000	Sub.	No	-
C	B (SS 50%)	P	3	\$3,000	Sub.	No	-					
D	B,C	T	5	\$5,000	2R1+2R2	Yes	\$1,500	4	\$6,000	3R1+3R2	Yes	\$1,500
E	-	T	4	\$4,000	2R1+2R2	Yes	\$700					
F	E (SS 20%)	P	9	\$9,000	Sub.	No	-					
G	F	T	7	\$7,000	Sub.	No	-	5	\$9,000	Sub.	No	-
H	G,D	T	3	\$3,000	2R1+2R2	No	-	2	\$4,000	3R1+3R2	No	-

* T = Time-Based relationship; P = Production-Based relationship

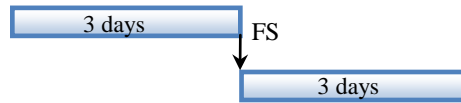
To later facilitate the resolution of deadline and resource constraints, the CPS model allows each activity to have associated optional cost estimates that represent the activity variables that offer a wide range of possible solutions to project constraints. It should be noted that the optional estimates represent practical options that vary from inexpensive and slow to fast and expensive. These estimates can represent different subcontractor quotes, crews with different skill levels, different equipment, or simply overtime work hours. Given the sequence of the activities and the various construction options, it is possible to arrive at a least-expensive plan that meets both deadline and resource limits.

4.3.2 Step 2: Translation of Relationships into the CPS Format

Based on the activity data, the new network representation is formulated as follows:

1. For each activity (i), first create the time segments so that the number of time segments equals the duration of the activity as indicated by the method index (M_i). Then, create a start milestone before the first time segment and a finish milestone after the last time segment. Connect the milestones and the time segments by FS relationships.
2. Read the relationships between the activities and create FS links between appropriate segments, as follows:
 - a. In the case of an FS relationship with a zero lag time, create an FS relationship between the finish milestone of the predecessor and the start milestone of the successor (Figure 4.3).

CPM:



CPS:

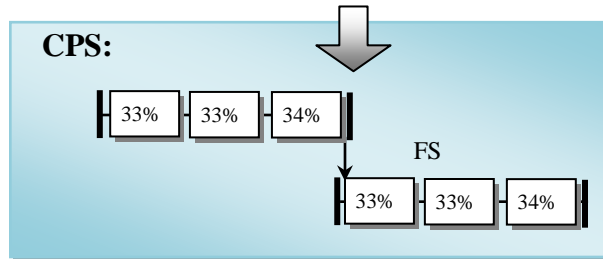
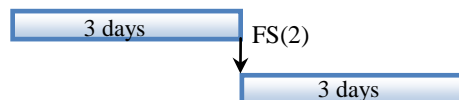


Figure 4.3: CPS representation of an FS relationship with a zero lag

- b. In the case of an FS relationship with a lag time, create start-delay time segments and link them to the first time segment of the successor, so that the number of these time segments equals the lag time. Then create an FS relationship between the finish milestone of the predecessor and the start milestone of the successor (Figure 4.4).

CPM:



CPS:

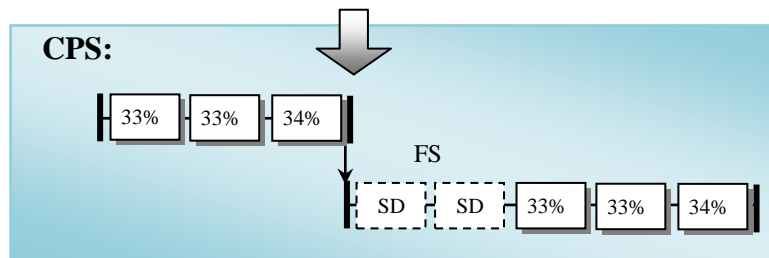


Figure 4.4: CPS representation of an FS relationship with a lag time

- c. In the case of an SS relationship with a zero lag time, create an FS relationship between the start milestone of the predecessor and the start milestone of the successor, as shown in Figure 4.5.

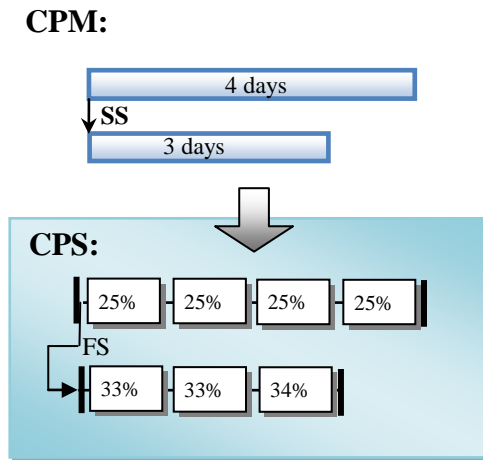


Figure 4.5: CPS representation of an SS relationship with a zero lag

- d. In the case of an SS relationship with a lag time (L), create an FS relationship between the time segment number (L) of the predecessor and the first time segment of the successor (Figure 4.6).

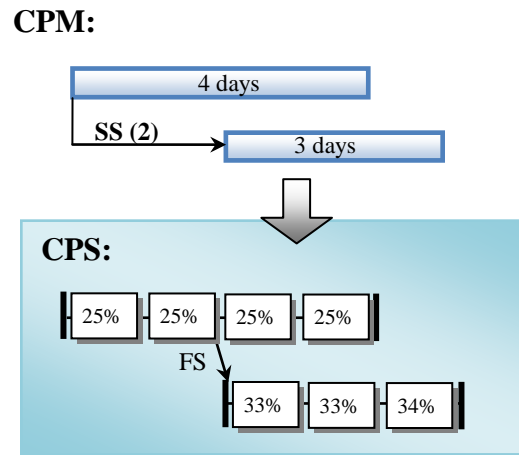
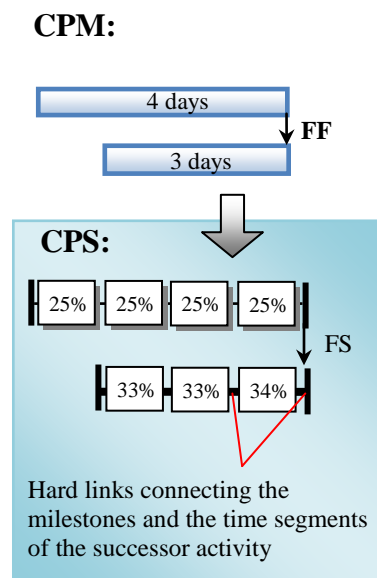


Figure 4.6: CPS representation of an SS relationship with a lag time

- e. In the case of an FF relationship with a zero lag time, create an FS relationship between the finish milestone of the predecessor and the finish milestone of the successor. In addition to the FS relationship, create hard links between the time segments of the successor. These hard links are indicated by the bolded lines in Figure 4.7. The purpose of these hard links is to maintain the logic during project execution and schedule updating, as explained later in Figure 4.10.
- f. In the case of an FF relationship with a lag time (L), create an FS relationship between the last time segment of the predecessor and the time segment number: $D_{suc} - L + 1$ of the successor, where D_{suc} is the duration of the successor or the total number of the successor's time segments. In addition to the FS relationship, create hard links between the time segments of the successor (Figure 4.7b).

(a) FF with zero lag time



(b) FF with a lag time

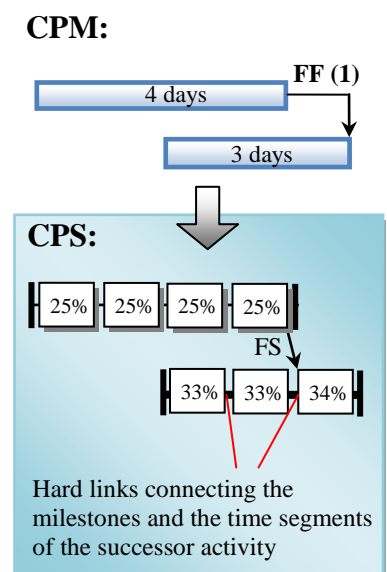


Figure 4.7: CPS representation of FF relationships

- g. In the case of an SF relationship with a zero lag time, create an FS relationship between the start milestone of the predecessor and the finish milestone of the successor. In addition to the FS relationship, create hard links between the time segments of the successor, as shown in Figure 4.8.

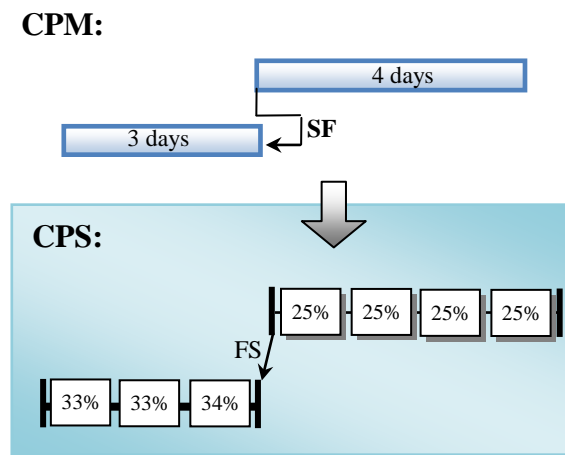


Figure 4.8: CPS representation of an SF relationship with a zero lag

- h. In the case of an SF relationship with a lag time (L), create an FS relationship between the start milestone of the predecessor and the time-segment number: $D_{suc} - L + 1$ of the successor, where D_{suc} is the total duration of the successor or the total number of the successor's time segments. In addition to the FS relationship, create hard links between the time segments of the successor, as shown in Figure 4.9.

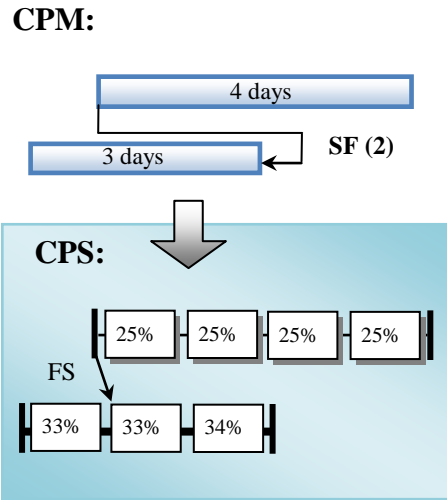


Figure 4.9: CPS representation of a SF relationship with a lag time

- i. In the case of using both SS and FF relationships (with lag times) to link two activities (Figure 4.11), either both relationships are converted to FS relationships as indicated above or they can be converted to a production-based relationship (depending on the purpose of the relationship or the intent of the user), as shown in Figure 4.11.

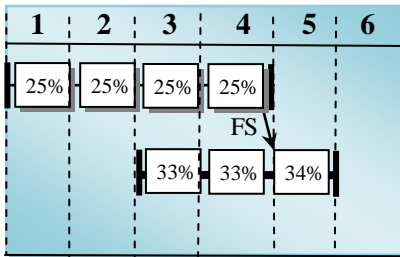
Using hard links between the activity segments in the cases of FF and SF relationships avoids the creation of unnecessary interruptions, as shown in Figure 4.10.

Case 1: FF with 1 day lag

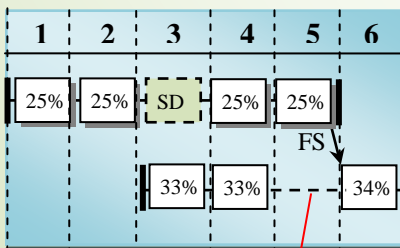
CPM:



CPS:

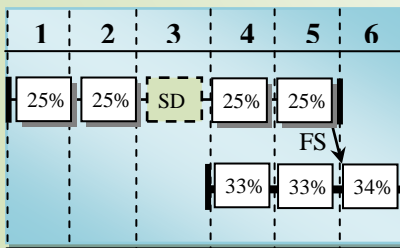


Without hard links:



Unnecessary interruption will be created if any segment of the predecessor is delayed for any reason

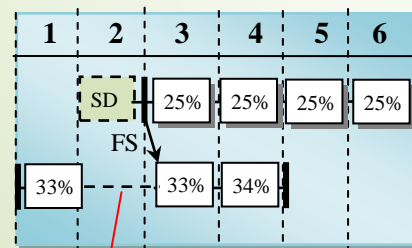
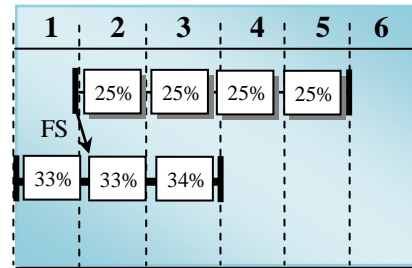
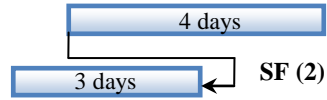
With hard links:



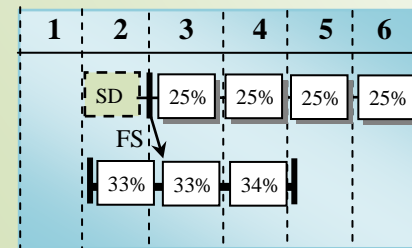
Unnecessary interruption is avoided due to the use of hard links between the time segments of the successor

Case 2: SF with 2-day lag

CPM:



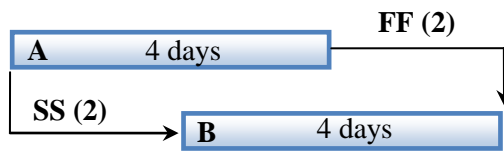
Unnecessary interruption will be created if the start of the predecessor is delayed for any reason



Unnecessary interruption is avoided due to the use of hard links between the time segments of the successor

Figure 4.10: The purpose of hard links when FF or SF relationships are converted

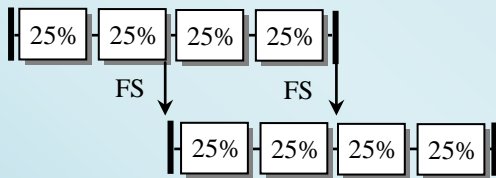
CPM:



Purpose/intent of the relationship

- B starts after 2 days of starting A
- B finishes after 2 days of finishing A

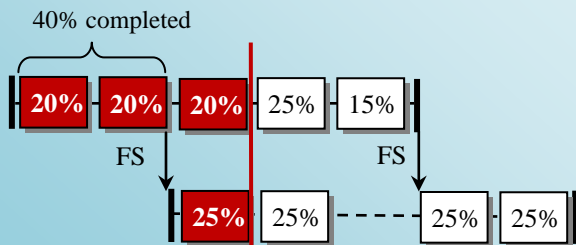
CPS



Time-based CPS



During Execution:

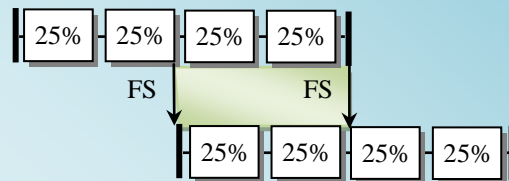


Progress Date

B can be started 2 days after A regardless of progress

Every 50% of A is followed by 50% of B

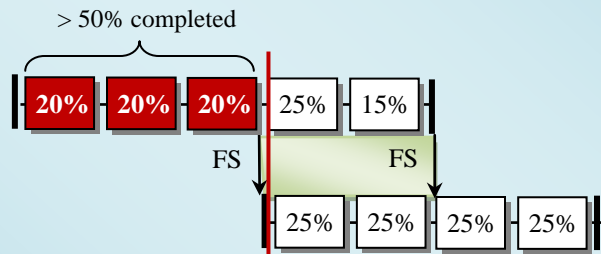
CPS



Production-based CPS



During Execution:



Progress Date

B can be started only after 50% of A has been completed

Figure 4.11: Converting SS and FF relationships to a production-based relationship

4.3.3 Step 3: CPS Forward Pass Scheduling Calculations

As mentioned earlier, CPS uses a forward pass computation; however, the process is applied not at the whole activity level but at the individual time-segment level. The CPS forward pass calculates the earliest start and finish dates for each time segment by working through the schedule from its start to its finish. At the same time, in the case of resource constraints, the CPS forward pass allocates the resources to each time segment, avoiding any resource over allocation that might occur due to resource constraints.

Several exact and heuristic methods have been proposed for solving the problem of resource constrained scheduling. The goal of exact methods, such as dynamic programming, zero-one programming, and implicit enumeration with branch and bound, is to find the optimal solution. However, these exact methods need a great deal of computational time, making them inappropriate for large and complex projects. On the other hand, heuristic methods, such as priority-based scheduling, can find a solution very quickly, which makes them very practical. Heuristic solutions may not be optimal but may be near optimal (Kastor and Sirakoulis 2009). Most project management software, such as Primavera and Microsoft Project, employ priority-based heuristics for resolving resource over-allocation (e.g., the resource leveling tool in Microsoft Project). Examples of these heuristics include giving higher priority to activities that have the earliest late start time (ELS), the earliest late finish time (ELF), the shortest total float (STF), and the greatest resource demand (GRD). Except for the GRD rule, these rules require that the CPM calculations for both forward and backward pass to be completed before the rule can be applied. The GRD, therefore, most suits the CPS mechanism, which involves only a forward pass calculation.

CPS allocates the resources on a day-by-day basis during the forward pass calculation, using the GRD rule to resolve any resource over-allocations. The GRD rule assigns priority on the basis of the total resource-unit requirement for all types of resources, with higher priorities being allocated to greater resource demands. The priority of an activity is calculated as presented by Davis and Patterson (1975):

$$\text{Priority} = d_i \sum_{k=1}^m r_{ki} \quad (4.1)$$

where d_i is the duration of activity i , r_{ki} is the per-period requirement for resource type k by activity i and m is the number of different resource types. With this heuristic rule being the basis for resolving resource constraints, the CPS forward pass process defines three sets of time segments that are evaluated for each day in the process:

- Time segments that have been already allocated,
- Eligible time segments (their predecessors have been allocated),
- Non-eligible time segments (their predecessors are not yet allocated).

Accordingly, the forward pass calculation proceeds as follows:

1. Set schedule time ($T = 0$) at the beginning of the project;
2. Set the start time ($ST_{ij} = 0$) and the finish time ($FT_{ij} = 1$) for all time segments that have no predecessors;
3. Define the eligible time segments (their predecessors have been allocated);

4. Check the availability of resources for the eligible time segments:
 - a. Calculate the GRD rule for activity i associated with an eligible time segment j ;
 - b. Sort the eligible time segments according to the GRD rule values (Equation 4.1):
 - i. In the case of a tie that is, when two eligible time segments have the same GRD value, priority is given to the eligible time segment that is not the first time segment of an activity: priority is given to the activities that have already started;
 - ii. In the case of a tie, priority is given to the eligible time segment that requires the largest number of resources;
 - c. Assign each resource (R_{ijk}) of type K to the top-ranked eligible time segments in order to satisfy the resource needs R_{mi} specified for construction method M_i ;
 - d. Once resources are assigned to a time segment, move the time segment from the eligible set to the allocated set. In addition, calculate the start time (ST_{ij}) and finish time (FT_{ij}) for each allocated time segment j of each activity i as follows:

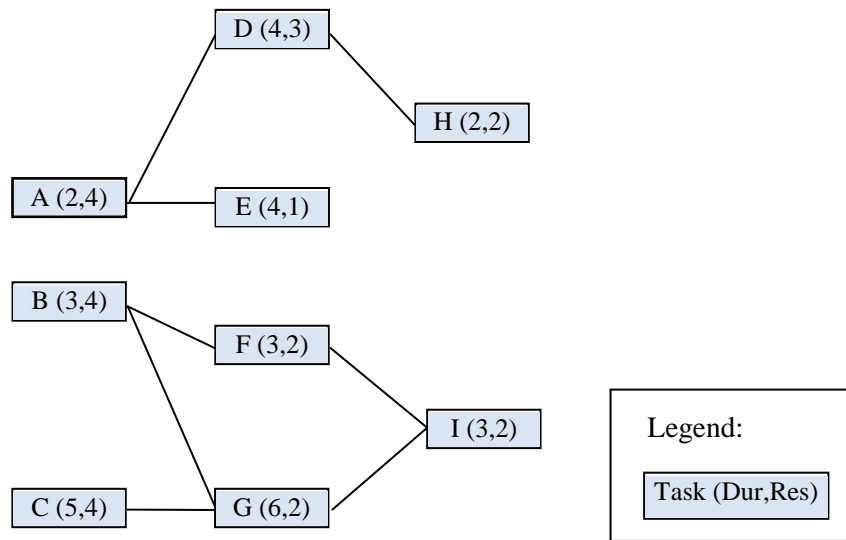
$$ST_{ij} = \text{Current schedule time } (T) \quad (4.2)$$

$$FT_{ij} = \text{Current schedule time } (T) + 1 \quad (4.3)$$

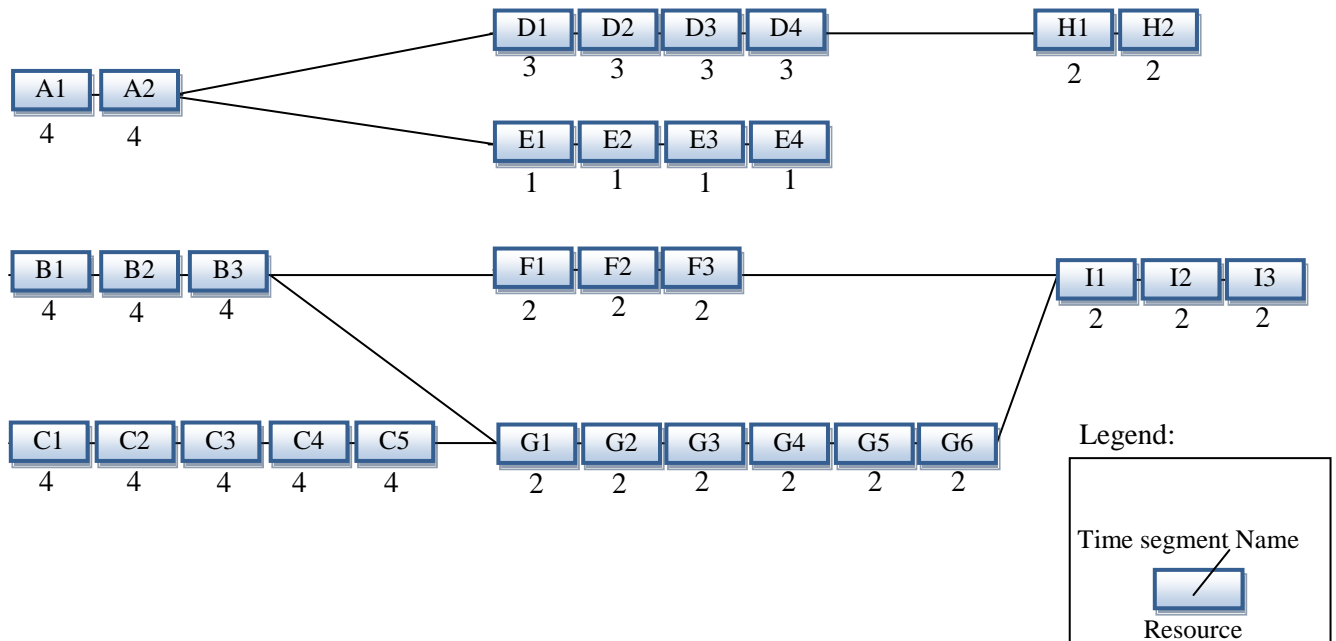
- e. Once all available resources are allocated, if some eligible time segments have not been allocated, keep these time segments in the eligible set and add a start delay (SD_{ij}) value of 1 to these eligible time segments;
5. Increment the schedule time ($T = T + 1$) and repeat steps 3 to 5 until all time segments are in the allocated set;
6. Set the project duration D_{proj} to be equal to the maximum finish time of all time segments.

$$D_{proj} = \max(FT_{ij}) \quad (4.4)$$

To illustrate the forward pass calculation that incorporates resource allocation, a simple case study that was reported in Ahuja et al. (1994) was considered (Figure 4.12). This case has also been used in several studies on resource scheduling (Kim and de la Garza 2005b; Lu and Li 2003; Lu and Lam 2008). The case study involves nine activities and requires one type of resource (labor) with a daily availability limit of six. The activity durations and resource requirements are shown on the Activity-on-Node (AON) network in Figure 4.12. The CPS resource allocation process that was applied to this case is illustrated in Table 4.2.



(a) AON network for a case study



(b) CPS equivalent for the case study

Figure 4.12: AON network and its CPS equivalent for the case study

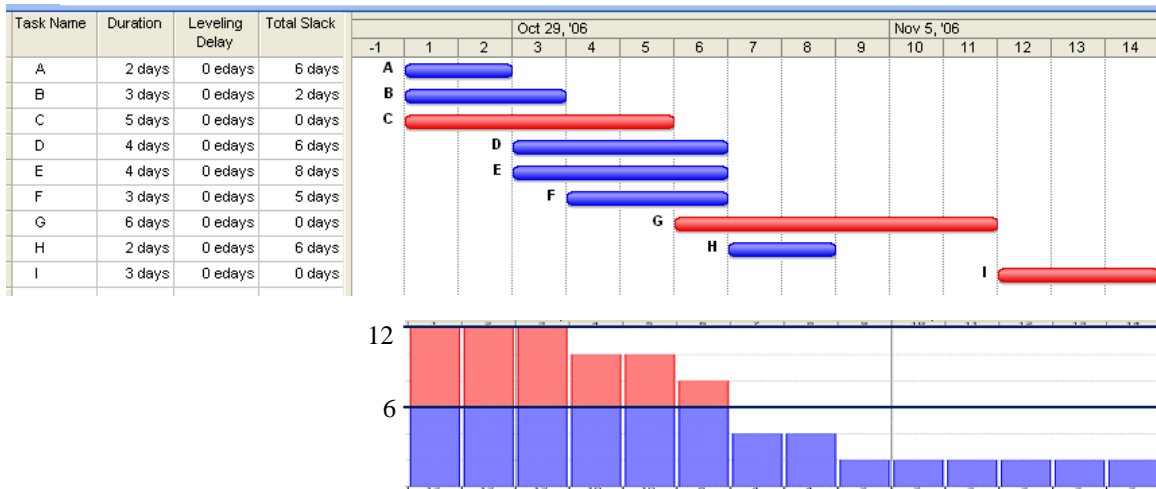
Table 4.2: CPS resource allocation for the case study

Time	Eligible Time Segment	Resource (Limit = 6)	Activity Duration	GRD Rule	Decision	Finish Time
0	A1	4	2	8	Delay	-
	B1	4	3	12	Delay	-
	C1	4	5	20	Start	1
1	A1	4	2	8	Delay	-
	B1	4	3	12	Delay	-
	C2	4	5	20	Start	2
2	A1	4	2	8	Delay	-
	B1	4	3	12	Delay	-
	C3	4	5	20	Start	3
3	A1	4	2	8	Delay	-
	B1	4	3	12	Delay	-
	C4	4	5	20	Start	4
4	A1	4	2	8	Delay	-
	B1	4	3	12	Delay	-
	C5	4	5	20	Start	5
5	A1	4	2	8	Delay	-
	B1	4	3	12	Start	6
6	A1	4	2	8	Delay	-
	B2	4	3	12	Start	7
7	A1	4	2	8	Delay	-
	B3	4	3	12	Start	8
8	A1	4	2	8	Start	9
	F1	2	3	6	Delay	-
	G1	2	6	12	Start	9
9	A2	4	2	8	Start	10
	F1	2	3	6	Delay	-
	G2	2	6	12	Start	10

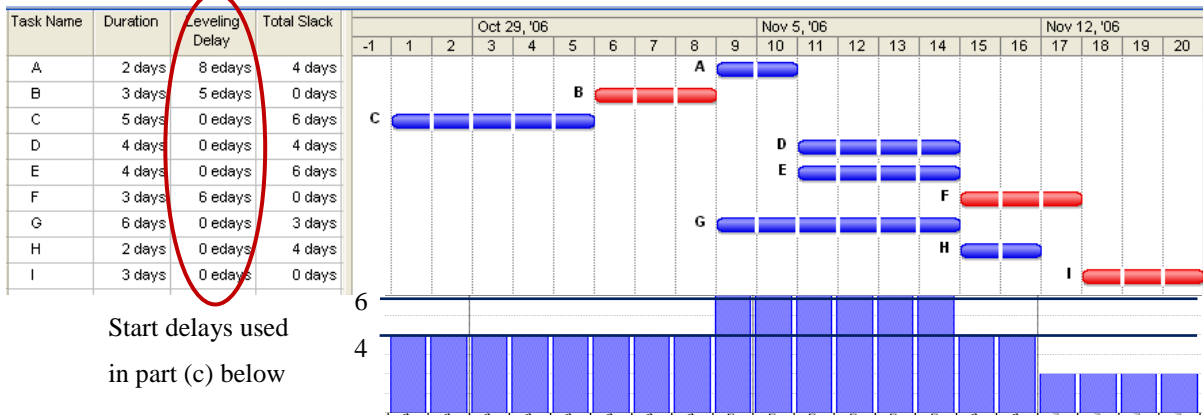
10	G3	2	6	12	Start	11
	F1	2	3	6	Delay	-
	D1	3	4	12	Start	11
	E1	1	4	4	Start	11
11	G4	2	6	12	Start	12
	F1	2	3	6	Delay	-
	D2	3	4	12	Start	12
	E2	1	4	4	Start	12
12	G5	2	6	12	Start	13
	F1	2	3	6	Delay	-
	D3	3	4	12	Start	13
	E3	1	4	4	Start	13
13	G6	2	6	12	Start	14
	F1	2	3	6	Delay	-
	D4	3	4	12	Start	14
	E4	1	4	4	Start	14
14	F1	2	3	6	Start	15
	H1	2	2	4	Start	15
15	F2	2	3	6	Start	16
	H2	2	2	4	Start	16
16	F3	2	3	6	Start	17
17	I1	2	3	6	Start	18
18	I2	2	3	6	Start	19
19	I3	2	3	6	Start	20

Using the GRD rule, the resulting resource-loaded schedule indicates that the project duration is extended from 14 days to 20 days, as shown in Figure 4.13.

a) The schedule before resource allocation



b) The schedule after resource allocation using the GRD rule



c) Network after resource allocation, showing start delay values

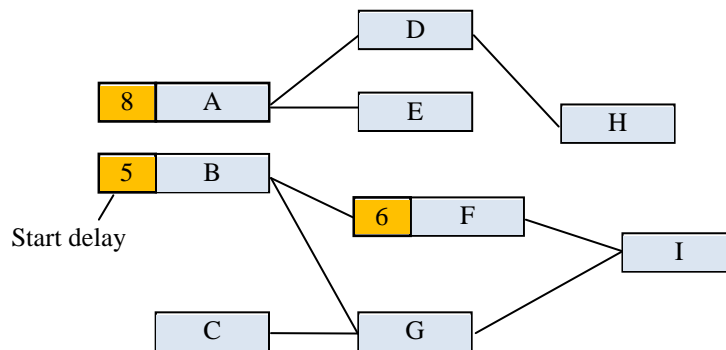


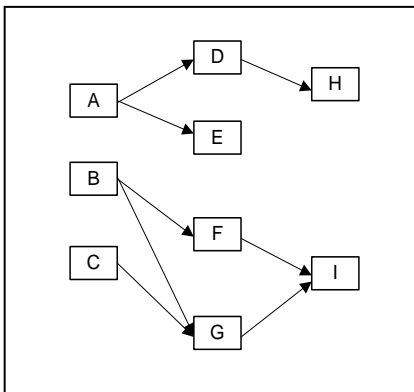
Figure 4.13: Resource allocation for the case study using the GRD rule

4.3.4 Step 4a: Float Calculation in the Case of No Resource Constraints

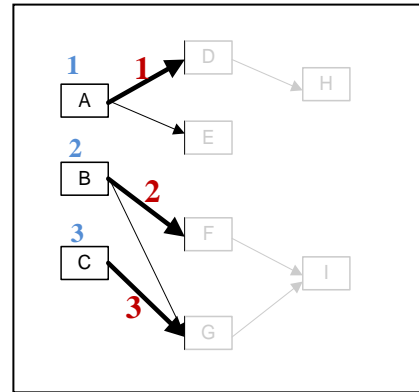
In the case of no resource constraints, this research generalizes the forward pass scheduling process explained in section 4.2 so that it can be applied to the CPS mechanism. While the forward pass is ongoing, the CPS total float determination process first identifies all the paths in the project network and determines the total float of each path. Each time segment is then assigned a total float value based on the total float of the paths it inhabits. The process is structured as follows:

1. Determine all the paths and their durations:
 - a. Starting with time segments that have no predecessors, set a path number for each time segment (PN_j) so that $PN_j = 1$ for the first time segment, and $PN_i = \text{largest } PN_j + 1$ for each of the other time segments (Figure 4.14). Steps 2 to 6 in Figure 4.14 show how paths are indicated along the forward pass. The end result (step 6) indicates for each activity which paths it lies on.
 - b. For each time segment, identify all immediate successors. For the first successor set its path number to be equal to the path number of its predecessor so that $PN_{j(\text{pred})} = PN_{j(\text{Suc})}$. For the remaining successors, set a different path number so that each time segment has a path number $PN_j = \text{largest } PN_j + 1$.
 - c. Once a path number has been assigned to each successor, the path number for the predecessor is updated so that the predecessor time segment belongs to all the paths assigned to its successors.

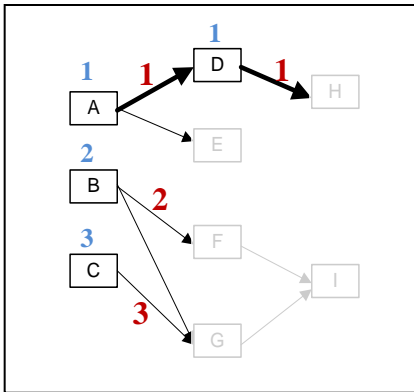
Step 1: Network



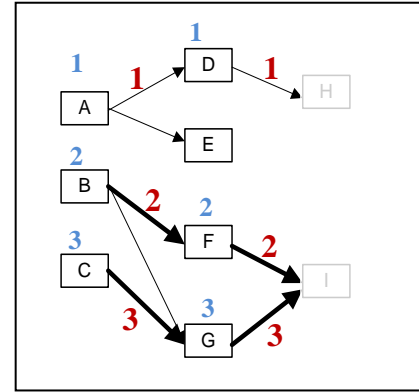
Step 2: Define paths 1, 2, and 3



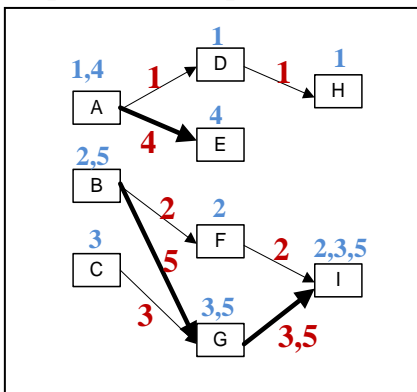
Step 3: Continue path 1



Step 4: Continue paths 2 and 3



Step 5: Continue paths 1, 4 and 5



Step 6: Activity paths

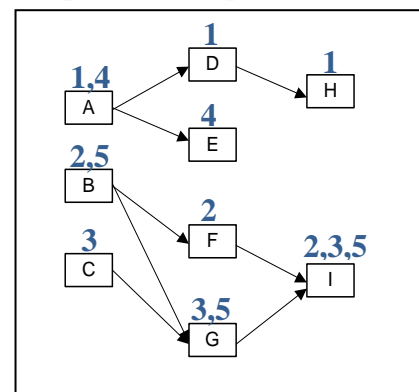


Figure 4.14: Example of the CPS total float determination process

2. Calculate the duration of each path, and identify the longest path (Table 4.3):

$$Path\ Duration = \sum [Start\ Delay\ (SD_{ij}) + Duration\ of\ the\ Time\ Segment\ (D_{ij})] \quad (4.5)$$

- a. Calculate the float for each path as follows:

$$PF = Duration\ of\ the\ longest\ path\ (D_{LP}) - Duration\ of\ the\ current\ path\ (D_P) \quad (4.6)$$

3. Assign a total float value for each time segment based on the float of the time segment's paths as follows:

$$TF_j = \min (PF_j) \quad (4.7)$$

To illustrate the CPS process for total float calculation, the previous case study explained in Figure 4.12 and Figure 4.14 was considered. Step 6 in Figure 4.14 shows the network and all the five paths identified. In the case of no resource constraints, the total float for each path is calculated as shown in Table 4.3

Table 4.3: Calculation of path float

Path	Path Duration	Longest Path	Path Float
1. A-D-H	2+4+2= 8	14	14 -8= 6
2. B-F-I	3+3+3= 9		14 -9= 5
3. C-G-I	5+6+3= 14		14 -14= 0
4. A-E	2+4= 6		14 -6= 8
5. B-G-I	3+6+3= 12		14 -12= 2

To simplify the illustration and because all the relationships between the activities are FS relationships, the total float values are determined at the activity level rather than the time segment level since the time segments of each activity have the same total float value.

In the case of no resource constraints, the path duration is calculated at the activity level as follows:

$$Path\ Duration = \sum (Activity\ Duration) \quad (4.8)$$

Based on the total float value of each path in the network, each activity is then assigned a total float value, as shown in Table 4.4.

Table 4.4: Activity total floats based on the path floats

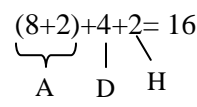
Activity	Paths	Path Total Float	Activity Total Float
A	1, 4	6 (path 1), 8 (path 2)	Min (6,8) = 6
B	2, 5	5 (path 2), 2 (path 5)	Min (5,2) = 2
C	3	0	0
D	1	6	6
E	4	8	8
F	2	5	5
G	3, 5	0 (path 3), 2 (path 5)	Min (0,2) = 0
H	1	6	6
I	2, 3, 5	5 (path 2), 0 (path 3), 2 (path 5)	Min (5,0,2) = 0

4.3.5 Step 4b: Float Calculation with Consideration of Resource Constraints

The approach explained in the previous section can be also used to calculate initial total float values after resource allocation during the CPS forward pass calculation. For the same case study illustrated in Figure 4.12 and based on the results of resource allocation shown in Table 4.2 and Figure 4.13, the total float for each path and the total float value for each activity were calculated as shown in Figure 4.15 and Table 4.5, respectively.

After resource allocation, the path duration can be calculated as follows:

$$\text{Path Duration} = \sum (\text{Start Delay of the activity} + \text{Activity Duration}) \quad (4.9)$$

Path	Path Duration	Longest Path	Path Folat
1. A-D-H	$(8+2)+4+2= 16$ 	20	$20-16= 4$
2. B-F-I	$(5+3)+(6+3)+3= 20$		$20-20= 0$
3. C-G-I	$5+6+3= 14$		$20-14= 6$
4. A-E	$(8+2)+4= 14$		$20-14= 6$
5. B-G-I	$(5+3)+6+3= 17$		$20-17= 3$

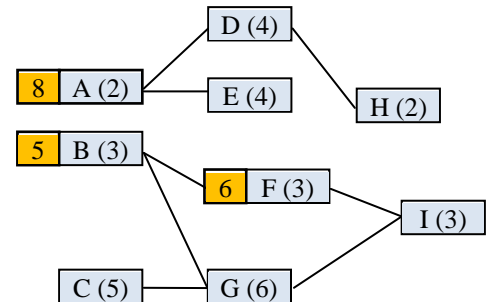


Figure 4.15: Calculation of path floats after resource allocation

Table 4.5: Activity total float values after resource allocation

Activity	Paths	Path Total Float	Activity Total Float
A	1, 4	4 (path 1), 6 (path 2)	Min (4,6) = 4
B	2, 5	0 (path 2), 3 (path 5)	Min (0,3) = 0
C	3	6	6
D	1	4	4
E	4	6	6
F	2	0	0
G	3, 5	6 (path 3), 3 (path 5)	Min (6,3) = 3
H	1	4	4
I	2, 3, 5	0 (path 2), 6 (path 3), 3 (path 5)	Min (0,6,3) = 0

It should be noted that these total float values are the same total floats calculated by any existing scheduling software. However, these total float values are based on the assumption that an activity can be delayed by these values without extending the project duration or violating logical relationships. There is no check to determine whether delaying the activity within these values will cause resource over-allocation that would need to be resolved again and that may delay the project in this case. These total float values therefore need refinement so that they reflect the criticality of the activities not only in terms of precedence constraints but also in terms of resource constraints.

4.3.6 Step 4c: Float Refinement

In the case of resource-constraint scheduling, CPS uses a refinement approach (Figure 4.16) to refine the total float values obtained from the forward pass and described in the previous section. The primary purpose of the refinement approach is to calculate accurate total float values that consider resource-related constraints in addition to precedence constraints.

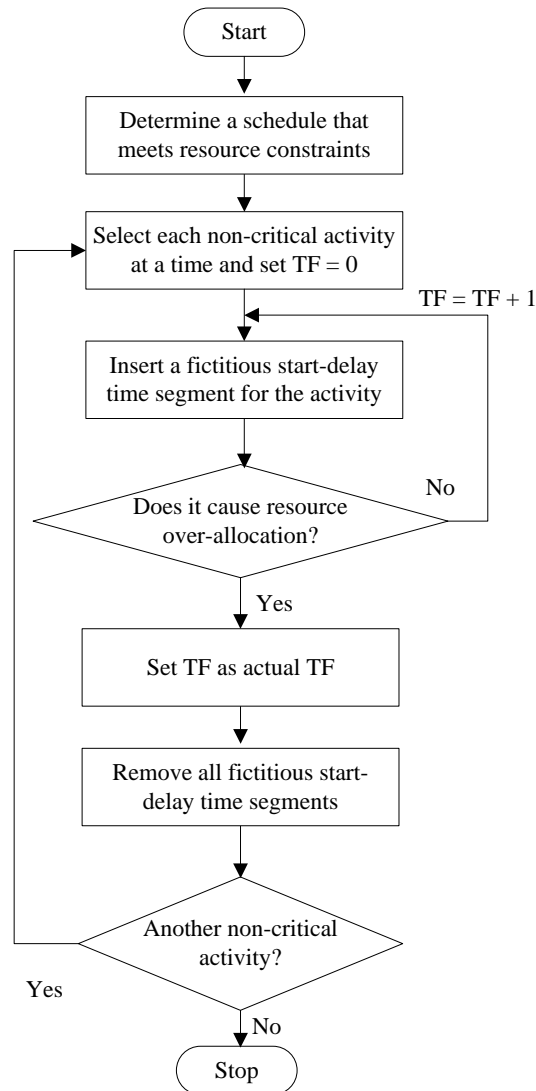


Figure 4.16: CPS total float refinement with consideration of resource constraints

During the adjustment process, CPS begins with the schedule determined by the forward pass calculations and examines one activity at a time. Using a fictitious time segment, the process inserts a start-delay for each non-critical activity and analyzes its impact on resources. The non-critical activities in the network are identified based on their initial total float values, which are calculated as described in the previous section. Using the case study shown in Figure 4.12 and Figure 4.17, the total float refinement approach was simulated as shown in Figure 4.18.

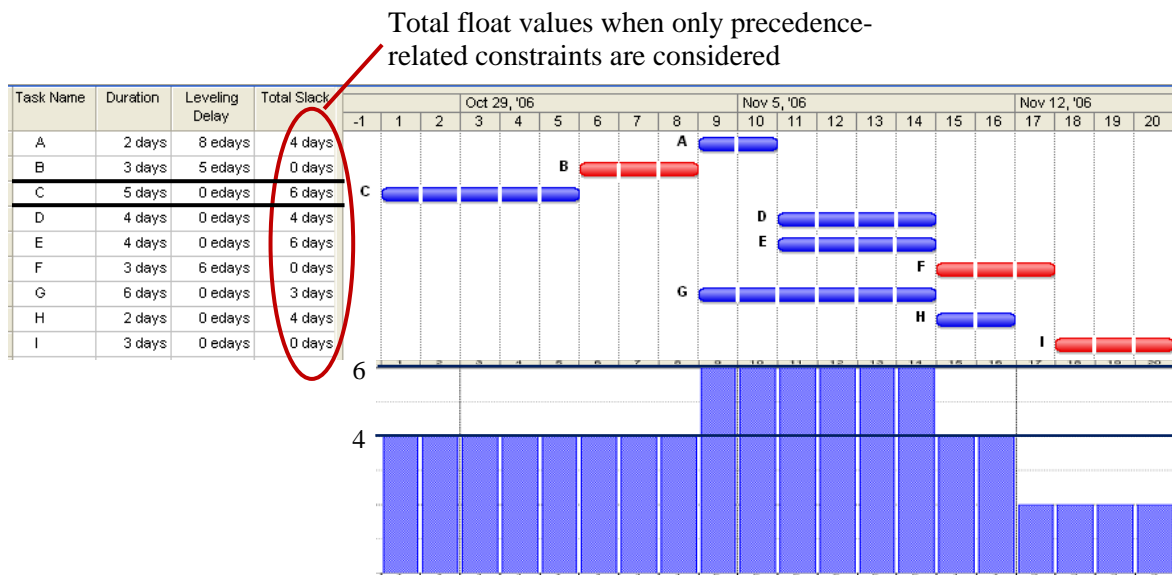


Figure 4.17: Total float calculation in the case of resource constraints

Although Figure 4.17 shows that activity C has a total float of 6 days, delaying activity C by only one day causes resource over-allocation, as shown in Figure 4.18. Accordingly, the actual total float value for activity C should be zero (Table 4.6).

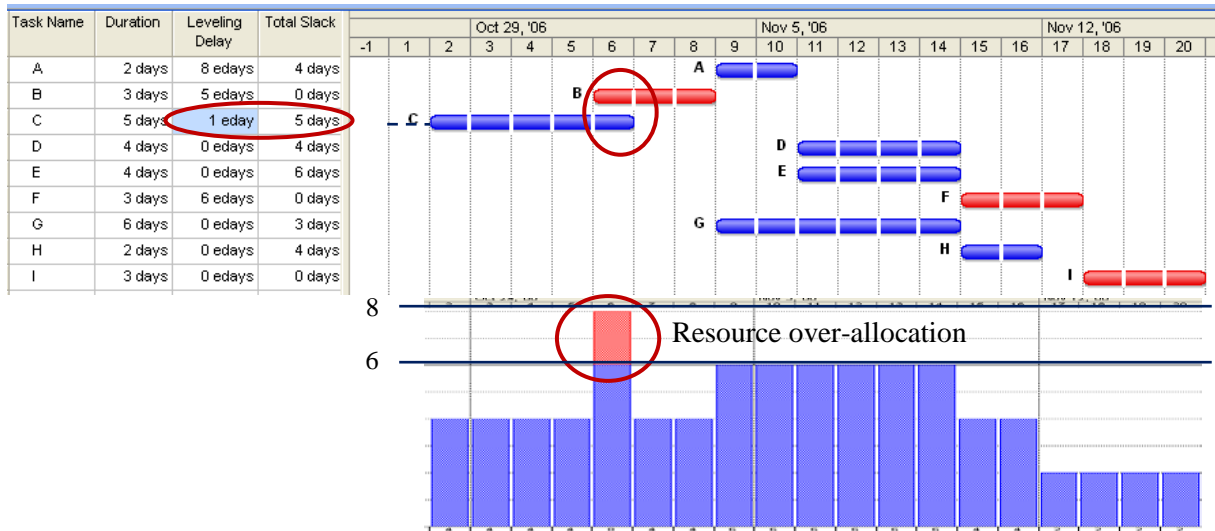


Figure 4.18: Simulation of the float refinement process

Table 4.6: Refined total float values after resource-related constraints are considered

Activity	Total Float
A	4
B	0
C	0 (resource critical)
D	4
E	6
F	0
G	3
H	4
I	0

It should be noted that this refinement process is different from the approach presented by Lu and Lam (2008) in which the duration of each activity was extended one day at a time and the project duration was monitored after any resource over-allocations were resolved. CPS inserts a start-delay time segment into each non-critical activity and analyzes its impact on the resource schedule; the duration of the activity in this case is kept intact as determined according to the appropriate construction method used, particularly if durations are determined automatically through their integration with estimating functions.

4.4 Generalized CPS Formulation with Consideration of Progress Information

All of the previous formulation has dealt with project scheduling before the start of construction. The scheduling process that takes place during construction and progress analysis needs special attention. Many activities can be only partially complete, with part of the work remaining. The resolution of resource constraints for ongoing activities and the determination of corrective actions therefore need to be incorporated into the CPS scheduling process that is applied to the remaining part of ongoing activities. This CPS formulation is described in detail following an explanation of the basic terminology used to define the types of schedules experienced during execution:

4.4.1 Baseline Schedule

The baseline schedule is developed at an early stage of the project for the purpose of tracking progress and payments. The development of a baseline schedule is crucial for the success of any construction project because it can be used not only for project control but also for legal matters such as claim analysis. The baseline data used to develop this schedule includes the

project activities, their expected durations, and the logical relationships among those activities. The baseline saves all this information (Figure 4.19a) and is used as a starting point for comparing actual progress. It should be noted that the first baseline is effective from the project's planned start date. If changes in the duration or/and logical relationships of the activities result in more than one baseline being saved during the execution of the project, then progress is always compared to the last baseline saved.

4.4.2 Current Schedule

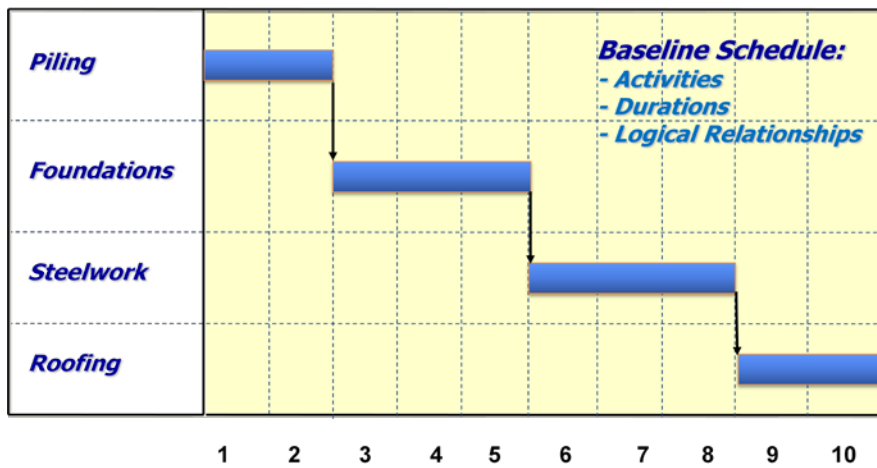
The current schedule is a combination of two parts: the actual schedule (the left side of Figure 4.19b) plus the remaining schedule (the right side of Figure 4.19b and Current 2 in Figure 4.20). At the beginning of the project, the current schedule consists of only the remaining schedule since the actual part is zero. Similarly, at the end of the project, the current schedule consists of only the actual schedule since the remaining part is zero.

The actual schedule represents all the events that have actually happened and that have been documented from the planned start date of a project to a specific progress date. It is therefore fixed and cannot be changed. At the beginning of a project, the amount of work represented by the actual schedule is zero, and also at the end of the project, the actual schedule represents the full as-built schedule.

The remaining schedule represents the calculated portion of unfinished work, from the last progress date until all tasks are 100% complete. At the beginning of the project, the amount of work remaining is 100% (Current 1 in Figure 4.20), while at the end of the project, the percentage remaining is zero. As more actual work is completed, the remaining portion

becomes smaller. Because the remaining schedule is variable and sensitive to any changes in activity data, sequence changes, and rearrangement of remaining time segments, it represents the schedule's flexibility with respect to the incorporation of corrective action.

a) Baseline schedule



b) Current schedule that combines actual and remaining schedules

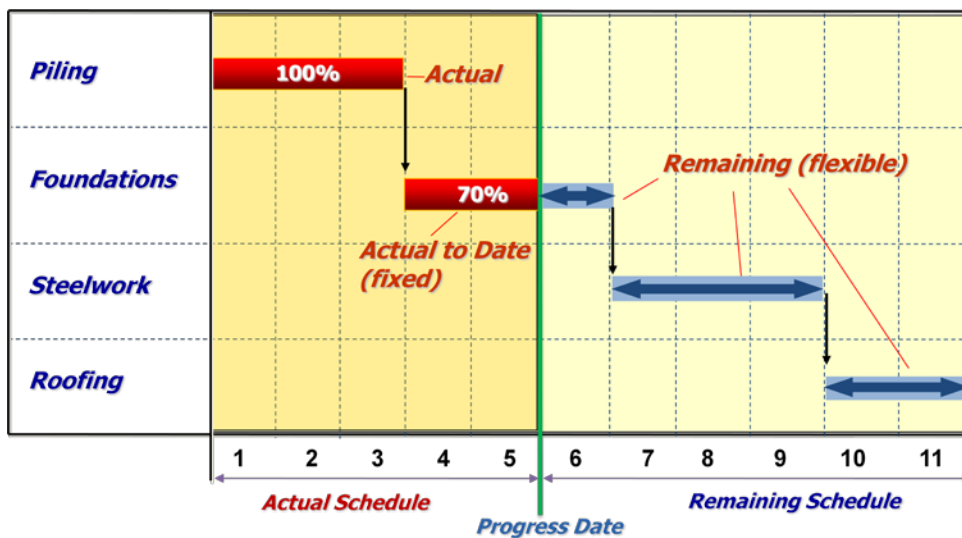


Figure 4.19: Different types of schedules

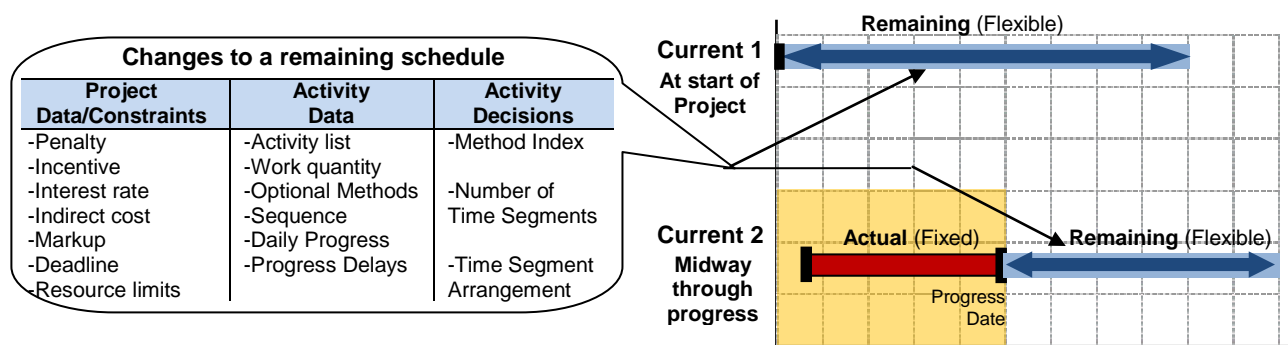


Figure 4.20: Portion of the schedule that represents flexibility for corrective action

4.5 Detailed Scheduling Mechanism

As the above description of the different types of schedules indicates, all data, decisions, and calculations apply only to the remaining schedule because it is the basis of the current schedule and baselines. Accordingly, the CPS generic scheduling process and related calculations are as follows:

1. For each activity, a decision is made with respect to which execution method is used (*Mi*). The total duration, cost, and resources for an activity are therefore known; a decision must be made with respect to the appropriate segment size, typically one day.
2. If a project has not yet started, but a baseline was previously saved, then baseline data, which includes activities' relationships, the activities execution methods; the detailed baseline time segments, and the delays determined for any time segments, is loaded. No calculations are performed unless any changes are made.
3. If a project has started but not yet finished (i.e., the last progress date is greater than the project start date), then a baseline and actual information exist. The first step is to load

the last baseline schedule, activities' relationships, the activities execution methods, the detailed baseline time segments, and the delays determined for any time segments. Against this baseline, the current schedule is then formulated from the fixed actual part and from the calculated remaining part that represent the schedule from the date of current progress until the end of the project.

4. Fix the schedule of the actual part (segments), and then calculate the remaining duration (R_{Di}) for each activity (i), having a total duration specified in the construction method (M_i)
5. Calculate the remaining duration (number of time segments) based on either the planned production or the actual production as follows:

$$RD_i = \text{Roundup} [(1 - \%complete)/\text{planned progress}] \text{ or} \quad (4.10)$$

$$RD_i = \text{Roundup} [(1 - \%complete)/\text{actual progress}] \quad (4.11)$$

4. Calculate the production (P_{ij}) for each time segment (j) in activity (i) as follows:

$$P_{ij} = P_{\text{planned } (mi)} \text{ or } P_{\text{actual } i} \text{ (so that } \sum P_{ij} = 1.0) \quad (4.12)$$

5. The total cost for each segment (C_{ij}) is then calculated as follows:

$$C_{ij(\text{Planned})} = (P_{ij} \% * C_{mj})/100 \quad (4.13)$$

$$C_{ij(\text{actual})} = (P_{ij} \% * C_{\text{actual}})/\sum P_{\text{actual}} \quad (4.14)$$

6. During the course of the actual work, the baseline schedule is updated with the actual progress. After each update, the proposed model again performs the CPS analysis in order to calculate the project duration while taking into consideration any new constraints. The new project duration is then compared with the previous duration. If the project is expected to be delayed, the following steps occur:
 1. Schedule analysis (Hegazy and Menesi 2008b) is performed in order to allocate responsibility for the delay.
 2. The optimization model (described in Chapter 5) is used to optimize the schedule and suggest corrective action in order to recover the project delays.

4.6 Summary and Conclusions

This chapter has presented the mathematical formulation for the critical path segments method (CPS). A detailed mechanism for converting networks with different types of relationships into CPS networks has been described along with illustrative cases. The CPS forward pass calculations, which also include a resource allocation mechanism, have then been presented. It should be noted that, for calculation purposes, CPS uses only forward pass analysis without the need for a backward pass. Two different approaches for calculating total float values have been introduced. One approach is used to calculate initial total float values in the case of no resource constraints, while the other approach calculates accurate total float values that consider both precedence-related and resource-related constraints. The CPS formulation for progress analysis has also been described and the common types of schedules have been defined.

Chapter 5

Resolving Project Constraints in CPS

5.1 Introduction

Real-life projects often involve multiple constraints and challenges. Schedules may not simulate reality if they do not incorporate the project constraints, such as activity durations, the project deadline, and the limited availability of resources. Neglecting these constraints in the scheduling process can affect project control and hence delay the completion of the project. This chapter first describes the traditional optimization formulation that is used with CPM. A general optimization formulation that can be used with CPS is then presented. The difference between schedule optimization at the activity level and at the time-segment level is described along with an example that demonstrates the benefits of optimization at the time segment level.

5.2 Resolving Constraints in CPM (Activity Level)

To demonstrate the simple approach to facilitating the resolution of multiple constraints for continuous activities proposed by Hegazy (2006), a small but comprehensive case study can be considered (Figure 5.1). The estimates for the four activities in the case study are shown. The general project information indicates a strict 10-day deadline, a late penalty of \$2,000 per day, a \$100 per day indirect cost, and a strict resource limit of 2 per day. It should be noted that some activities have more than one estimate in order to represent practical options that vary from inexpensive and slow to fast and expensive. These estimates can be based on considerations such as different subcontractor quotes or the option of using crews for their normal hours versus adding overtime.

Deadline = 10 days & Resource Limit = 2 / day

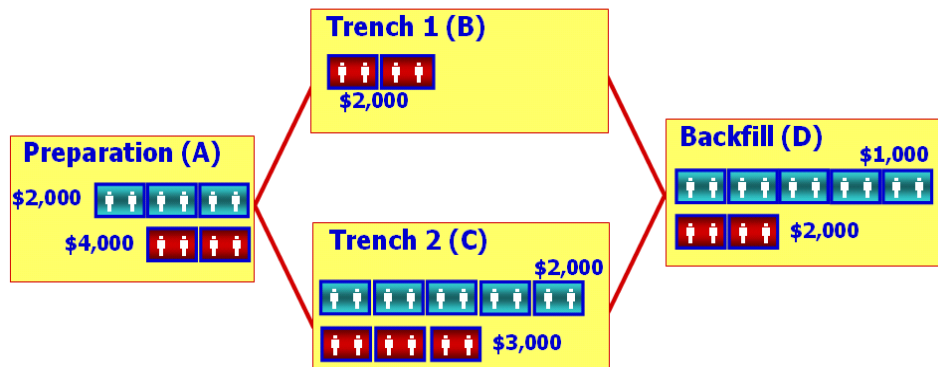


Figure 5.1: Case study activities and their optional estimates

A quick look at the project network reveals that activities "Trench 1" and "Trench 2" run in parallel and require 4 resources (limit is 2). In addition, using the cheapest method (estimate 1) for each activity, the project duration becomes 13 days (3 days beyond the deadline) with a total cost of \$14,300 (\$7,000 direct cost + \$1,300 indirect cost + \$6,000 penalty). To meet the constraints, i.e., the deadline and resource limit, it is possible to experiment with a variety of decisions. Given the sequence of the activities and the various construction options, it is possible to arrive at the least-expensive plan that meets both the deadline and the resource limit. The solution shown in Figure 5.2, for example, represents a plan with a 10-day project duration, which meets the deadline, and in which all the activities are scheduled so that the resource limit is not exceeded.

It is important to note that the solution shown in Figure 5.2 includes two required quantitative decisions: (a) an index of the method selected from the optional estimates for each activity that makes a good trade-off between the duration and the cost of the activity (i.e., TCT

analysis); and (b) the start-delay values, which apply only to the start of a specific activity, that will ensure that resource over-allocations are resolved.

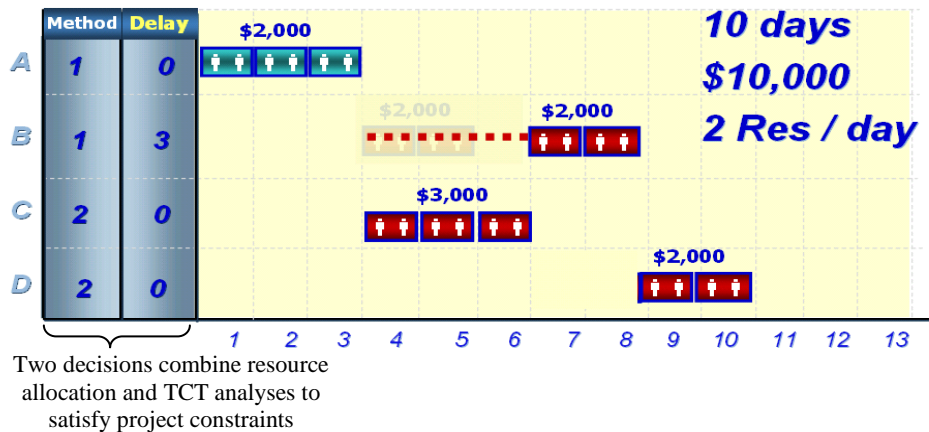


Figure 5.2: Details of a solution that meets all constraints

This simple case study shows that resolving CPM constraints requires the identification of the activity options and a proper mechanism for determining optimum values for activity variables. The two activity decisions shown in Figure 5.2 thus represent key variables that govern how corrective actions are performed during actual progress. If a project is delayed, for example, then a suitable corrective action is to choose modified values for the two decisions. The corresponding calculations for activity duration, floats, and baselines and the consequent modifications to the critical path(s) thus need to be formulated as a function of these variables. Almost no commercial scheduling software systems, however, have the ability to deal with this simple case study.

5.3 Resolving Constraints in CPS (Time-Segment Level)

The approach presented above is even more powerful for facilitating decisions when it is reformulated to work with the developed model. Because each time segment of an activity is treated as a separate activity, the start-delay decision is reformatted because it affects only the starting segment of an activity.

Several activities in construction projects can be split with only low or negligible startup and restarting costs. To better represent practical instances in actual construction projects, the model allows the inner segments of an activity to be flexible so that they can be adjusted as well. This feature is somehow similar to the suggestion by Son and Mattilla (2004), who permit selected activities to stop and restart. Buddhakulsomsiri and Kim (2007) also proved that the splitting of activities results in an improvement to the resource-leveling solution. The resource-leveling algorithm in the CPS model therefore produce more practical and realistic schedules because it enables all individual time segments to be stopped and restarted, as necessary, so that a limited resource can be reallocated to a more critical activity. A generic representation of decisions in the CPS model is shown in Figure 5.3.

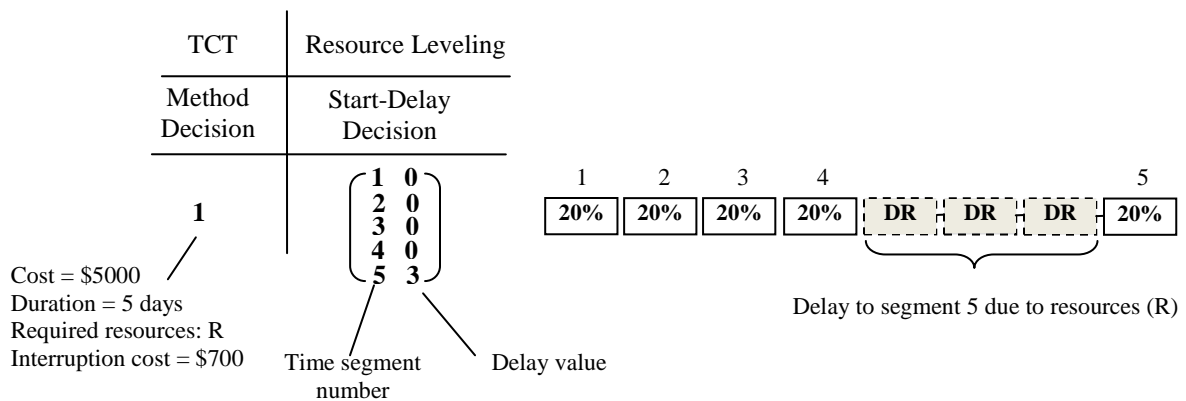


Figure 5.3: CPS representation of resource leveling and TCT decisions

Several schemes, such as mathematical approaches, heuristic methods, and genetic algorithms (GAs), have been developed to solve the resource-constrained project scheduling problem. However, most of these approaches are based on the assumption that activities in progress are non-preemptive or that they cannot be interrupted. As a result, very little is known about the potential benefits of preemption for solving the resource-constrained scheduling problem (Zhang et al. 2006). Since the CPS analysis is performed at the time segment level rather than the activity level, preemption is permitted at any integer time instant. However, unnecessary splits in activities may be introduced by the search algorithm. It is important, therefore, for resource allocation and schedule optimization, to add a constraint that minimizes the disruption to activities.

5.4 Schedule Optimization for Project Control (Activity Level)

As the project starts, the schedule is updated with the actual progress, which may require rescheduling of the remaining part of the project. Schedule optimization for project control is performed during the execution of the project in order to continue satisfying the project objectives with respect to time, cost, and resources. Two common optimization procedures for project control are to resolve resource constraints, and to meet the project duration deadline. These optimization procedures are described in the following subsections, which highlight the respective objective function, variables, and constraints for each procedure.

5.4.1 Resolving Resource Constraints

The general formulation of the resource-constrained scheduling problem is as follows (Tabot and Patterson 1978):

$$\text{Minimize } f_N \quad (5.1)$$

Subject to

$$\max_{n \in P_j} \{f_n\} + d_j \leq f_j, \quad j = 1, \dots, N, \quad (5.2)$$

$$\sum_{j \in S_t} r_{jk} \leq R_{kt}, \quad k = 1, \dots, K; \quad t = 1, \dots, HP \quad (5.3)$$

where f_N is the finish time for the last activity, P_j is the set of all immediate predecessors of activity j , d_j is the duration of activity j , f_j is the finish time for activity j ($j = 1, \dots, N$), N is the total number of activities in the project, S_t is the set of eligible activities in time period t , r_{jk} is the amount of resource k required by activity j , R_{kt} is the amount of resource k available in time period t ($k = 1, \dots, K$), K is the number of resource types, t is the time period ($t = 1, \dots, HP$), and HP is a known completion time for the project.

The objective is to minimize the completion time of the unique finish activity in the project, thereby minimizing project duration. Precedence relationships are maintained by Eq. (5.2). The resource constraint given in Eq. (5.3) insures that resource usage does not exceed resource availability in any given time period.

Optimization Variables for CPM (Activity Level): The variables consist of the activity delay values that resolve resource over-allocations.

Optimization Variables for CPS (Time-Segment Level): The variables consist of time-segment delay values that resolve resource over-allocations. Based on the user input regarding the permissibility of interrupting an in-progress activity, CPS considers the

possibility of delaying time segments that are not the first time segment of an activity in order to resolve resource constraints.

Optimization Constraints: The constraint is that the daily resources required must be less than or equal to the daily resources available.

5.4.2 Meeting Deadline Duration

To complete activities, project managers may choose different equipment, crew sizes, and construction methods; their selections generally involve a trade-off between time and cost. Since most construction projects must be completed within a specific duration, a time-cost trade-off analysis is essential for optimizing the activity durations so that the result is the desired project duration at a minimum cost.

The general formulation of a time-cost trade-off analysis is as follows:

$$\text{Minimize } C = \sum_{i=1}^n \sum_{k=1}^{m_i} C_{ik} X_{ik} \quad (5.4)$$

Subject to

$$\sum_{k=1}^{m_i} X_{ik} = 1 \quad i = 1, 2, \dots, n \quad (5.5)$$

$$SF_i - SF_p - \sum_{k=1}^{m_i} d_{ik} X_{ik} \geq 0 \quad p = 1, 2, \dots, NP \quad (5.6)$$

$$SF_E \leq DL \quad E = 1, 2, \dots, NE \quad (5.7)$$

where C_{ik} is the cost of activity i when the k th option is employed, X_{ik} is a zero-one variable for activity i when the k th option is employed; SF_i is the scheduled finish time of activity i ;

SF_p is the scheduled finish time of the predecessor of activity i ; d_{ik} is the duration of activity i when the k th option is employed; NP is the number of activities preceding activity I ; SF_E is the scheduled finish time for ending activities; NE is the number of ending activities; and DL is the deadline duration of the project, for which the upper and lower bound on DL are the normal project duration and crash project duration, respectively.

Optimization Objective: The optimization objective is to minimize the project duration in the least costly manner based on the application of time-cost trade-off analysis. In CPS, the cost of interrupting an activity is also considered in the analysis.

Optimization Variables: The variables consist of the construction method indices that provide a good trade-off between the duration and the cost of the activities in order to meet the project deadline.

Optimization Constraints: The constraint is that the project duration should be less than or equal to the deadline duration.

5.5 Generalized Optimization Formulation for CPS

Consider a construction project with n activities, each having a duration (D_i) and cost (C_i), as specified in the construction method (M_i). Every construction method M_i represents a way of carrying out activity i .

To insure that only one construction method is selected per activity, a zero-one variable X , is introduced for each construction method for each activity. The duration (D_i), which also

indicates the number of time segments, and cost (C_i) for an activity i , can be expressed in terms of the zero-one variable as follows:

$$D_i = D_{i1}X_{i1} + D_{i2}X_{i2} + \dots + D_{iM_i}X_{iM_i} = \sum_{k=1}^{M_i} D_{ik} X_{ik} \quad (5.8)$$

$$C_i = \sum_{k=1}^{M_i} C_{ik} X_{ik} \quad (5.9)$$

where X_{ik} is a zero-one variable that belongs to the construction method number k for activity i . If $X_{ik} = 1$, then the k th option will be used to perform activity i , while $X_{ik} = 0$ means the otherwise. The sum of zero-one variables of all options should be equal to 1.

$$\sum_{k=1}^{M_i} X_{ik} = 1 \quad i = 1, 2, \dots, n \quad (5.10)$$

If the direct cost of the project is noted as PDC and the indirect cost as PIC , the total project cost (TPC) can be expressed as

$$TPC = PDC + PIC \quad (5.11)$$

The direct cost of the project is the summation of the costs of all activities and can be expressed mathematically as

$$PDC = \sum_{i=1}^n \sum_{k=1}^{M_i} C_{ik} X_{ik} \quad (5.12)$$

The indirect costs of the project are time dependent: the longer the project duration, the more indirect costs are incurred. The relationship between PIC and the project duration (T) can be expressed as

$$PIC = IC_0 + IC * T \quad (5.13)$$

where IC_0 is the initial cost (e.g., permits, mobilization cost, temporary hookups, temporary facilities, purchase advances), IC is the indirect cost per time period (i.e., daily expenditures), and T is the total project duration.

The CPS forward pass calculates the project duration (T), the earliest start time (ES_i), the latest start time (LS_i), the earliest finish time (EF_i), the latest finish time (LF_i), and total float (TF_i) for each activity i .

The CPS optimization model allows the interruption of an activity in order to resolve resource over-allocations. However, an added cost is associated with such interruptions, such as startup and restarting costs. The objective is to resolve resource over-allocations in a way that provides a trade-off between the extra cost of delays versus the extra cost of the interruption of the activity. If the cost of interrupting an activity i is noted as Ct_i and the number of interruptions created in activity i to resolve resource over-allocation is noted as NS_i , the total cost of interruption (TC_{int}) can be expressed as

$$TC_{int} = \sum_{i=1}^n Ct_i NS_i \quad (5.14)$$

If the project is delayed or expected to be delayed, the project delay cost (C_{pd}) can be expressed as

$$C_{pd} = Y * C_d * Dt \quad (5.15)$$

where Y is a zero-one variable representing project delay so that $Y = 1$ when the deadline duration is exceeded, C_d is the cost of the delay per time period (e.g., a delay penalty), and Dt is the number of time periods delayed (actual duration – deadline duration).

If the project is completed or expected to be completed before the deadline, the incentive for early completion can be expressed as

$$IN = Z * B * S_v \quad (5.16)$$

where Z is a zero-one variable representing an early completion incentive so that $Z = 1$ when the project is finished before the deadline duration, B is the incentive payment per time period saved, and S_v is the number of time periods saved (deadline duration – actual duration).

Objective Function

The objective of the optimization model is to minimize the sum of the costs described above. In addition, a penalty for resource over-allocation can be included in the objective function as a cost in order to avoid any resource over-allocation. The objective function can be formulated as follows

$$\begin{aligned} \text{Minimize } C = & \sum_{i=1}^n \sum_{k=1}^{M_i} C_{ik} X_{ik} + IC_0 + IC * T + \sum_{i=1}^n Ct_i NS_i + Y * C_d * Dt - \\ & Z * B * S_v + RO * \text{penalty} \end{aligned} \quad (5.17)$$

where RO is a zero-one variable that indicates a resource over-allocation so that $RO = 1$ if any resource over-allocation occur during the project.

Network Logic Constraints

The network logic constraints are considered to be hard constraints. The logical relationship between any time segment (ij) and its immediate predecessor (p), can be expressed mathematically as

$$SS_{ij} - SF_p \geq 0 \quad p = 1, 2, \dots, NP \quad (5.18)$$

where SS_{ij} is the scheduled start time of time segment j , which belongs to activity i ; SF_p is the scheduled finish time of the predecessor of time segment ij ; and NP is the number of time segments preceding the time segment ij .

Project Completion Constraint

The project completion constraint is considered to be a soft constraint and is expressed in the optimization model as

$$SF_E \leq DL \quad E = 1, 2, \dots, NE \quad (5.19)$$

where SF_E is the scheduled finish time of the ending time segments, NE is the number of ending time segments, and DL is the deadline duration of the project.

Resource Constraints

The project resource constraint is considered to be a soft constraint and is expressed in the optimization model as

$$\sum_{i \in S_t} r_{ijl} \leq R_{lt}, \quad l = 1, \dots, L; \quad t = 1, \dots, HP \quad (5.20)$$

where S_t is the set of eligible time segments in time period (t), r_{il} is the amount of resource l required by time segment j , which belongs to activity i , R_{lt} is the amount of resource k available in time period t ($l = 1, \dots, L$), L is the number of resource types, t is the time period ($t = 1, \dots, HP$), and HP is the completion time for the project.

Interruption Constraints

The main purpose of this constraint is to minimize the number of interruptions that are created in the activities in order to resolve the over-allocation of resources. The interruption constraint is considered to be a soft constraint and is expressed in the optimization model as

$$\sum_{i=1}^n NS_i = 0 \quad (5.21)$$

where NS_i is the number of interruptions created in activity i in order to resolve resource over-allocation.

5.6 Evolutionary Optimization

Evolutionary techniques have the advantage of not being trapped in the local minimum, and have the ability to search for a near-optimum solution for large size problems (Rogalska et al. 2008, Zheng et al. 2004). In addition, evolutionary techniques provide a number of potential solutions to a given problem, and the choice of the final solution is left to the user. In cases where a particular optimization problem, such as a scheduling problem, does not have one individual solution, then evolutionary algorithms are potentially useful for identifying alternative solutions simultaneously. Because the problem is expected to be large, an evolutionary (random-based) optimization model can be used with CPS in order to search for the optimum schedule that also satisfies the project constraints.

Examples of evolutionary algorithms include genetic algorithms (GA), shuffled frog leaping, particle swarm optimization, and ant colony optimization. As shown in Figure 5.4, implementing evolutionary algorithms involves four main steps: setting the solution structure, determining the evaluation criterion, generating an initial population of solutions, and applying the evolutionary process in order to generate new solutions.

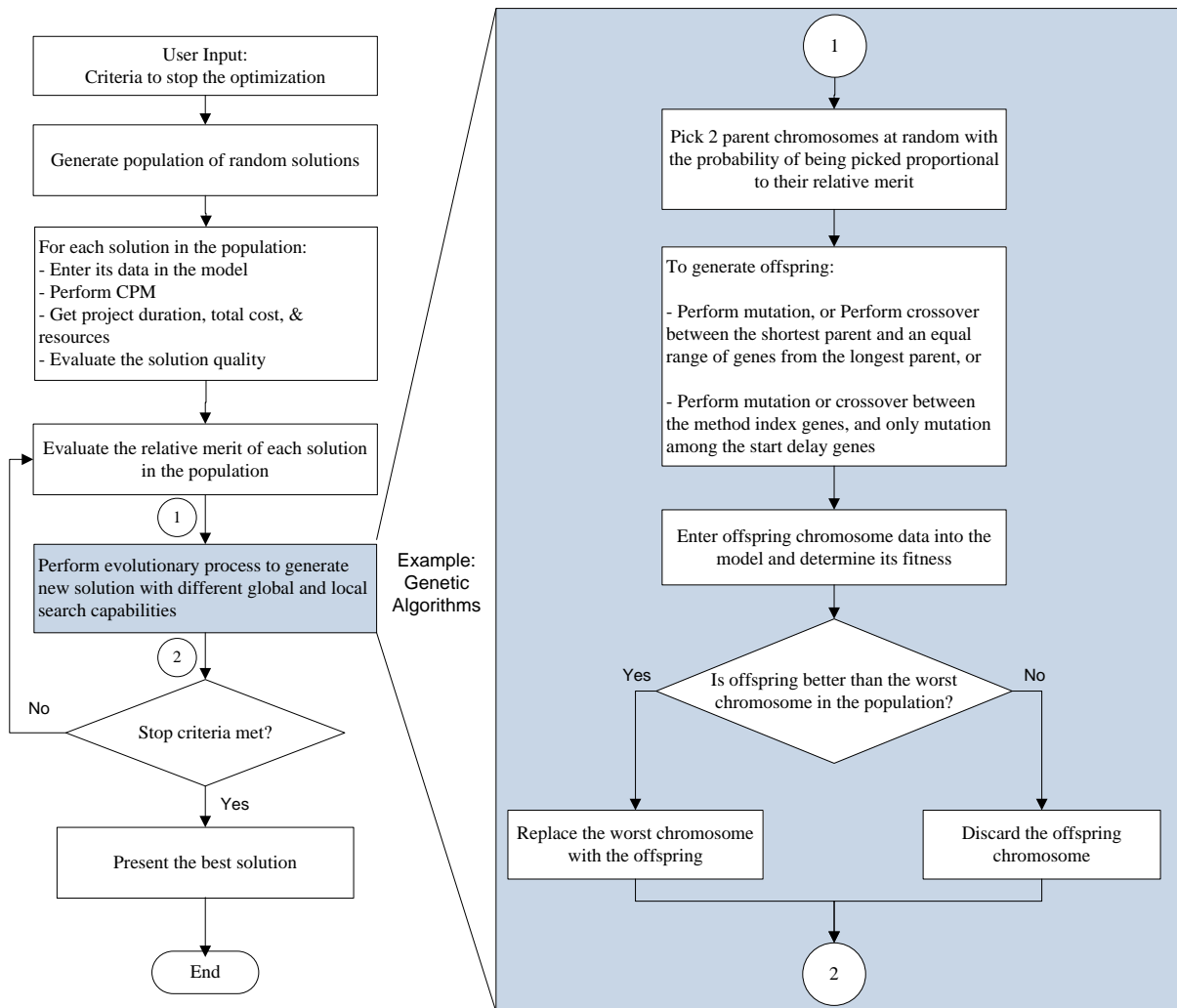


Figure 5.4: Optimization process using evolutionary algorithms

5.6.1 Solution Structure (Activity Level)

The solution structure for optimization at the activity level can be formulated as an array that is twice the length of the number of activities in the project. The first half of the solution structure consists of elements that contain individual delay values for each activity. The second half consists of elements that contain the construction method indices for each activity. This solution structure is illustrated in Figure 5.5.

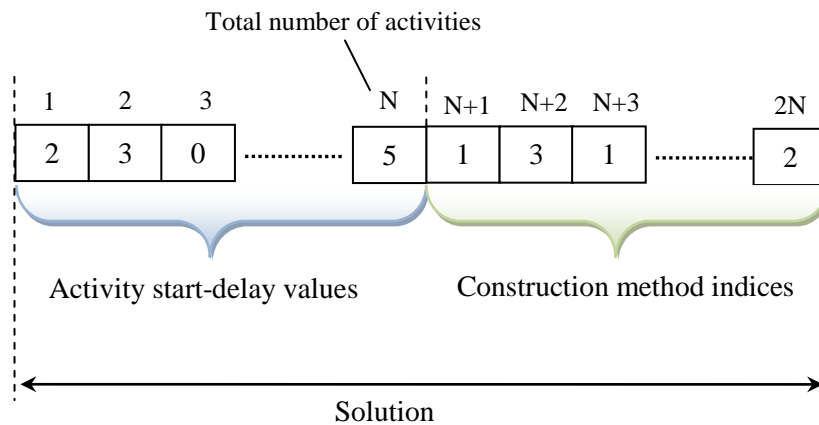


Figure 5.5: Solution structure for activity-level optimization

5.6.2 CPS Solution Structure (Time-Segment Level)

The CPS solution structure is made up of a string of values associated with the problem variables, as shown in Figure 5.6. The possible solution represented in Figure 5.6 is comprised of construction method indices and time-segment delay values. To determine the quality of a solution, its associated project duration and total cost are calculated and used as measures of the quality of the solution.

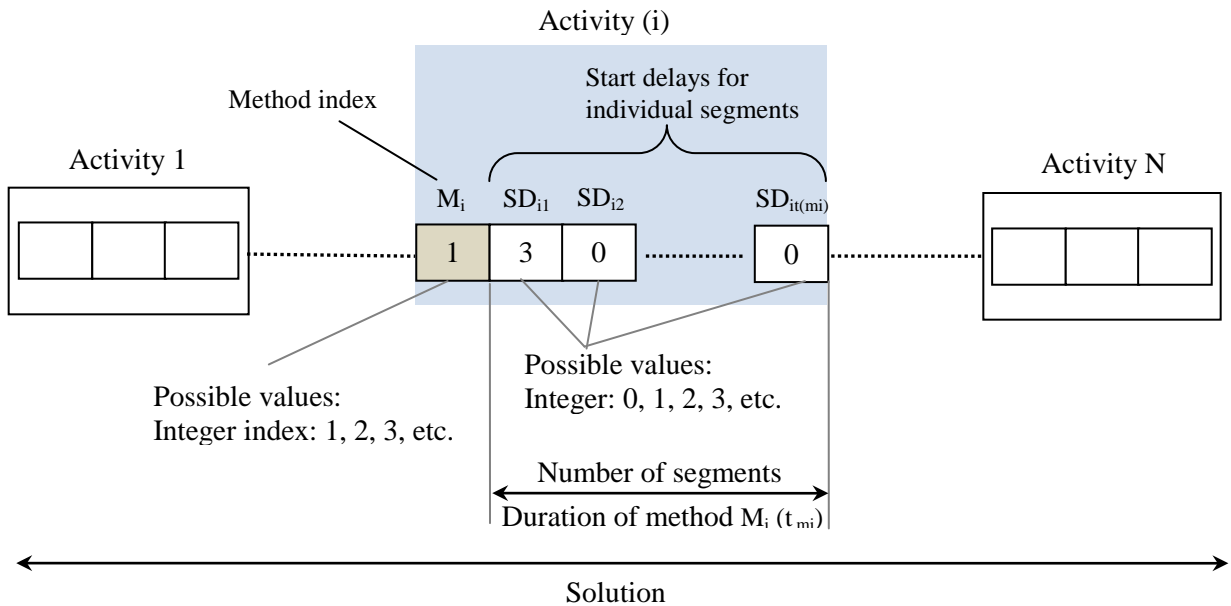


Figure 5.6: Solution structure

5.6.3 Generating an Initial Population

When the representation of the GA chromosomes has been determined, the first step in using GAs is to create an initial population. The initialization of the population is usually achieved by generating the required number of individuals using a random number generator that selects numbers that are uniformly distributed within the desired range.

5.6.4 Evaluation Criterion

The objective function is a measure of how individuals perform in the problem domain. For a minimization problem, individuals most often selected will have the smallest numerical value of the associated objective function (Equation 5.17). This measure is usually used only as an intermediate stage in determining the relative performance of individuals in a GA. Another

function, called the fitness function, is normally used to transform the objective function value into a measure of relative fitness.

5.7 Example of Schedule Optimization

Figure 5.7 illustrates a simple case study similar to the one reported in Son and Mattila (2004). Table 5.1 shows the activities and their estimates, while Figure 5.8 shows the general data for the project, including the start date; the working days; the key resources and their daily limits; the project deadline duration (15 days); and other contract provisions, such as a \$ 5000 per day penalty. As shown in Figure 5.8, three resources are required for this project: the equipment used for all activities and two types of labor. The resource limits are 6, 3, and 2 for resource one (R1), resource two (R2), and resource three (R3), respectively.

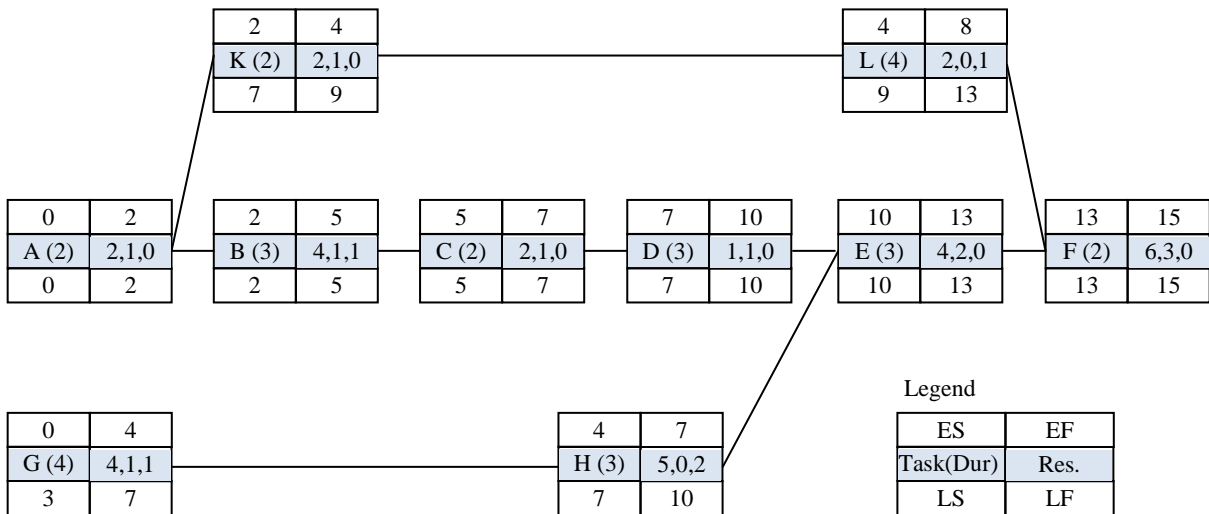


Figure 5.7: Project network of a case study example

Table 5.1: Activities data for the example

Activity	Estimate 1				Estimate 2			
	Duration	Cost	Interruptible	Interruption Cost	Duration	Cost	Interruptible	Interruption Cost
A	2	\$1,000	No	-				
B	3	\$1,500	No	-	2	\$2,100	No	-
C	2	\$1,000	No	-				
D	3	\$1,500	Yes	\$500				
E	3	\$1,500	No	-				
F	2	\$1,000	No	-				
G	4	\$2,000	Yes	\$700	3	\$2,800	No	-
H	3	\$1,500	Yes	\$500				
K	2	\$1,000	No	-				
L	4	\$2,000	Yes	\$700	3	\$3,200	No	-

Project Information			
Key Resources:			Start Date: 4-Jan-10
Code:	Limit:	Used:	Deadline (Days): 15.0
R1	6.0	11.0	Penalty (\$/d): 5,000
R2	3.0	3.0	Incentive (\$/d): 1,000
R3	2.0	4.0	Indirect (\$/d): 100
Workdays: SA <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> FR			Report Every (d): 2
No. of Activities: 10			i / Period (%): 1.00
Project End Date: 18-Jan-10			Markup (%): 10.00
Project Cost = \$15,500			Hold Back (%): 5.0
Duration (days) = 15.0			Down Payment (%):
			Suppliers credit (%):

Figure 5.8: General data for the sample project

Using the cheapest method (estimate 1) for each activity results in a project duration of 15 days and a total cost of \$15,500 (\$14,000 direct cost + \$1,500 indirect cost). Figure 5.9 shows the Gantt chart and the resource utilization profiles for the project schedule after CPM

is applied. The resource profiles show that both R1 and R3 are over-allocated for several days.

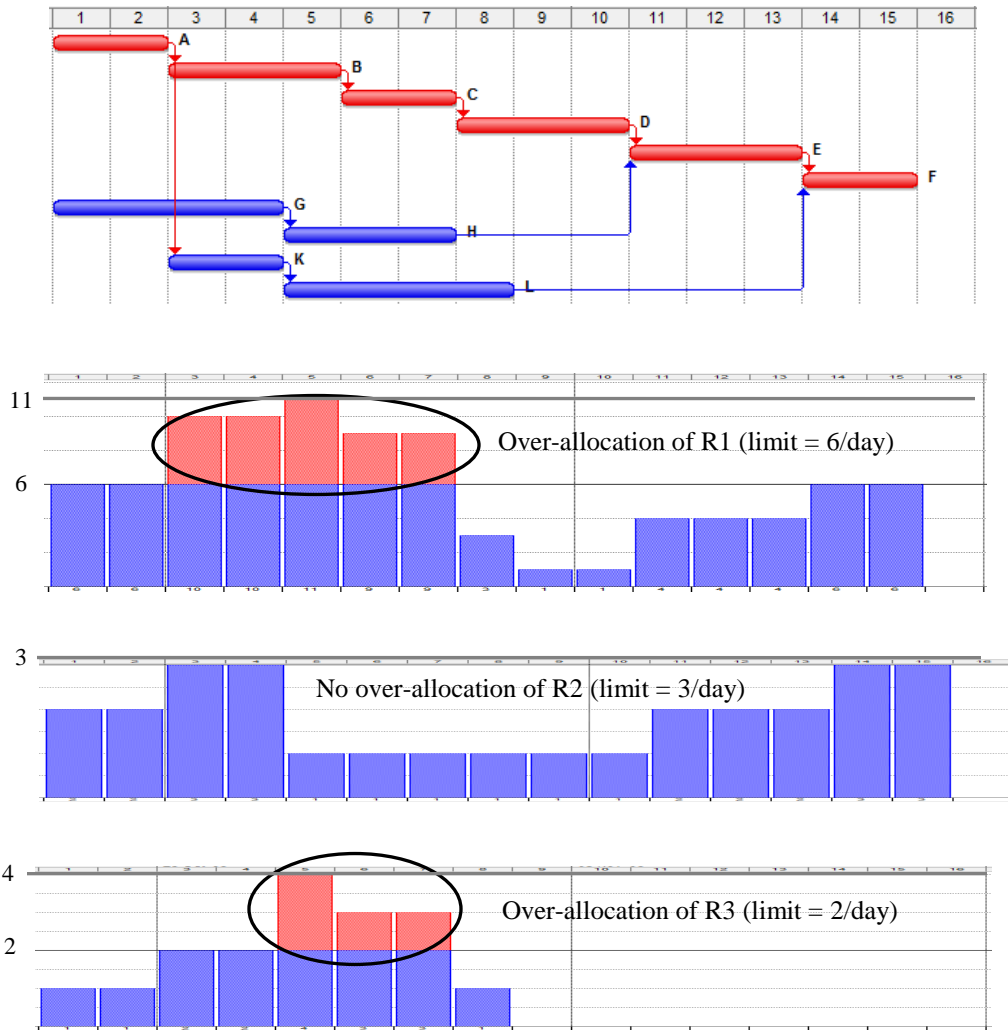


Figure 5.9: Schedule and resource utilization profiles for the case study

Using the resource-leveling feature in MS Project, the software attempted to modify the schedule in order to resolve the over-allocations. The resulting resource-loaded schedule,

shown in Figure 5.10, indicates that the project duration is extended from 15 days to 17 days. Accordingly, the total cost of the project is increased to \$25,700 (\$14,000 direct cost + \$1,700 indirect cost + \$10,000 delay penalty). It is important to note that the resource-leveling feature of MS Project includes an option to split activities that have not yet started. However, allowing MS Project leveling to create splits in the activities could not resolve the over-allocations.

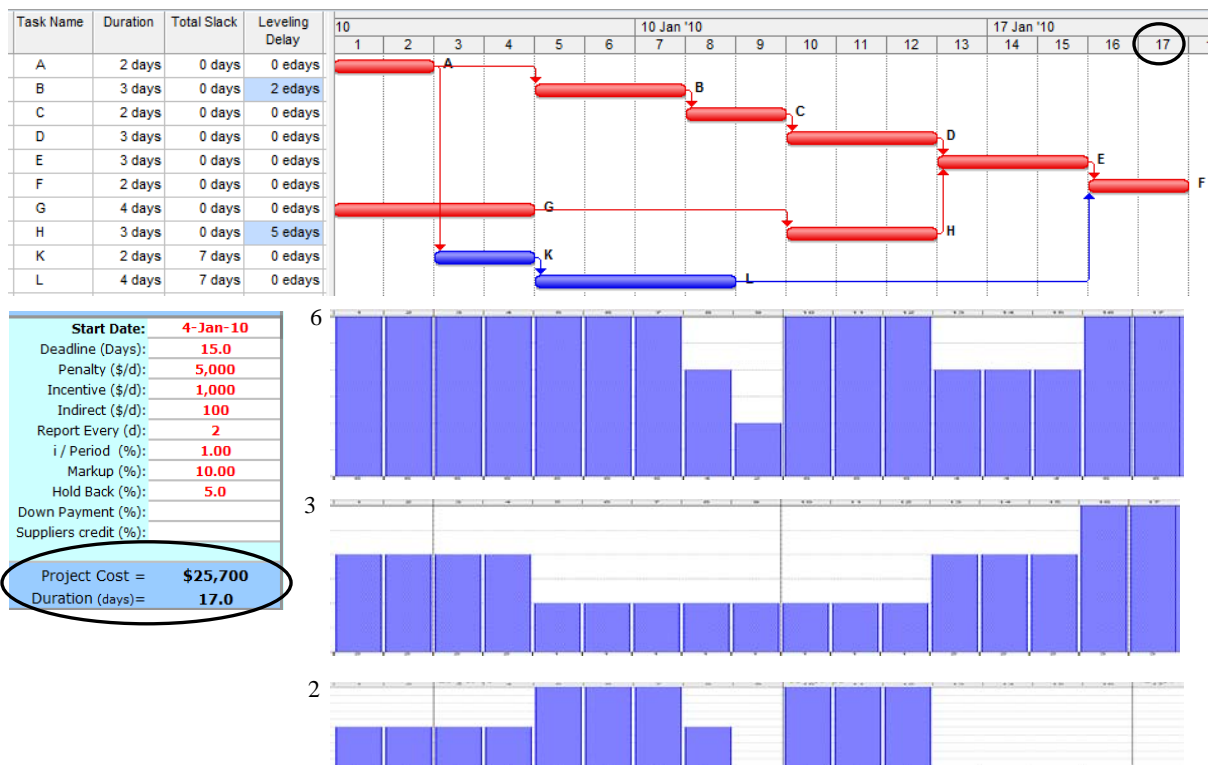


Figure 5.10: MS Project solution to the over-allocations of R1 and R3

Since MS Project does not have the capability of resolving the deadline constraint, the project schedule has been imported to a computer tool called EasyPlan (Hegazy, 2009), which integrates estimating, scheduling, resource management, and project control. The

EasyPlan program has been developed using the VBA language of Microsoft Excel software. Some of EasyPlan's features that facilitate project control and schedule optimization are as follows:

- It allows the user to specify up to three estimates (duration and cost) for each activity.
- It allows the user to enter up to three key resources and to specify the daily limit of these resources.
- It notifies the user if the resource limits are exceeded.
- It allows the user to change the method of executing any activity.
- It permits more than one baseline to be saved.
- It allows the user to enter the daily progress of an activity either as a percentage or as a delay caused by a certain party.
- It represents project progress using two bars for each activity: the top represents the baseline, and the bottom represents the progress. It thus shows whether the actual progress is faster or slower than that planned.
- It calculates and shows the actual project duration while the daily progress is being entered, with consideration of all delays, accelerations, and slowdowns.
- It allows the user to specify the project deadline and notifies the user if the project duration exceeds that deadline.

For this case study, once the second estimates for activities B, G, and L were added in EasyPlan to facilitate time-cost trade-off analysis, the optimization tool in EasyPlan were used to generate a schedule that satisfies the resource constraints and, at the same time, meets the deadline duration. Using the optimization tool in EasyPlan resulted in a schedule that has a 15-day project duration, which meets the deadline, and in which all activities are scheduled so that resource limits are not exceeded, as shown in Figure 5.11. Based on the resulting schedule, the total cost of the project is \$18,100 (\$16,600 direct cost + \$1,500 indirect cost).

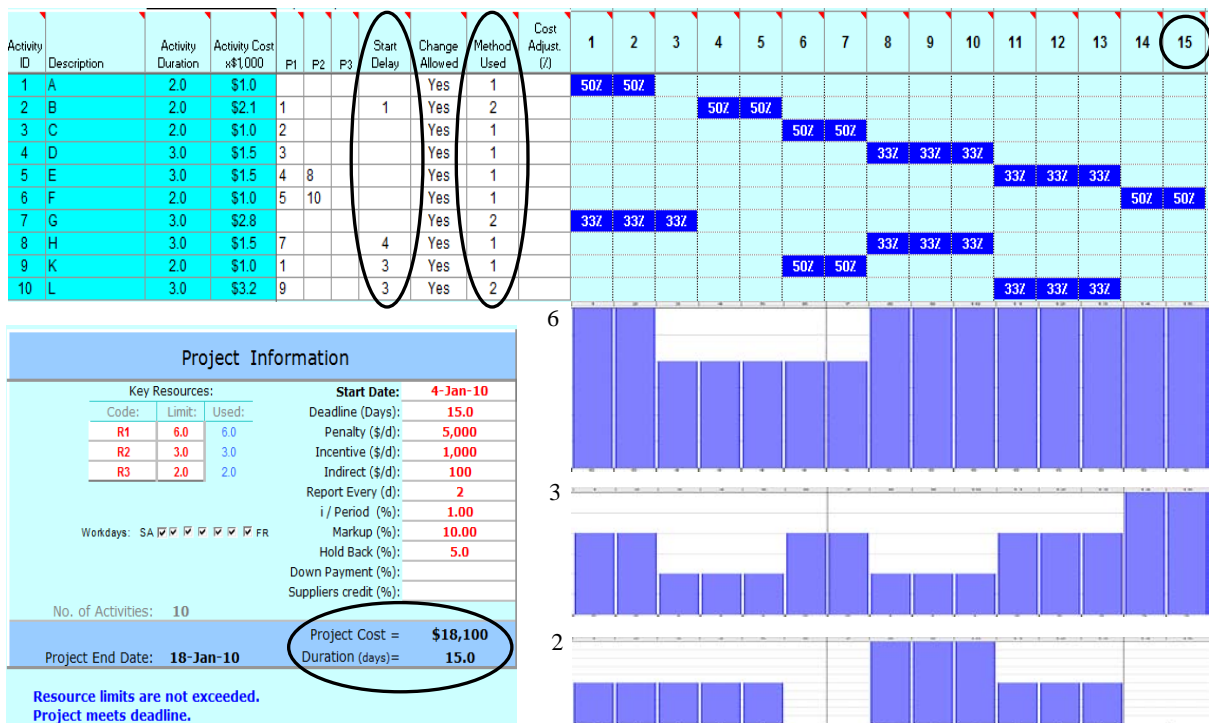


Figure 5.11: EasyPlan solution to resolve project constraints

Although EasyPlan was able to generate a schedule that satisfies both the resource and deadline constraints, the optimization process did not consider the option of splitting

activities while attempting to resolve over-allocations. For comparison purposes, the CPS representation was simulated on EasyPlan, with each time segment indicated as a separate activity with a one-day duration, as shown in Figure 5.12. The optimization procedure was applied to the simulated CPS schedule, the result was a 15-day schedule with a total cost of \$16,900 (\$14,000 direct cost + \$1,400 interruption cost + \$1,500 indirect cost), as shown in Figure 5.13.

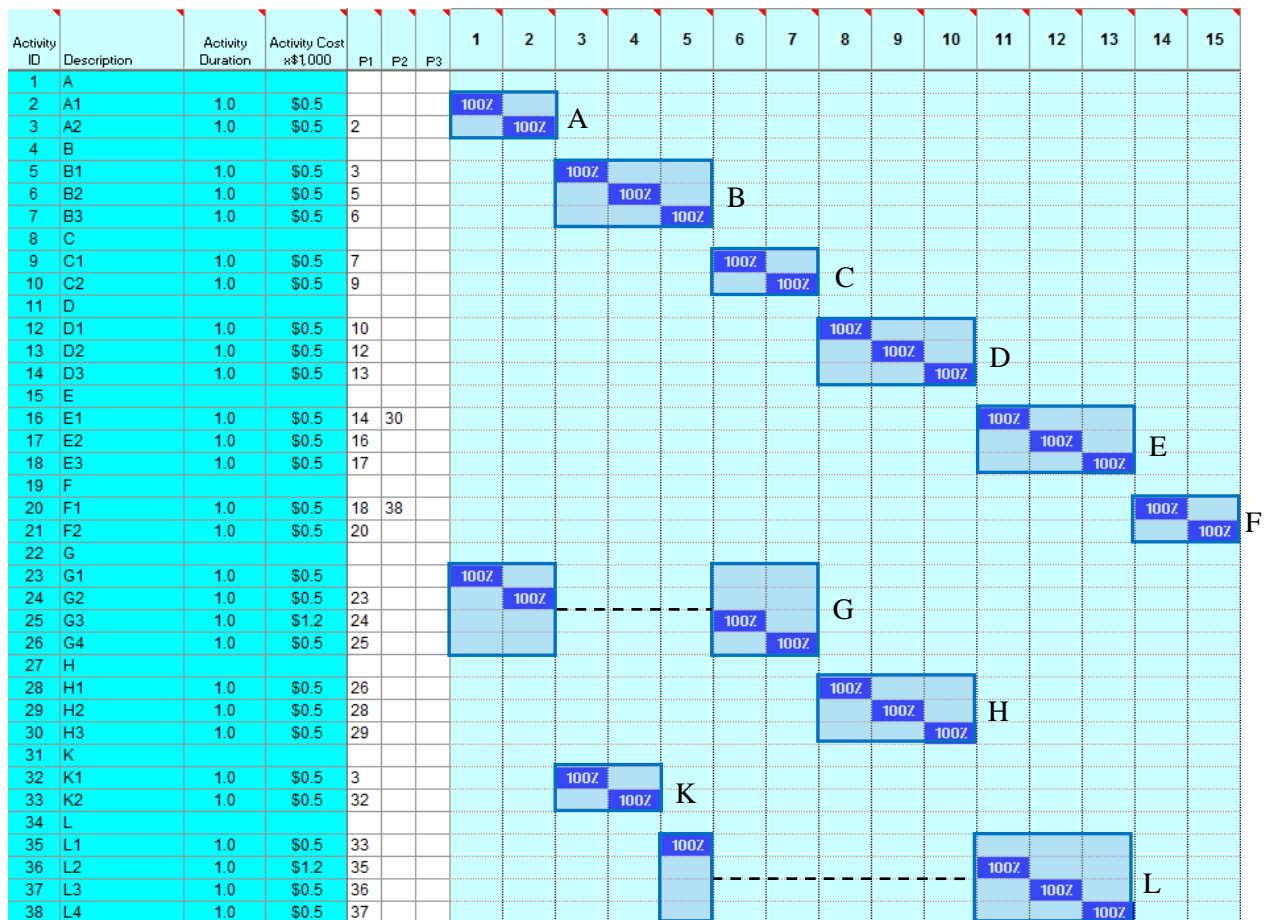


Figure 5.12: Simulated CPS schedule on EasyPlan

A start delay of 5 days added to time segment number 2 in activity L

A start delay of 3 days added to time segment number 3 in activity G

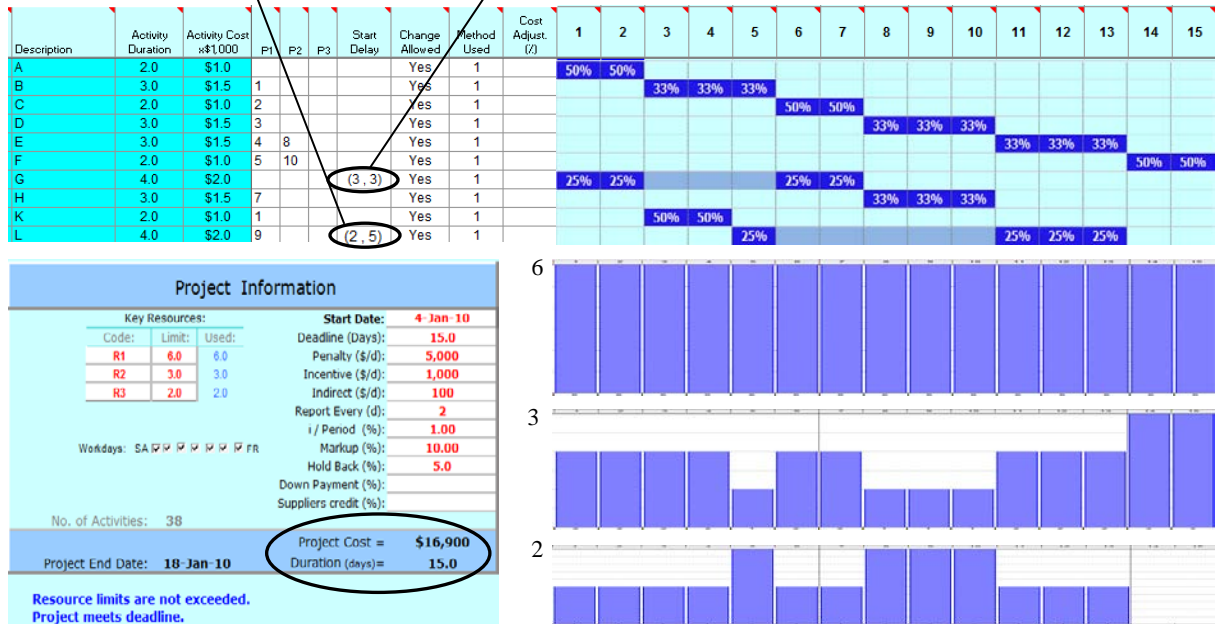


Figure 5.13: EasyPlan solution using simulated CPS schedule

5.8 Summary and Conclusions

This chapter has presented the CPS optimization formulation. A small case study has been presented in order to demonstrate a simple approach that can facilitate the resolution of multiple constraints at the activity level. The two main decisions required for this approach are: an index of the method selected that satisfies the deadline and start-delay values that insure the proper allocation of limited resources. These decisions represent key areas of focus during the scheduling and execution of a project. To suit the CPS representation, the same approach is reformulated so that the decisions are made at the time-segment level rather than at the activity level.

This chapter has also described the two common optimization procedures for project control, highlighting the respective objective function, variables, and constraints. The formulation of an optimization model for CPS has been presented and illustrated through an example. Evolutionary optimization has been described as an optimization technique that can be used effectively with CPS for schedule optimization.

Chapter 6

Validation: Case Studies

6.1 Introduction

This chapter presents two case studies that demonstrate the effectiveness of CPS for resolving constraints and for enhancing the efficiency of project control and delay analysis.

These two cases are described in the following sections.

6.2 Better Resolution of Constraints

In Case 1, the amount of a resource (L) has been entered for each activity, as shown to the right of the activity bars in Figure 6.1. As can be seen in the figure, the resource graph indicates that 13 L resources are needed for days 5 and 6, and 10 L resources are needed for days 3, 4, and 7, which are higher than the given resource limit of 6 per day.

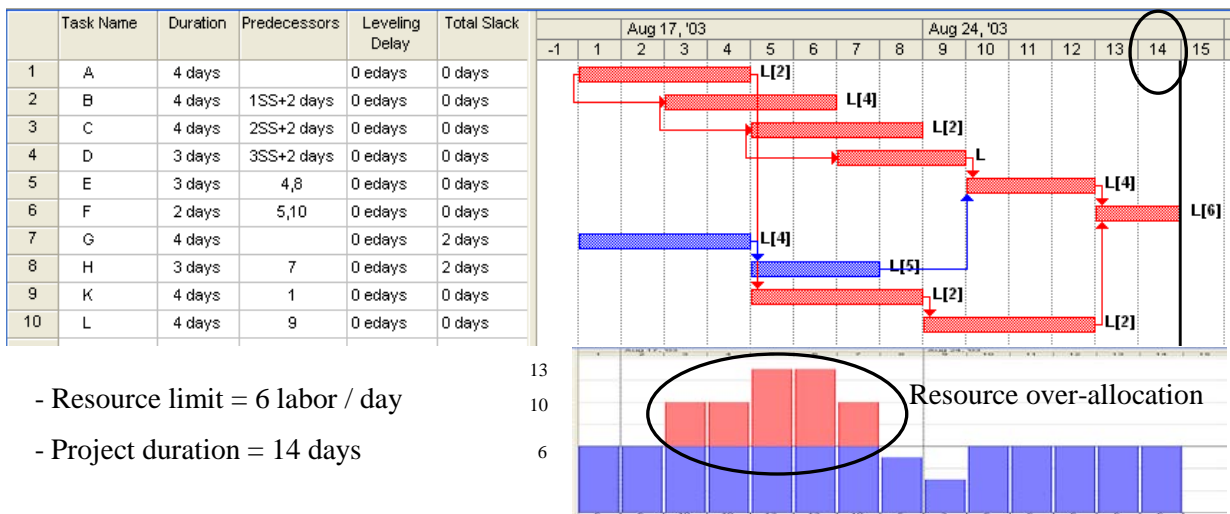


Figure 6.1: Traditional schedule and resource profile before resource leveling

Using the resource-leveling feature in MS Project, the software attempted to modify the schedule to resolve the over-allocation. The resulting resource-loaded schedule (Figure 6.2) indicates that the project duration has been extended from 14 days to 20 days. Allowing MS Project leveling to create splits in the activities results in a 19-day schedule, a saving of one day compared with the solution without splitting, as shown in Figure 6.3.

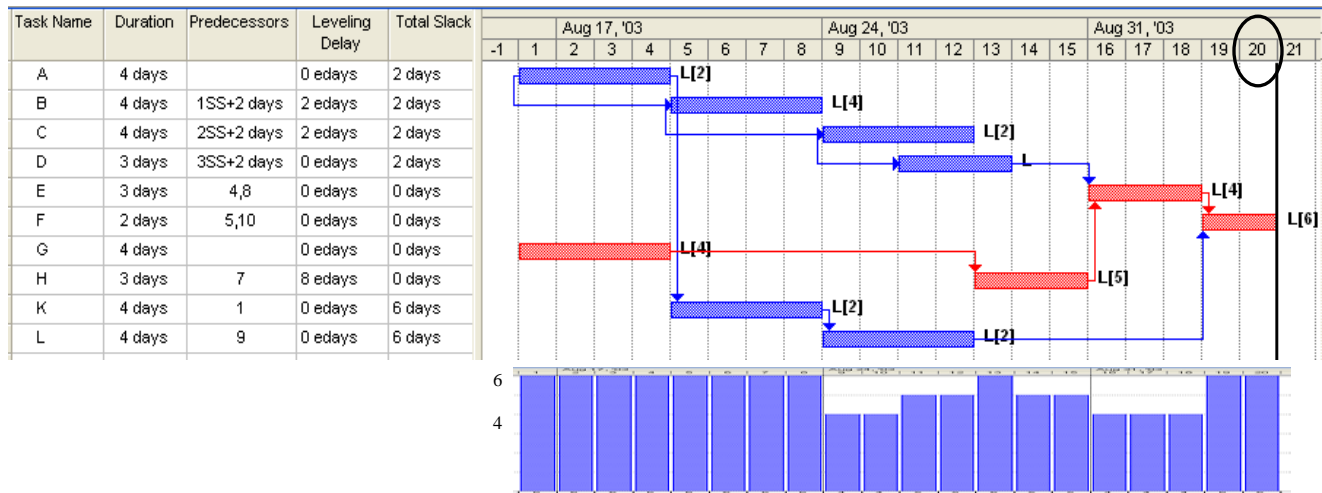


Figure 6.2: MS Project solution to resource over-allocation (without splitting)

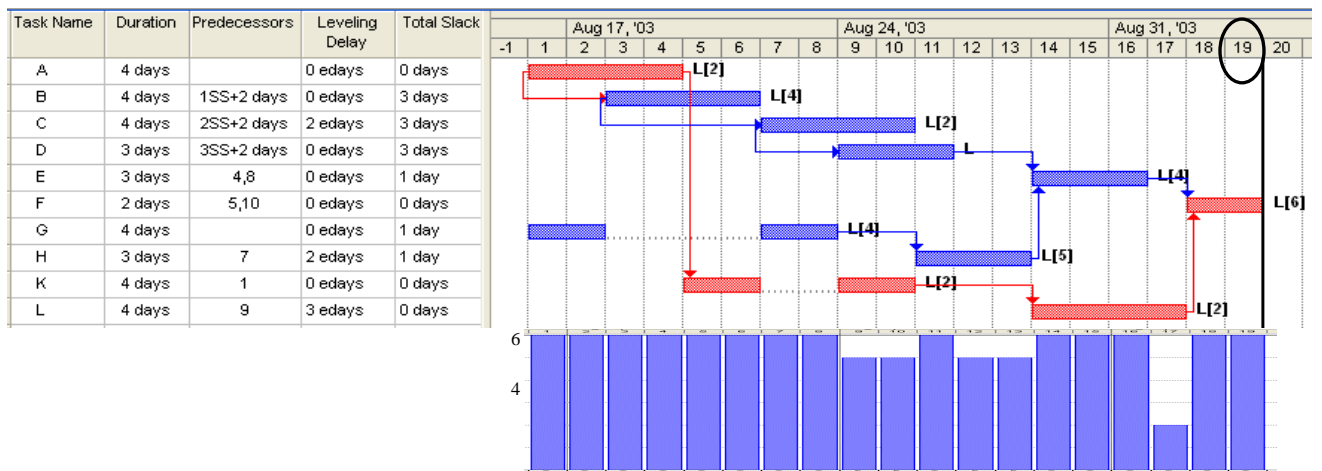


Figure 6.3: MS Project solution to resource over-allocation when splitting is allowed

To compare the traditional solution to resource over-allocation to the CPS method, an experiment was conducted using the simulated model of the CPS for the same case study. The result was a 19-day schedule, as shown in Figure 6.4. While the resource profile is smoother than that obtained with the traditional approach, the schedule shows a greater number of activity splits. This extra splitting can be minimized in CPS schedule optimization through the use of the interruption constraint (Eq. 5.21).

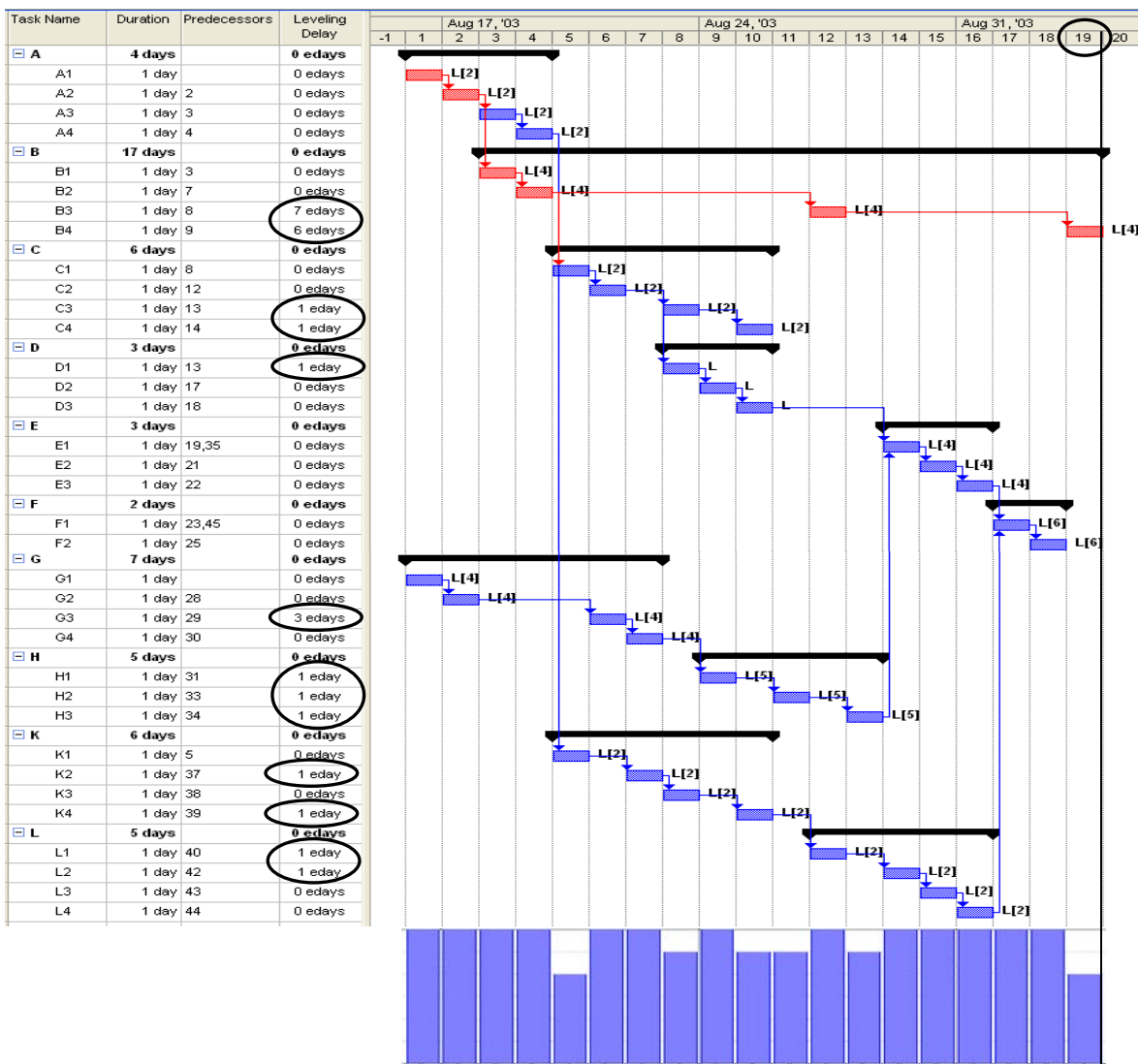


Figure 6.4: Project schedule and resource profile when activities are segmented

6.3 Efficient Project Control and Delay Analysis

Recent research has reported that delay analysis should consider the consequences of site events, caused by the different parties, with respect to resource over-allocations that, when resolved, cause project delays (Hegazy and Menesi 2008a, Ibbs and Nguyen 2007, Kim and de la Garza 2003, Kim 2009). As an example that demonstrates the accuracy of CPS in this regard, Figure 6.5 illustrates the as-planned schedule of a seven-activity case study. During the course of the actual work, the owner caused a delay in activity B on day 3. Although the delay did not affect the critical path, it rendered the initial resource allocation for the remaining work impractical. As shown in Figure 6.6, the resource would be over-allocated at day 7. Using the resource-leveling tool in MS Project resulted in a project duration of 14 days, as shown in Figure 6.7. This result indicates that the contractor would be forced to delay the project by one day. Accordingly, the contractor may claim that he/she is entitled to a one-day extension due to the resource over-allocation resulting from the delay caused by the owner.

When the CPS approach is used, it can be shown that resource leveling can result in a 13-day schedule. This method avoids inaccurate delay analysis that mistakenly assumes that the contractor is incapable of mitigating the resource over-allocation problem due to the owner-caused delay on day 3. As shown in Figure 6.8 a 13-day schedule can be achieved when activity D is stopped for one day and restarted again on day 8. As previously mentioned, although existing software includes an option for splitting activities during the resource-leveling process, it does not permit activities that have already been started to be split (Son and Mattila 2004).

Note: Each of activities B, C, and D requires one R1 resource/day, and the daily R1 limit = 2/day

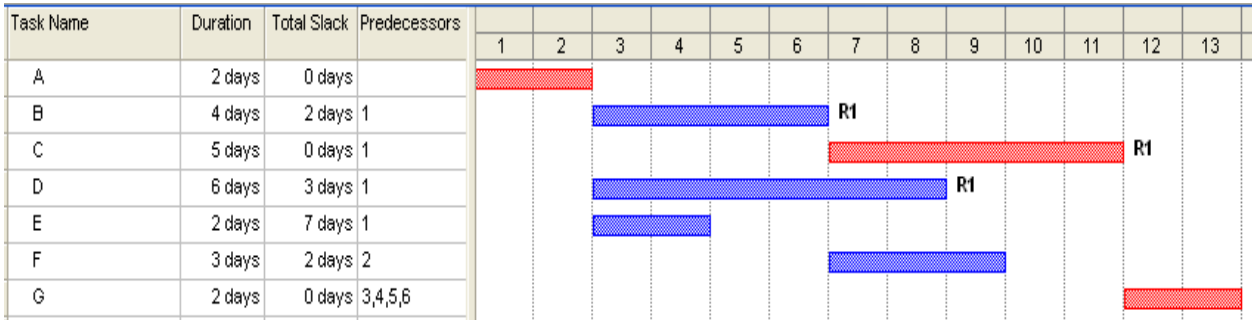


Figure 6.5: As-planned schedule for a case study (13 days)

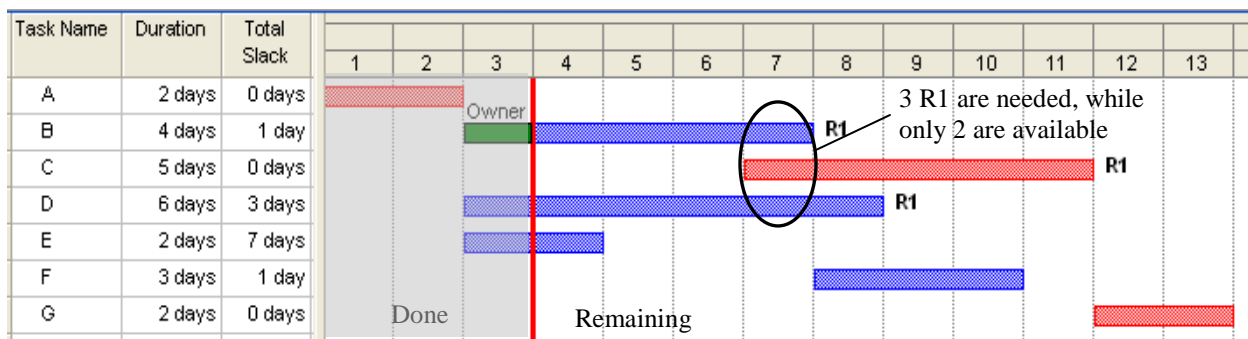


Figure 6.6: Simulated owner delay on day 3 causing resource over-allocation on day 7

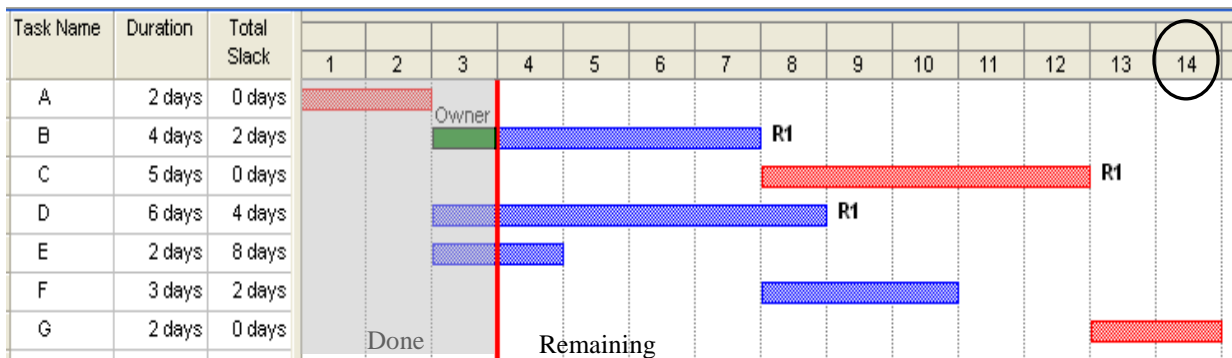


Figure 6.7: Resolving the resource over-allocation results in a one-day delay

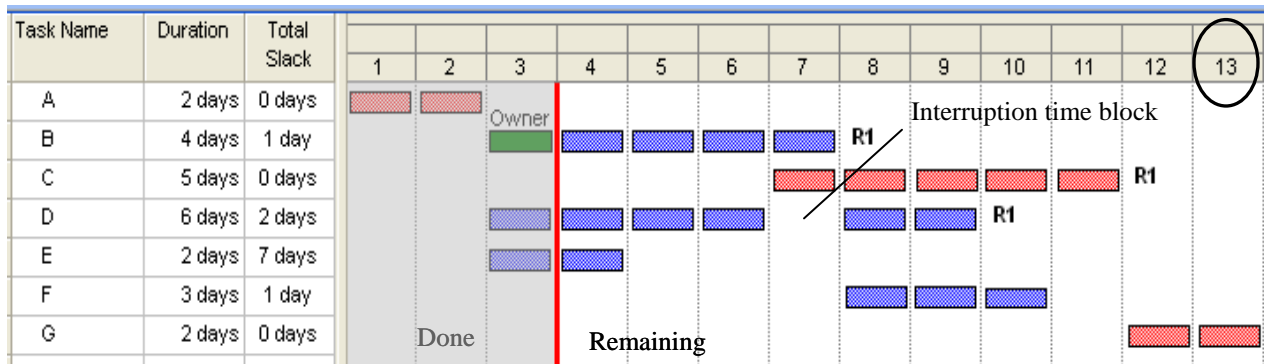


Figure 6.8: Resolving the resource over-allocation using CPS

6.4 Summary and Conclusions

In this chapter, two case studies have been presented in order to demonstrate the ability of CPS to achieve a better allocation of limited resources and to facilitate accurate analysis of project delays. For comparison purposes, the CPS representation has been simulated on Microsoft Project Software, with each time segment being simulated as a separate activity with a one-day duration. Based on the results of the two cases, CPS has proven to be a scheduling technique that offers more flexible resource leveling solutions, particularly for project updates, corrective action plans, and detailed schedule analysis.

Chapter 7

Conclusions and Future Research

7.1 Summary

Although the CPM method has been widely used for scheduling construction projects, it nevertheless fails to deliver structured decision support for projects. Despite the simplicity of CPM calculations, CPM schedules are difficult to analyze because of many well-documented factors, including the use of complex relationships, mid-activity critical-path fluctuation, and errors in float calculation. CPM also has no formulation that can account for multiple project constraints, such as deadline and resource limits. These factors have an impact on the accuracy and repeatability of CPM calculations and hinder the use of CPM as a decision support tool for corrective actions and delay analysis.

To overcome the drawbacks of CPM, this research has presented a Critical Path Segments (CPS) approach for micro-level critical path analysis. As opposed to the traditional representation of the duration of an activity as a continuous block of time that spans the entire activity duration, CPS represents each activity as a number of separate consecutive time segments that add up to the total duration of the activity. This approach permits the direct conversion of any complex logical relationship (SS, FF) into a simple Finish-to-Start (FS) relationship. Lag times that cause float calculation problems in the traditional CPM are not needed, mid-activity critical-path fluctuation can be determined, and multiple calendars can be more effectively analyzed. CPS also provides the ability to define the relationship between activities not only as time-based but also as production-based. When daily percentages are reported on the time segments, it is possible to clearly convey information

related to the speed of the construction (e.g., actual versus planned) and to show the use of different resource calendars on different days or for different activities. These features, in essence, offer a more granular level of analysis that uses a time segment -by- time segment analysis, rather than the traditional activity-by-activity analysis.

In CPS, all progress data are represented on a time-segment basis, and all events caused by various parties are incorporated. Schedule analysis can therefore be carried out accurately with less disagreement among the parties. In the developed representation, work progress expressed in percentage is shown on the associated time segments. The progress percentages do not mandate daily input but, rather, can be averaged over a number of days or weeks. Additional time segments can also be inserted in order to represent known events that occurred on specific dates and that were caused by the owner “O”, the contractor “C”, and/or another party “N” (e.g., the weather). Such a generic activity representation clearly shows the evolution of all activity events, including the effect of decisions such as acceleration and resource allocation. This representation is therefore general enough to facilitate the calculation of the remaining duration and floats, as well as to allow for corrective actions and schedule analysis.

To facilitate the resolution of deadline and resource constraints, CPS permits the representation of each activity to include optional cost estimates that correspond to activity variables that offer a wide range of possible solutions to project constraints. It should be noted that the optional estimates represent practical options that vary from inexpensive and slow to fast and expensive. Given the sequence of the activities and the variety of

construction options, it is possible to arrive at a least-expensive plan that meets both the deadline and the resource limit. Two quantitative decisions must be made: an index for the method that provides a good trade-off between the duration and cost of the activity (i.e., time-cost tradeoff analysis) and the start-delay time that is applied to the start of each time segment to be used for the resolution of resource over-allocations. The formulation of the key activity variables (decisions) presented in Chapter 5 therefore simplifies their straightforward incorporation into the mathematics of the CPS algorithm, not only for scheduling before construction but also for corrective actions during construction. If a project is delayed, for example, then a suitable corrective action is to decide on modified values for the two decisions, which will affect the remaining portion of the schedule.

7.2 Discussion of the CPS Approach

Based on the results of the cases presented in the previous chapters, CPS has proven to be a good basis for improving the scheduling process. One of the benefits of using CPS in background computations of a schedule is the fact that it offers few changes to the manner in which scheduling basics are taught. Segmenting activity duration (simply into days) also adds the necessary level of detail that is consistent with the findings of other research. Hegazy and Menesi (2008), for example, proved that daily analysis is needed in order to produce accurate delay analysis. Al-Gahtani (2009) also presented an approach for dividing the float and allocating it on a daily basis among the project parties according to the levels of risk they assume. He then proposed a day-to-day system for monitoring the dynamics of float management.

7.2.1 Issues Related to Planning

Several studies in the literature have discussed the granularity of construction activities from the prospective of process or product (e.g., Song and Chua 2007, Chua and Shen 2005, Morad and Beliveau 1994, Winstanley et al. 1993, Hendrickson et al. 1987). These important studies have been focusing mainly on planning and have been used to generate generic baseline schedules by the derivation of precedence relationships from product models. Moreover, research on the concepts of lean construction such as the short-interval planning that goes into crew-level details has proven to be tremendously effective in improving productivity (Kim 2002). According to Kim, even limited implementation of short-interval planning (which the CPS is capable of providing) can be far more effective than the typical planning efforts currently employed in the construction industry. It is important to note that the CPS technique is a detailed scheduling technique that is most advantageous for documenting and analyzing as-built schedules. The CPS technique can therefore work well in collaboration with these efforts in order to provide the lower level of granularity that allows efficient project control through better resource management, delay analysis, corrective actions, and recording of site events. The difference between CPM and CPS is thus similar to the difference between design drawings and the more detailed shop drawings that are used during construction.

Segmenting activity durations may seem to add a greater computational burden to the critical path analysis. However, this impression is false because all the relationships are simplified in CPS. In addition, CPS uses a new scheduling process that includes only a forward pass

calculation, thus saving almost 50% of the processing time because backward pass calculation is eliminated.

7.2.2 Issues Related to Project Control

- The deadline constraint is considered in the optimization model as a soft constraint because there might be no feasible schedule that satisfies the deadline constraint, given the sequence of the activities and the availability of resources. However, in case the project has a very restricted deadline, the deadline constraint can be considered as a hard constraint and the schedulability of the project should be investigated.
- In order to accommodate the growing data requirements for project control and schedule analysis, detailed as-built information need to be recorded more frequently. The use of Personal Digital Assistants (PDAs) could be beneficial in recording site events on the schedule.
- To avoid dealing with potential uncertainties during construction (e.g., absenteeism, late deliveries of materials, and weather conditions), tools that are available for risk management and that are used with CPM, such as the program evaluation and review technique (PERT) and Monte Carlo simulations, can be also used with CPS.
- For large scale construction projects, CPS has the potential to be used in low-level planning stage where detailed information of segments as well as their relationships is better identified.

7.3 Contributions

CPS has the potential to revolutionize the way schedules are generated and managed. Based on the current development, this research makes a number of contributions:

- **Better understanding of construction scheduling needs:** This study has provided an in-depth review of the research reported in literature with respect to the theoretical and practical drawbacks of CPM. Based on these drawbacks, areas in CPM scheduling procedures that require enhancement have been identified.
- **Critical Path Segments (CPS) approach:** This research has resulted in the development of a new scheduling technique with a finer level of granularity. This research has introduced a new network representation, activity representation, and detailed mathematical formulation for CPS.
- **Elimination of complex relationships in the schedule:** The CPS representation of project networks permits the direct conversion of any complex logical relationship (SS, FF) into a simple FS relationship without the lag times that cause float calculation problems in traditional CPM.
- **Better representation of the intention of activity relationships:** CPS provides more flexible options to better represent the intent of the relationships between the activities. CPS can define the relationship between activities not only as time-based but also as production-based.
- **Clear representation of activity progress:** Reporting the daily percentages on the time segments enables a clear indication of the information related to the speed of

construction (e.g., actual versus planned) and to the use of different resource calendars on different days or for different activities. In addition, schedule analysis can be carried out accurately with less disagreement among parties.

- **Accurate schedule calculations:** CPS provides more reliable and realistic scheduling data than CPM because CPS considers resource availability in the scheduling process, accurately calculates activity floats, and accurately identifies the critical path.
- **Mechanism to incorporate project constraints into the schedule:** CPS provides a flexible formulation with varying levels of resolution for the schedule so that a better range of possible solutions to the practical constraints can be generated. The CPS formulation results in more practical and realistic schedules with respect to resource allocation. CPS enables the stopping and restarting of activities through the use of start delay values for all individual time segments, as necessary, so that the limited resources are not exceeded.
- **Accurate schedule analysis during and after execution:** CPS provides micro-level critical path analysis, which is particularly suited to progress documentation, as-built schedule analysis, and corrective action optimization. It facilitates accurate and reliable schedule analysis because the complexity of schedules associated with the use of different relationships and lags is avoided and the representation of activities is improved.

7.4 Future Research

Several aspects of CPS could be improved through further research. The following areas are recommended for further study in order to enhance the capability of the current CPS scheduling technique so that it will be more practical for real construction projects:

- Many researchers have criticized the CPM because of its inability to model repetitive projects. Generalizing the CPS formulation to cover repetitive scheduling is an area for future research: a single project could be considered as a special case of multiple and repetitive projects.
- One of the criticisms of CPM is its inability to handle iterations, which discrete event simulation tools can handle (Fayez et al. 2003, Sawhney et al. 2003). Therefore, investigating the integration of CPS and simulation is a potential future extension. This can be done by introducing variable time segments for different activities. For example, activities that have a cyclic nature, such as earth-moving operations, can be modeled using a time segment of one second or one minute, and a simulation model of this operation could thus be integrated within the CPS network.
- Features related to financing decisions and project control could be incorporated into CPS, such as cash flow analysis, earned value, cost and schedule performance indices, and productivity analysis.
- Because a practical schedule should handle project constraints such as deadlines and resource availability simultaneously, further investigation could examine the

possibility of combining resource allocation techniques with time-cost trade-off analysis in a unified technique that is integrated into CPS.

- Different optimization setups and techniques could be experimented with in order to find the technique most suitable for CPS. Different optimization techniques, which include heuristic and evolutionary techniques, such as simulated annealing, genetic algorithms, ant colony optimization, particle swarm optimization, and shuffled frog leaping, could be used with CPS and compared with respect to results and processing time.
- A prototype could be developed for CPS so that all the developments are integrated in a user-friendly automated environment. It is also possible to link such a CPS prototype to commercial estimating software in order to determine the relevant costs of schedule delays and accelerations.
- The application of the CPS scheduling technique to real-life projects is essential future work in order to validate the practicality of the CPS technique in practice.

Appendix A

Evolutionary Algorithms

Evolutionary Algorithms (EAs) are stochastic search methods that mimic the metaphor of natural biological evolution and/or the social behavior of species. The behavior of such species is guided by learning, adaptation, and evolution (Lovbjerg, 2002). To mimic the efficient behavior of these species, various researchers have developed computational systems that seek faster and more robust solutions to solve complex optimization problems. The first evolutionary-based technique introduced in the literature, was the genetic algorithms, (Holland, 1975; Goldberg, 1989). In an attempt to reduce processing time and improve the quality of solutions, particularly to avoid local optima, other EAs have been introduced during the past 10 years, including various GA improvements and recently developed techniques: shuffled frog leaping (SFL), particle swarm optimization (PSO), and ant colony optimization (ACO).

In general, EAs share a common approach for their application to a given problem. The problem usually requires some representation to suit each method, then, the evolutionary search algorithm is applied iteratively to arrive at optimum or near-optimum solution. Elbeltagi et al. (2005) compared the performance of five evolutionary algorithms for solving general optimization problems and reported the powerful performance of Genetic Algorithms and the Shuffled Frog Leaping (SFL) techniques. A brief description of Genetic algorithms is presented in the following section.

A.1 Genetic Algorithms (GAs)

Genetic Algorithms were developed to mimic some of the processes observed in natural evolution; they employ a random yet directed search for locating optimal solution. John Holland (1975), from the University of Michigan began his work on genetic algorithms at the beginning of 60s; the first publication of his work was on 1975. The basic techniques of the GAs follow the principles first laid down by Charles Darwin of "survival of the fittest", since in natural competition among individuals for resources results in the fittest individuals dominating over the weaker ones (Forrest 1993). GA is a stochastic random optimization method for solving large scale problems. GAs differ from normal optimization techniques in several ways. First, the algorithm works for a population of strings, searching many peaks in parallel. By employing genetic operators, it exchanges information between the peaks, thus lessening the possibility of ending at a local minimum and missing the global minimum. Second, the algorithm needs to evaluate only the fitness function to guide its search and not the derivatives or other auxiliary knowledge.

To start solving any problem, a coding scheme is formulated to encode the problem parameters. Usually this is done in the form of a string called chromosome (or gene) as presented in Figure A.1. This coding representation is dependent on the problem and not unique. The genes are generated in a random fashion, i.e. the values of the parameters that are coded in the genes are random values and each gene represents one solution that is better or worse for the problem. The construction of a GA for any problem is classified into the following tasks: (1) determination of chromosome representation; (2) determination of fitness function; (3) determination of population size and number of generations; and (4)

determination of genetic operators (Chan and Tansri, 1994). Figure A.2 shows the basic steps of performing GAs algorithms (Lin and Lee 1996).

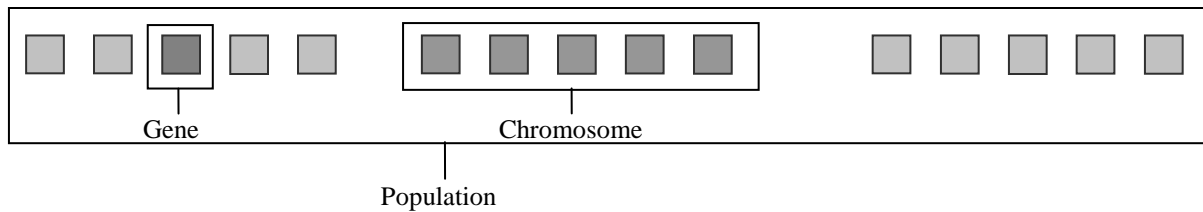


Figure A. 1: Population, gene and chromosome in GA

After defining the population, an objective function (fitness function) should be well defined for the problem. The fitness value of each string is computed from the fitness function. A good string is the one that scores a high fitness value. The size of the population is problem dependent and needs to be determined experimentally. Population size affects the quality of the end solution, as well as, the processing time it consumes. On the basis of the quality of a gene, the gene is assigned a fitness value. The solution will converge to near optimal solution after a certain number of generations (Chan and Tansri, 1994). The process continues for a large number of generations. Among all the possible solutions, the good solutions are selected, while the others are eliminated to simulate the process of “Survival of the fittest”. The selected solutions undergo the processes of reproduction, crossover, and mutation to create new generations of possible solutions. The new set of generations are expected to perform better than the previous ones, they will be evaluated and assigned a new fitness value. The process continues until convergence is achieved within the population (Ross 1995).

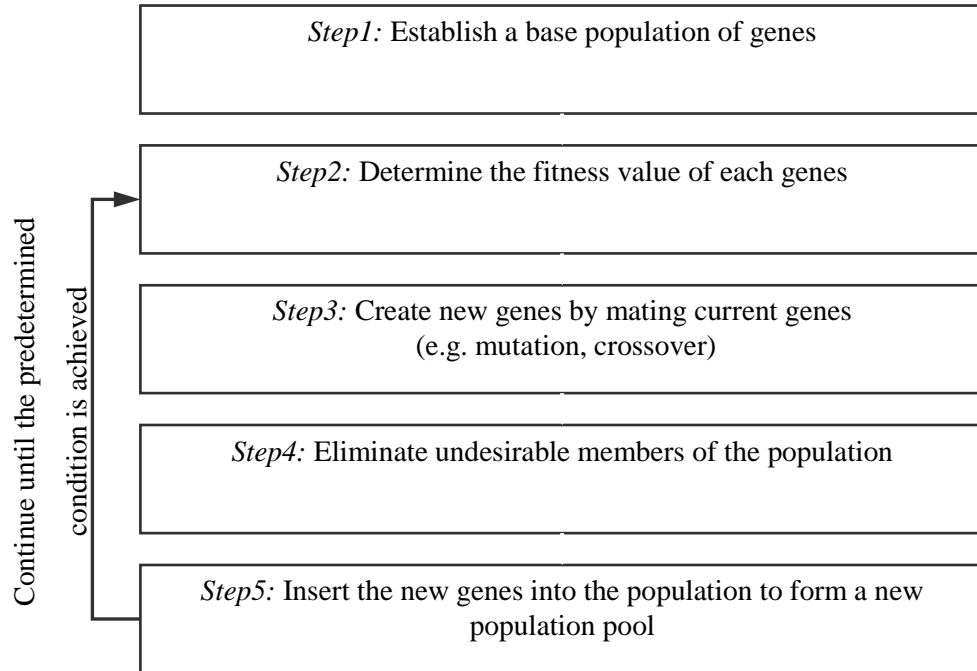


Figure A.2: Genetic algorithm process (Lin and Lee 1996)

A.1.1 Fitness Normalization

Fitness normalization is the process of converting row fitness value to one that behaves better. It gives high probabilities for selecting good solutions in new generation, while maintaining some chances of survival to poor solutions (Boesel et al. 1999). Fitness normalization can be carried out in three forms: (1) inversion normalization, (2) linear ranking, and (3) non-linear ranking. The inversion normalization is considered the popular method in normalizing the fitness; it is calculated as follows:

$$F = \sum_{i=1}^n fitness(g_i) \tag{A.1}$$

$$F_i = \frac{fitness(g_i)}{F} \tag{A.2}$$

A.1.2 Selection

The selection process is conducted by one of the following techniques: roulette-wheel parent technique or tournament selection. The roulette-wheel technique starts with generating a random number (m) between 0 and the total fitness (F). Then return the first population whose fitness, added to the fitness of the preceding population members (running total) is greater than or equal to m (Lin and Lee 1996). The wider span (best fit) for a chromosome, the higher the chance it will be selected. Figure A.3 shows a weighted roulette wheel for a population of 6 chromosomes. From Figure A.3, it can be noticed that chromosomes 2 and 5 are the fittest chromosomes and have higher probability over the rest of the population to be selected for further reproduction.

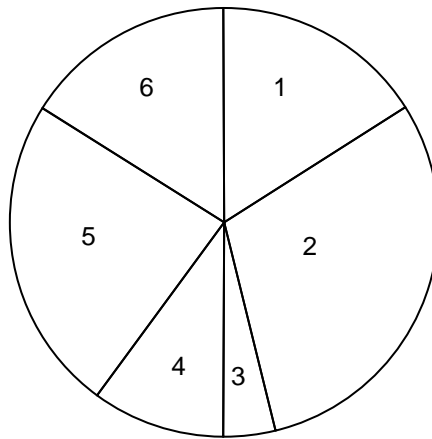


Figure A.3: Weighted roulette wheel (Lin and Lee 1996)

In the tournament selection, a number of chromosomes are chosen randomly from the population; the best fit chromosome is then selected and passed to the new generation (Goldberg and Deb 1991). Tournaments are performed for a tournament of size “ S ” which represents the number of competing chromosomes in the tournament. Usually, tournaments

consist of two chromosomes ($S=2$). The selection of the superior chromosome within a tournament is performed based on actual fitness values (without performing fitness normalization).

A.1.3 Crossover

Crossover is the process by which the chromosomes are able to mix and exchange their desirable qualities in a random fashion; it is considered the most important operator in the genetic algorithm (Lane 1993). Crossover (marriage) is conducted by selecting two parent chromosomes, exchanging their information, and producing an offspring. The two parent genes are selected randomly in a manner such that the probability of being selected is proportional to its relative fitness. This ensures that better chromosomes being selected in the process without violating the randomness. A random number is generated and compared to user-specified threshold value for crossover (P_c). The higher the crossover, the more quickly new structures are introduced to the population. The crossover proceeds in a simple way, for each couple of strings two random numbers are selected between [1 and $m-1$], where m is the chromosome length. The information between the two selected chromosomes is exchanged as shown in Figure A.4. This method is called “discrete crossover”. Another method is called “arithmetic crossover”, where an interpolation of genes values is performed in order to ensure that genes contents receive new values in the new generations (Kim and Adeli 2001).

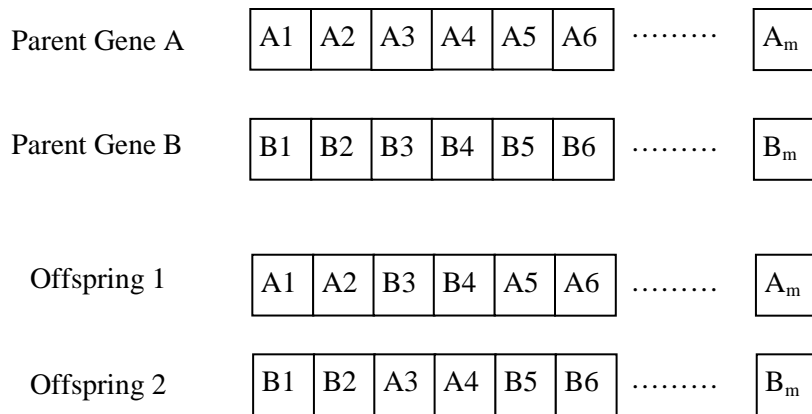


Figure A.4: Crossover operator to generate offspring genes

A.1.4 Mutation

Mutation is a rare process that resembles the process of a sudden generation of an offspring that turns out to be a genius (Goldberg 1989). During the creation of a generation, it is possible that the entire population of strings is missing vital information that is important for determining the correct or the optimal solution. Future generations and crossover might not be able to arrive at this missing gene, sometimes the population is stagnated. The mutation process is capable of changing the properties of the gene, thus insures the introducing of the missing information. For each chromosome gene, a random number is generated and compared against the user-specified threshold value for mutation (P_m). Usually mutation is rare in nature, which is an order of once in one hundreds.

A.1.5 Elitism

Elitism is the process to overcome the problem of losing the best chromosome in each population due to the random nature employed in selection and the effect of crossover and mutation. In elitism, the chromosome with the best fitness in each population is retrieved and used to replace the least fit chromosome in new generation.

Many efforts had been carried out in the development and application of Genetic Algorithm (GA) in civil engineering. It shows to be efficient in solving the site-layout optimization of facilities (Elbeltagi and Hegazy 2001, Cheung et al. 2002, Li and Peter 2000, and Osama et al. 2003). Also, in solving the cost optimization and cost trade off problems, Hegazy (1999a) presented an optimization genetic algorithm for minimizing the construction costs. GAs were implemented in solving the problem of resource scheduling and leveling in construction projects (Hegazy 1999b). The common conclusion among all the previous researches was the efficiency of implementing GA in solving complex problems and arriving at a near optimal solution in small time.

References

- Ackley, A., Baker, J., and Lowe, S. (2007). "Top ten CPM scheduling mistakes." *Rail Conference 2007 Proceedings*, June, 2007, Toronto, On.
- Ahuja, H., Dozzi, S. P., and AbouRizk, S. M. (1994). *Project management techniques in planning and controlling construction projects*, 2nd Ed., Wiley, New York.
- Ahuja, V., and Thiruvengadam, V. (2004). "Project scheduling and monitoring: current research status." *Construction Innovation*, 4(1), 19-31.
- Al-Gahtani, K. (2009). "Float Allocation Using the Total Risk Approach," *Journal of Construction Engineering and Management*, ASCE, 135 (2), 88–95.
- Arditi, D., and Pattanakitchamroon, T. (2006). "Selecting a delay analysis method in resolving construction claims." *International Journal of Project Management*, 24(2), 145-155.
- Basu, A. (2008). "Business rules in CPM scheduling software." *AACE International Transactions*, PS.01.1-PS.01.9.
- Baweja, S. S. (2006). "CPM schedules - Why and how." *AACE International Transactions*, PS.22.1-PS.22.5.
- Boesel, J., Nelson, B., and Ishii, N. (1999). *A Framework for Simulation-Optimization Software*. Technical Report, Department of Industrial Engineering and Management Science, North Western University.
- Bowers, J. A. (1995). "Criticality in resource-constrained networks." *Journal of the Operational Research Society*, 46(1), 80–91.
- Buddhakulsomsiri, J., and Kim, D.S. (2007). "Priority rule-based heuristic for multi-mode resource-constrained project scheduling problems with resource vacations and activity splitting." *European Journal of Operational Research*, 178, 374-390.
- Chan, K., and Tansri, H. (1994). "A study of genetic crossover Operations on the facilities layout Problem." *Computers and Industrial Engineering*, 26 (3), pp. 537-550.
- Chassiakos, A. P., and Sakellaropoulos, S. P. (2005). "Time-cost optimization of construction projects with generalized activity constraints." *Journal of Construction Engineering and Management*, ASCE, 131(10), 1115-1124.

- Chassiakos, A. P., Samaras, C. I., and Theodorakopoulos, D. D. (2000). "An integer programming method for CPM time-cost analysis." *Computer Modeling in Engineering and Sciences*, 1(4), 9-18.
- Cheung, S., Thomas K., and Tam, C., (2002), "Site Pre-Cast Yard Layout Arrangement Through Genetic Algorithms." *Automation in Construction*, 11 (1), January, pp. 35-46.
- Chua, D. K. H. and Shen, L. J. (2005) "Key Constraint Analysis with Integrated Production Scheduler." *Journal of Construction Engineering and Management*, ASCE, 131(7), 753-764.
- Davis, 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.
- Elbeltagi, E. and Hegazy, H. (2001), "A Hybrid-Based System for Site Layout Planning in Construction." *Computer-Aided Civil and Infrastructure Engineering*, 16 (2), pp. 79-93.
- Elbeltagi, E, Hegazy, T., and Grierson, D. (2005). "Comparison among five evolutionary-based optimization algorithms." *Advanced Engineering Informatics*, 19(1), 43-53.
- Eshtehardian, E., Afshr, A., and Abbasnia, R. (2008). "Time-cost optimization: using GA and fuzzy sets theory for uncertainties in cost." *Construction Management and Economics*, 26, 679-691.
- Fayez, M., Axelsson, P., Oloufa, A., and Hosni, Y. (2003). "DSM Versus CPM: Issues for Planning Design & Construction Activities," *Proceedings of the Construction Research Congress held in Honolulu, Hawaii, March 19-21, ASCE*.
- Feng, C. W., Liu, L., and Burns, S. A. (1997). "Using genetic algorithms to solve construction time-cost trade-off problems." *Journal of Computing in Civil Engineering*, ASCE, 11(3), 184-189.
- Finke, M. (1999). "Window analysis of compensable delays." *Journal of Construction Engineering and Management*, ASCE, 125(2), 96-100.
- 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.
- Forrest, S. (1993), "Genetic Algorithms: Principles of natural selection applied to computation", *Science*, Vol. 261, 872-878.

- Galloway, P. D. (2006). "Survey of the construction industry relative to the use of CPM scheduling for construction projects." *Journal of Construction Engineering and Management, ASCE*, 132(7), 697–711.
- Goldberg, D. (1989). *Genetic Algorithms in search, optimization, and Machine Learning*. Addison-Wesley publishing company, Inc., New York.
- Goldberg, D., and Deb, K. (1991). "A comparison analysis of selection schemes used in genetic algorithms". *Foundations of Genetic Algorithms*, pp.69-93.
- Gould, F. E. (2005). *Managing the Construction Process: Estimating, Scheduling and Project Control*. Pearson Education, Inc., Upper Saddle River, NJ, USA.
- Halpin, D.W. and Woodhead, R.W. (1998). *Construction Management*. 2nd Ed., John Wiley & Sons, Inc, New York.
- Hawkins, C. V. (2007). "Assessing CPM scheduling software for the small to mid-size construction firm." Master's thesis, Faculty of the Graduate School of the University of Maryland, College Park, Maryland.
- Hegazy, T. (1999a). "Optimization of Construction Time-Cost Trade-Off Analysis Using Genetic Algorithms." *Canadian Journal of Civil Engineering*, 26 (6), 685-697.
- Hegazy, T. (1999b). "Optimization of Resource Allocation and Levelling Using Genetic Algorithms." *Journal of Construction Engineering and Management, ASCE*, 125 (3), 167-175.
- Hegazy, T. (2002). *Computer-Based Construction Project Management*. Prentice Hall, Upper Saddle River, NJ, USA.
- Hegazy, T. (2006). "Simplified project management for construction practitioners." *Cost Engineering Journal, AACE International*, 48(11), 20-28.
- Hegazy, T. (2009). "EasyPlan Project Management System." Available from: <http://www.civil.uwaterloo.ca/tarek/EasyPlan.html>.
- Hegazy, T., Elbeltagi, E., and Zhang, K. (2005). "Keeping better site records using intelligent bar charts." *Journal of Construction Engineering and Management, ASCE*, 131(5), 513-521.
- Hegazy, T. and Menesi, W. (2008a). "Delay analysis considering dynamic resource allocation and multiple baselines." *AACE International Transactions*, CDR.14.1-CDR.14.8.

- Hegazy, T., and Menesi, W. (2008b). "Delay analysis under multiple baseline updates." *Journal of Construction Engineering and Management*, ASCE, 134(8), 575-582.
- Hegazy, T., and Menesi, W. (2010). "Critical Path Segments (CPS) scheduling technique." *Journal of Construction Engineering and Management*, ASCE, 136(10).
- Hegazy, T., and Petzold, K. (2003). "Genetic optimization for dynamic project control." *Journal of Construction Engineering and Management*, ASCE, 129(4), 396-404.
- Hegazy, T., and Zhang, K. (2005). "Daily windows delay analysis." *Journal of Construction Engineering and Management*, ASCE, 131(5), 505-512.
- Hendrickson, C., Zozaya-Gorostiza, C., Rehak, D., Baracco-Miller, E., and Lim, P. (1987). "Expert system for construction planning." *Journal of Computing in Civil Engineering*, ASCE, 1(4), 253-269.
- Herold, S. C. (2004). "Enhanced PDM - Concepts and benefits." *AACE International Transactions*, PS.09.1-PS.09.8.
- Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI.
- Ibbs, W., and Nguyen, L. D. (2007). "Schedule analysis under the effect of resource allocation." *Journal of Construction Engineering and Management*, ASCE, 133(2), 131-138.
- Jaskowski, P., and Sobotka, A. (2006). "Scheduling construction projects using evolutionary algorithm." *Journal of Construction Engineering and Management*, ASCE, 132(8), 861-870.
- Kartam, S. (1999). "Generic methodology for analyzing delay claims." *Journal of Construction Engineering and Management*, ASCE, 125(6), 409-419.
- Kastor, A., and Sirakoulis, K. (2009). "The effectiveness of resource levelling tools for resource constraint project scheduling problem." *International journal of project management*, 27 (5), 493-500.
- Kelleher, A. (2004). "An investigation of the expanding role of the critical path method by ENR's top 400 contractors." Master's thesis, Faculty of Virginia Polytechnic Institute and State University, Blacksburg, Va.

- Kelly, J. E. (1961). "Critical path planning and scheduling: mathematical basis." *Operations research*, 9(3), 167–179.
- Kim, D. (2002). *Exploratory Study of Lean Construction: Assessment of Lean Implementation*. Department of Civil Engineering, University of Texas at Austin, Doctoral Dissertation.
- Kim, H., and Adeli, H. (2001). "Discrete Cost Optimization of Composite Floors Using A Floating-Point Genetic Algorithm." *Journal of Engineering Optimization*, 33(4), pp. 485-501.
- Kim, K. (2009). "Delay analysis in resource-constrained schedules." *Canadian Journal of Civil Engineering*, 36(2), 295-303.
- Kim, K., and de la Garza, J. M. (2003). "Phantom float." *Journal of Construction Engineering and Management, ASCE*, 129(5), 507-517.
- Kim, K., and de la Garza, J. M. (2005a). "Evaluation of the resource-constrained critical path method algorithms." *Journal of Construction Engineering and Management, ASCE*, 131(5), 522-532.
- Kim, K., and de la Garza, J. M. (2005b). "Critical path method with multiple calendars." *Journal of Construction Engineering and Management, ASCE*, 131(3), 330-342.
- Korman, R., and Daniels, S. H. (2003). "Critics can't find the logic in many of today's CPM schedules users want software with flexibility, but is it true CPM?" *Engineering News Record, New York*, 250(20), 30-33.
- Kuhn, A. J. (2006). "CPM not working well on small projects." *AACE International Transactions*, PS.21.1-PS.21.3.
- Lane, A. (1993). *Programming with Genes*. *AI Expert*, 8 (2), pp. 16-19.
- Li, H., Cao, J. N., and Love, P. E. D. (1999). "Using machine learning and GA to solve time-cost trade-off problems." *Journal of Construction Engineering and Management, ASCE*, 125(5), 347–353.
- Li, H., Love, P. (2000), "Genetic search for solving construction site-level unequal-area facility layout problems." *Automation in Construction*, 9 (2), Mar., pp. 217-226.
- 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.

- Lin, C. and Lee, C. (1996). *Neural Fuzzy Systems*. Prentice Hall Inc., USA.
- Livengood, J. and Anderson, M. (2006). "A continuously changing as-planned baseline?" *AACE International Transactions*, CDR 08.1-CDR08.8
- Liu, L., Burns, S. A., and Feng, C. W. (1995). "Construction time-cost trade-off analysis using LP/IP hybrid method." *Journal of Construction Engineering and Management*, ASCE, 121(4), 446–454.
- Lovbjerg M., (2002). *Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality*. Masters Thesis, Aarhus Universitet, Denmark.
- Lowsley, S., and Linnett, C. (2006). *About Time: Delay Analysis in Construction*. RICS Business Services Limited, UK.
- Lu, M., and Lam, H.-C. (2008). "Critical path scheduling under resource calendar constraints." *Journal of Construction Engineering and Management*, ASCE, 134(1), 25-31.
- Lu, M., and Lam, H.-C. (2009). "Transform schemes applied on non-finish-to-start logical relationships in project network diagrams." *Journal of Construction Engineering and Management*, ASCE, 135(9), 863-873.
- Lu, M., and Li, H. (2003). "Resource-activity critical-path method for construction planning." *Journal of Construction Engineering and Management*, ASCE, 129(4), 412-420.
- Menesi, W. and Hegazy, T. (2008). "Five needed enhancements to the CPM for construction projects." *Proceeding Of 2008 Annual CSCE Conference of the Canadian Society for Civil Engineering*, Page No: CO-440 1-10, Québec, Canada, June, 2008.
- Moder, J. J., Phillips, C. R., and Davis, E. W. (1983). *Project Management with CPM, PERT, and Precedence Diagramming*. Van Nostrand Reinhold Company, NY, USA.
- Morad, A. A., and Beliveau, Y. J. (1994). "Geometric-based reasoning system for project planning." *Journal of Computing in Civil Engineering*, ASCE, 8(1), 52–71.
- Moselhi, O. (1993). "Schedule compression using the direct stiffness method." *Canadian Journal of Civil Engineering*, 20(1), 65-72.
- Moussourakis, J., and Haksever, C. (2004). "Flexible model for time/cost tradeoff problem." *Journal of Construction Engineering and Management*, ASCE, 130(3), 307-314.

- O'Brien, J., and Plotnick, F. (2006). *CPM in Construction Management*, McGraw-Hill, New York, NY.
- Ostrowski, V. M. (2006). "Construction CPM scheduling-precision without accuracy." *AACE International Transactions*, CDR.S03.1- CDR.S03.4.
- Tabot, B., and Patterson, J. H. (1978). "An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems." *Management Science*, 24(11), 1163–1174.
- Plotnick, F. L. (2006). "RDM - Relationship diagramming method." *AACE International Transactions*, PS.08.1-PS.08.10.
- Ponce de Leon, G. (2008). "Project planning using logic diagramming method." *AACE International Transactions*, PS.S05.1-PS.S05.6.
- Pogorilich, D. A. (1992). "The daily report as a job management tool." *Cost Engineering Journal*, *AACE International*, 34(2), 23-25.
- Prager, W. (1963). "A structured method of computing project cost polygons." *Management Science*, 9(3), 394–404.
- Rogalska, M., Bożejko, W., and Hejducki, Z. (2008). "Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling." *Automation in Construction*, 18(1), 24–31.
- Ross, J. (1995). *Fuzzy Logic with Engineering Applications*. McGraw Hill Co., New York.
- Russell, J. S., and Radtke, M. W. (1991). "Subcontractor failure. Case history." *AACE International Transactions*, E.02.1-E.02.6.
- Sakka, Z. I., and El-Sayegh, S. M. (2007). "Float consumption impact on cost and schedule in the construction industry." *Journal of Construction Engineering and Management*, ASCE, 133(2), 124–130.
- Sanders, M. C. (2005). "Transparent CPM." *AACE International Transactions*, PS.08.1-PS.08.8.
- Scavino, N. J. (2003). "Effect of multiple calendars on total float and critical path." *Cost Engineering Journal*, *AACE International*, 45(6), 11-15.

- Scott, S. (1990). "Keeping better site records." *International journal of project management*, 8(4), 243-249.
- Senouci, A. B., and Eldin, N. N. (2004). "Use of genetic algorithms in resource scheduling of construction projects." *Journal of Construction Engineering and Management, ASCE*, 130(6), 869-877.
- Siemens, N. (1971). "A simple CPM time-cost trade-off algorithm." *Management Science*, 17(6), 354-363.
- Son, J., and Mattila, K. G. (2004). "Binary resource leveling model: Activity splitting allowed." *Journal of Construction Engineering and Management, ASCE*, 130(6), 887-894.
- Song, Y., and Chua, D. K. H. (2007). "Temporal logic representation schema for intermediate function." *Journal of Construction Engineering and Management, ASCE*, 133(4), 277-286.
- Street, I. A. (2000). "The pitfalls of CPM scheduling on construction projects." *Cost Engineering Journal, AACE International*, 42(8), 35-37.
- Stumpf, G. R. (2000). "Schedule delay analysis." *Cost Engineering Journal, AACE International*, 42(7), 32-43.
- Sawhney, A., Mund, A., Chaitavatputtiporn T. (2003) "Petri net-based scheduling of construction projects," *Civil Engineering and Environmental Systems*, 20 (4), pp. 255-271.
- Wickwire, J. M., and Ockman, S. (2000). "Industry crisis: Construction scheduling software." *2000 AACE International Transactions*, CDR.02.1-CDR.02.8.
- Wiest, J. D. (1964). "Some properties of schedules for large projects with limited resources." *Operations research*, 12, 395-418.
- Winstanley, G., Chacon, M. A., and Levitt, R. E. (1993). "Model-based planning: Scaled-up construction application." *Journal of Computing in Civil Engineering, ASCE*, 7(2), 199-217.
- Winter, R. M. (2003). "Computing the near-longest path." *AACE International Transactions*, PS.11.1-PS.11.9.
- 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.

- Woolf, M. (2008). "When is the critical path not the most critical path?" *AACE International Transactions*, PS.07.1-PS.07.9.
- Zahraie, B. and Tavakolan, M. (2009). "Stochastic time-cost-resource utilization optimization using nondominated sorting genetic algorithm and discrete fuzzy sets." *Journal of Construction Engineering and Management*, ASCE, 135(11), 1162–1171.
- Zhang, H., Li, H., and Tam, C. M. (2006). "Particle swarm optimization for preemptive scheduling under break and resource-constraints." *Journal of Construction Engineering and Management*, ASCE, 132(3), 259–267.
- Zheng, D. X. M., Ng, S. T., and Kumaraswamy, M. M. (2004). "Applying a genetic algorithm-based multiobjective approach for time-cost optimization." *Journal of Construction Engineering and Management*, ASCE, 130(2), 168–176.