The Concurrent Development Scheduling Problem (CDSP)

by

Leroy W. Paul

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Industrial and Management Systems Engineering
College of Engineering
University of South Florida

Major Professor: Michael X. Weng, Ph.D.
Glen H. Besterfield, Ph.D.
Tapas K. Das, Ph.D.
Alexander Nauda, Ph.D.
William A. Miller, Ph.D.

Date of Approval:
October 27, 2005

Keywords: merge point, critical path, task distribution, task duration, standard deviation

© Copyright 2005, Leroy W. Paul
Dedicated to my dear wife Joan without whose constant encouragement and help I would have never succeeded.
Acknowledgement

Dr. Michael X. Weng is acknowledged for his tireless efforts in providing inspiration and ideas, for being a mentor and for critiquing my work over a period of five years as the Concurrent Development Scheduling Model took shape.
# Table of Contents

List of Tables vi
List of Figures viii
Abstract ix

## Chapter 1  Introduction
1.1 Overview 1
1.2 The Concurrent Development Scheduling Problem (CDSP) 1
1.3 Investigation Organization 3

## Chapter 2  Literature Review
2.1 Introduction 6
2.2 Scheduling
  2.2.1 Project Scheduling 9
  2.2.2 Job Shop Scheduling 12
  2.2.3 Timetabling Scheduling 13
  2.2.4 Work Force Scheduling 14
2.3 Overall Scheduling Process 15
  2.3.1 Understanding Project Scope 16
  2.3.2 Organizational Structure of Scheduling 17
  2.3.3 Human Aspects 19
  2.3.4 Technical Aspects of Scheduling 20
2.4 Task Identification 24
  2.4.1 Analyzing “Task” 25
  2.4.2 Impact of Project Size 26
2.5 Task Duration 27
  2.5.1 Single Point Estimates 28
  2.5.2 Three Point Estimates 29
  2.5.3 Multi-point Estimates 30
2.6 Networking 32
  2.6.1 Deterministic Methods 34
    2.6.1.1 Critical Path Method (CPM) 35
    2.6.1.2 Precedence Diagram Method (PDM) 37
    2.6.1.3 Linear Scheduling Model (LSM) 41
    2.6.1.4 Line of Balance (LOB) Method 42
    2.6.1.5 Critical Chain Project Management (CCPM) 42
  2.6.2 Non-deterministic Methods 44
2.6.2.1 Program Evaluation Review Technique (PERT) 45
2.6.2.2 Probabilistic Network Evaluation Technique (PNET) 48
2.6.2.3 Narrow Reliability Bounds (NRB) Method 50
2.6.2.4 Monte Carlo Simulation (MCS) Method 53
2.6.2.5 Simplified Monte Carlo Simulation (SMCS) 55
2.6.2.6 Simulation 56
2.6.2.7 Perry and Greig Method 57
2.6.2.8 Comparison of Non-deterministic Methods 58
2.6.2.9 Curve Fitting 61
2.6.2.10 Non-deterministic Conclusions 63

2.6.3 Optimal Networking 64
2.6.3.1 Enumerative 64
2.6.3.2 Branch and Bound 65

2.6.4 Heuristic Search Methods 65
2.6.4.1 Simulated Annealing (SA) 65
2.6.4.2 Tabu Search (TS) 66
2.6.4.3 Genetic Algorithms (GA) 66
2.6.4.4 Fuzzy Logic 67
2.6.4.5 Petri Nets 68
2.6.4.6 Neural Networks 68
2.6.4.7 Analytical Hierarchy Process (AHP) 68
2.6.4.8 Artificial Intelligence 69
2.6.4.9 Dijkstra’s Algorithm 69

2.6.5 Other Methods 70
2.6.5.1 Project Decomposition 70
2.6.5.2 Material Resource Planning (MRP) 70
2.6.5.3 Fast-Tracking 71
2.6.5.4 “S” Curves 71
2.6.5.5 Learning Curves 72
2.6.5.6 Queuing Theory 72
2.6.5.7 Look-Ahead Techniques 72
2.6.5.8 Manfred’s Distributions 73

2.6.6 Networking Conclusions 73

2.7 Resource Loading 73
2.7.1 Issues with Resource Loading Today 75
2.7.2 Resource Loading Objectives 76
2.7.2.1 Maximizing Net Present Value (NPV) 77
2.7.2.2 Minimizing Project Duration 77
2.7.3 Resource Constrained Project Scheduling 78
2.7.4 Resource Loading Summary 79

2.8 Progress Tracking 80

2.9 Commercial Software 81
2.9.1 Microsoft® Project 2002 (Microsoft Corporation) 82
2.9.2 Primavera Enterprise® (Primavera Systems, Inc.) 83
2.9.3 Primavera TeamPlay® (Primavera Systems, Inc.) 84
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.9.4</td>
<td>Open Plan® (Welcom)</td>
</tr>
<tr>
<td>2.9.5</td>
<td>PS 8™ (Scitor Corporation)</td>
</tr>
<tr>
<td>2.9.6</td>
<td>Commercial Software Comparison</td>
</tr>
<tr>
<td>2.10</td>
<td>Organizations Conducting/Encouraging Scheduling Research</td>
</tr>
<tr>
<td>2.10.1</td>
<td>Project Management Institute (PMI)</td>
</tr>
<tr>
<td>2.10.2</td>
<td>Institute Industrial Engineering (IIE)</td>
</tr>
<tr>
<td>2.10.3</td>
<td>International Council on Systems Engineering (INCOSE)</td>
</tr>
<tr>
<td>2.10.4</td>
<td>Software Program Managers Network (SPMN)</td>
</tr>
<tr>
<td>2.10.5</td>
<td>ProjectWorld</td>
</tr>
<tr>
<td>2.11</td>
<td>Summary</td>
</tr>
</tbody>
</table>

Chapter 3  The Concurrent Development Scheduling Problem (CDSP) 98

3.1 Introduction 98

3.1.1 New Scheduling Technique 98

3.1.2 Assessing Optimism 99

3.2 Overall Process 99

3.3 Task Identification 100

3.4 Task Duration 101

3.5 Networking 101

3.6 Resource Loading 102

3.7 Progress Tracking 102

3.8 Summary 103

Chapter 4  Methodology 104

4.1 Introduction 104

4.2 Fundamentals for Model Development 104

4.2.1 Dependent and Independent Variables 105

4.2.2 Statistics Used 106

4.2.3 Confidence Interval 107

4.2.4 Distribution Functions 108

4.2.4.1 Normal Distribution 109

4.2.4.2 Beta Distribution 109

4.2.4.3 Triangular Distribution 112

4.3 Special Tools and Techniques for Model Development 113

4.3.1 Simulation Techniques 113

4.3.1.1 Monte Carlo Method 113

4.3.1.2 Latin Hypercube Sampling 115

4.3.2 Non-Linear Regression 119

4.3.3 Curve Fitting 119

4.3.4 Correlation Coefficient Determination 122

4.4 Model Development 123

4.4.1 Full Factorial Design with 4 Factors and 3 Treatments 124

4.4.2 Merge Point Contribution to Optimism 128

4.4.3 Full Factorial Design 3 Factors and 3 Treatments 129

4.4.4 Number of Tasks Impact 129

4.4.5 Interaction Between Tasks and Paths 131
4.4.6 Varying Lengths of Parallel Paths 132
4.4.7 Varying Task Durations 135
4.4.8 Scaling the Task Duration Distribution 138
4.4.9 Final CD Scheduling Model (CDSM) 139
  4.4.9.1 Examine Baseline Schedule 143
  4.4.9.2 Produce Network Diagram 143
  4.4.9.3 Produce Sub-Schedule of CD Schedule 143
  4.4.9.4 Use CDSM on Each Merge Point Section 144
  4.4.9.5 Determine Overall Optimism 144
4.5 Comparing the Proposed Model with Typical CD Projects 145
4.6 Comparing the Proposed Model on Real Life CD Schedules 145
4.7 Methodology Summary 146

Chapter 5  Model Development and Simulation Experiments 148
  5.1 Introduction 148
  5.2 Model Development 148
    5.2.1 Full Factorial Design 4 Factors and 3 Treatments 149
    5.2.2 Merge Point Contribution to Optimism 155
    5.2.3 Full Factorial Design 3-Factor 3-Treatment 156
    5.2.4 Number of Tasks Impact 160
    5.2.5 Interaction Between Tasks and Paths 164
    5.2.6 Varying Lengths of Parallel Paths 167
    5.2.7 Varying Task Durations 172
    5.2.8 Scaling the Task Duration Distribution 178
    5.2.9 Final CD Scheduling Model (CDSM) 181
  5.3 Comparing the CDSM to Typical CD Schedules 181
  5.4 Comparing the CDSM with Actual CD Schedules 187
  5.5 Model Development and Simulation Experiments Summary 193

Chapter 6  Findings and Recommendations 194
  6.1 Findings 194
    6.1.1 Reduce Task Duration Estimation Uncertainty 195
    6.1.2 Reduce Task Duration Shapeness 195
    6.1.3 Analyze Longest Merge Point Section First 195
    6.1.4 Break Large Tasks into Small Tasks 195
    6.1.5 Reduce Number of Parallel Paths 195
    6.1.6 Size Task Durations in a Path to be About the Same Duration 196
    6.1.7 Ignore Parallel Paths Less Than 10 Percent of Critical Path 196
    6.1.8 Standard Deviation Determines Optimism 197
    6.1.9 Optimism is Relative to Other Task Durations 197
  6.2 Recommendations for Future Research 197

References 200
List of Tables

Table 2-1 Comparison of Models

Table 4-1 Monte Carlo Method Verses LHS

Table 4-2 Distributions Comparisons

Table 5-1 Full Factorial Design 4-Factors 3-Treatments Results

Table 5-2 ANOVA Full Factorial Design 4 Factors 3 Treatments

Table 5-3 Correlation Coefficient of Step 1 Model

Table 5-4 Step 1 Model Results

Table 5-5 Merge Point Contribution (Results)

Table 5-6 Full Factorial Design 3-Factor 3-Treatment Results

Table 5-7 Analyzing Increasing Numbers of Tasks

Table 5-8 Incorporating a Wider Range of Number of Tasks

Table 5-9 Optimism Versus Number of Paths

Table 5-10 Incorporating Interactions between Tasks and Paths

Table 5-11 Varying Lengths of Parallel Paths with Shapeness +/- 50%

Table 5-12 Varying Lengths of Parallel Paths with Shapenes +/- 25%

Table 5-13 Varying Lengths of Parallel Paths with Shapeness of +/-10%

Table 5-14 Varying the Number of Tasks in a Path

Table 5-15 Varying Task Durations Part 1 of 2

Table 5-16 Varying Task Durations Part 2 of 2
Table 5-17 Scaling Task Duration Distribution Results 180
Table 5-18 Typical CD Schedule Merge Section 1 184
Table 5-19 Typical CD Schedule Merge Point Section 2 and 3 185
Table 5-20 Comparing Models with Simulations and CDSM 187
Table 5-21 Project 3 Analyzed 189
Table 5-22 Completed CD Projects 190
Table 5-23 Predicting the Results with the CDSM on an Actual Project 192
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>The Scheduling Process</td>
<td>21</td>
</tr>
<tr>
<td>2-2</td>
<td>Activity on Arrow (or Arc) (AOA)</td>
<td>36</td>
</tr>
<tr>
<td>2-3</td>
<td>Activity on Node (AON) Diagram</td>
<td>38</td>
</tr>
<tr>
<td>2-4</td>
<td>Task Dependencies</td>
<td>39</td>
</tr>
<tr>
<td>2-5</td>
<td>Triangular Duration Density Function</td>
<td>54</td>
</tr>
<tr>
<td>4-1</td>
<td>CD Scheduling Sample</td>
<td>127</td>
</tr>
<tr>
<td>4-2</td>
<td>CDSM Part 1</td>
<td>141</td>
</tr>
<tr>
<td>4-3</td>
<td>CDSM Part 2</td>
<td>142</td>
</tr>
<tr>
<td>5-1</td>
<td>Merge Point Contribution (Models)</td>
<td>157</td>
</tr>
<tr>
<td>5-2</td>
<td>Varying Lengths of Parallel Paths</td>
<td>168</td>
</tr>
<tr>
<td>5-3</td>
<td>Varying Task Durations</td>
<td>173</td>
</tr>
<tr>
<td>5-4</td>
<td>Varying the Number of Tasks in a Path</td>
<td>174</td>
</tr>
<tr>
<td>5-5</td>
<td>Scaling Task Duration Distribution</td>
<td>179</td>
</tr>
<tr>
<td>5-6</td>
<td>Typical CD Schedule</td>
<td>183</td>
</tr>
<tr>
<td>5-7</td>
<td>Typical CD Schedules</td>
<td>186</td>
</tr>
</tbody>
</table>
The Concurrent Development Scheduling Problem (CDSP)

Leroy W. Paul

ABSTRACT

The concurrent development (CD) project is defined as the concurrent
development of both hardware and software that is integrated together later for a
deliverable product. The CD Scheduling Problem (CDSP) is defined as most CD
baseline project schedules being developed today are overly optimistic. That is, they
finish late. This study researches those techniques being used today to produce CD
project schedules and looks for ways to close the gap between the baseline project
schedule and reality. In Chapter 1, the CDSP is introduced. In Chapter 2, a review is
made of published works. A review is also made of commercial scheduling software
applications to uncover their techniques as well as a review of organizations doing
research on improving project scheduling. In Chapter 3, the components of the CDSP are
analyzed for ways to improve. In Chapter 4, the overall methodology of the research is
discussed to include the development of the Concurrent Development Scheduling Model
(CDSM) that quantifies the factors driving optimism. The CDSM is applied to typical
CD schedules with the results compared to Monte Carlo simulations of the same
schedules. The results from using the CDSM on completed CD projects are also
presented. The CDSM does well in predicting the outcome. In Chapter 5, the results of
the experiments run to develop the CDSM are given. In Chapter 6 findings and
recommendations are given. Specifically, a list of findings is given that a decision maker can use to analyze a baseline project schedule and assess the schedule’s optimism. These findings will help define the risks in the CD schedule. Also included is a list of actions that the decision maker may be able to take to reduce the risk of the project to improve the chances of coming in on time.
Chapter 1

Introduction

1.1 Overview

This investigation looks at all aspects of the Concurrent Development (CD) Scheduling Problem (CDSP) to find ways to improve the initial schedule development process so that the resulting schedules more closely match reality. The CD project is defined as the concurrent development of both hardware and software that will be integrated together to produce a deliverable product. The CDSP is defined as most all baseline CD project schedules being developed today are overly optimistic. That is, they finish late. The concept is if a project finishes later than originally predicted, the original schedule was optimistic. Optimism can be a negative number, which is interpreted as pessimism. Here the project completed earlier than expected and the original schedule was pessimistic.

1.2 The Concurrent Development Scheduling Problem (CDSP)

Not surprisingly, most program offices resort to some sort of software application program for their scheduling development and tracking progress needs. There are many scheduling and tracking software applications programs available today and they are used in great number. About 92% of the companies recently surveyed use a project management software package for scheduling and about 80% of them also use it for tracking (Pollack-Johnson and Liberatone 1998). The most popular application tool is Microsoft® Project with a reported use by over 72% of project offices. However, there
are many competing application tools. The amount of time and resources needed to both schedule and track a project can be considerable, but the produced plans are rarely more accurate predictions of what it will take to complete the project. A study (Githens 1998) shows that virtually all projects experience cost over-runs, although those with more planning do over-run less. Cost, schedule and performance are tightly interwoven and often more than not when one fails to meet target the other two also do not meet their targets. Projects that involve concurrent development of both hardware and software are particularly challenging. The task is difficult enough when Microsoft® wants to develop a new version of their word processor or Intel® wants to develop a new microprocessor but when both are done concurrently and needs to be integrated together to produce a deliverable product the task becomes extremely complex. Meeting schedule is particularly important in the CD project with the rapid advance in technology today. A schedule not met may mean the market will be missed or if a customer is paying for the development, the customer may cancel the order, not to mention the longer-term impact on the company’s reputation of being able to deliver products on time. Improving the accuracy of the initial CD schedule duration is the focus of this research. In the ideal world, that project schedule will accurately depict all the tasks required to perform the project, the task durations will give the actual amount of time and effort required to accomplish the tasks and the tasks will be in the right relational order to make the most efficient use of resources available. This ideal schedule will also be able to be updated easily and will accurately reflect the progress of the project. The project manager will be able to use this ideal schedule to assess progress and take actions that will have predicted results to correct identified problems as they are uncovered. Project scheduling is of
critical importance to every project manager. Unfortunately, evidence shows (Githens 1998) schedules developed at the beginning of a project rarely predict the outcome.

1.3 Investigation Organization

This investigation looks for ways to close the gap between the initial schedule and reality. In Chapter 2 Literature Review the results of a literature search is presented of the project scheduling techniques and ideas that have been offered. The conclusion is there is a vast array of techniques but only a few have found acceptance in the market place as evidenced by what providers of scheduling software are putting into their scheduling application software. The most popular technique is one of the oldest and that being the Precedence Diagramming Method (PDM). This method is often incorrectly referred to as the Critical Path Method (CPM), which preceded PDM by a few years. The differences between CPM and PDM are discussed in Sections 2.6.1.1 and 2.6.1.2. An integral part of PDM and many like techniques is that of finding the critical path or that chain of tasks that is the longest path through the network. That critical path is then most often given as the expected duration of the project. However, the literature review has shown that this critical path or expected duration is often optimistic. Many reasons were offered why this is so, from the scheduling technique used to project office organization to motivation to a purely structural point of view as to how the schedule is developed. This study focused on the structural factors impacting optimism. Chapter 3 explores this structural phenomenon in more detail. The reasons for this structural optimism arise from a number of sources, which were uncovered from the literature review and by analyzing the components of schedule development: overall process, task identification, task duration, networking, resource loading and progress tracking.
One structural factor found to produce optimism is the task duration distribution. Task durations are often entered as single point estimates as in PDM with the most likely task duration being that point estimate. However, most tasks have a task duration distribution, which give rise to optimism. Also, these tasks are rarely normally distributed which is often assumed when a distribution is assumed but they are really distributed in a way that gives rise to even further optimism.

Another structural factor contributing to optimism is the merge point phenomenon. A merge point is where two or more tasks or chain of tasks (or paths) entered another task (a merge point) where all the entering tasks must be completed before the merge point task or to the task(s) to follow can be started. The result is the more parallel paths into a merge point the more the optimism. The merge point phenomenon is especially severe when the merge point is on or close to the critical path.

Other structural factors found contributing to schedule optimism were the number of tasks in the schedule, the limits of the task duration distribution and the overall uncertainty of the task duration estimate. The limits of a task duration distribution describe the maximum percent under the mode of a task duration distribution to the maximum percent over the mode. These limits will be quantified in a parameter called shapeness represented by the letter S. All these structural factors contribute to the initial CD schedule optimism.

Chapter 4 titled Methodology takes on the challenge to identify those structural factors impacting optimism, their relationships and determines the magnitude of their impact on optimism. From this the Concurrent Development Scheduling Model (CDSM) was developed. The CDSM was then tried on a variety of CD schedules. The results of
the CDSM were compared to Monte Carlo simulations of the same schedules. The
CDSM was also tried on the initial schedules of several completed CD projects. The
results were compared to the actual results. Monte Carlo simulations were also
performed and compared to both the CDSM and the actual results.

Chapter 5 titled Model Development and Simulation Experiments gives the
experimental data used to develop the CDSM. As the CDSM evolved a number of
experiments were designed and run in order to gain insight into the relationships and
magnitude of the impact. The results are presented here.

Chapter 6 titled Findings and Recommendations states those structural factors
found to impact optimism, their relationships and by how much they impact optimism. A
decision maker can use these findings to focus in on those parts of the schedule that adds
to the schedule optimism. Also listed are actions that may be taken to lessen the impact.
Chapter 2

Literature Review

2.1 Introduction

In this chapter a literature review is made of the research conducted on improving project scheduling. To start this search a review was made of recent project surveys on scheduling and who is using scheduling, a review of those using scheduling and an outline of the rest of the chapter. A number of surveys have been conducted on the state of project scheduling from general to a specific focus. These surveys gave suggestions for the overall literature review. One general survey (Kolisch and Padman 2001) was conducted on the overall topic of project scheduling primarily focusing in on deterministic project scheduling from problem generation to scheduling to special problems. Their survey presented a good review of techniques developed to date. Another survey (Icmeli, Erenguc et al. 1993) specifically addressed the following scheduling problems: the resource-constrained project scheduling problem (RCPSP); time/cost trade-off problem (TCTP); and payment scheduling problem (PSP). Another survey (Domschke and Drexl 1991) considered advances in modeling and solving the RCPSP. A recent survey (Herroelen, Reyck et al. 1998) dealt with methods just to solve resource constrained projects. Another survey (Elmaghrabya 1995) considered the advances in scheduling from 1987 to 1995. This researcher considered a program manager has the following concerns: representation and modeling for visualization and
analysis, scheduling activities which are subject to resource constraints, financial issues related to project compression and cash flows and uncertainty in activity durations. The survey was conducted in these areas. Another survey (Brucker, Drex.lb et al. 1999) reviewed projects to establish a common notation especially to bridge the gap between job shop and project scheduling methods. Other surveys included one (Herroelen, Dommelen et al. 1997) that reviewed all recent methods to maximize the net present value (NPV). The above surveys were reviewed for their impact on the concurrent development scheduling problem as recorded below.

Many disciplines are interested in improving project scheduling. For example, a study (Kuo, Liu et al. 2001) was conducted to optimize a project on the irrigation of crops. They used a simulated annealing approach (see Section 2.6.4.1). In another case (Baber and Mellor 2001) the researchers were concerned with optimizing a human computer interaction project. They used the critical path method (CPM) (see Section 2.6.1.1). CPM is being used in a wide variety of applications from construction to claims management (Singh 2002). Another set of researchers (Chu and Cesnik 1998) were concerned with optimizing a health care project. They used the Precedence Diagramming Method (PDM)(see Section 2.6.1.2).

In this study we are interested in improving CD project scheduling. These scheduling techniques can be divided into two groups. One group of techniques strives to simply describe the project schedule as accurately as possible and the other group strives to optimize the project schedule for some reason such as the shortest development time. Much has been written on both and they will be reviewed in this chapter. A key impetus to current day scheduling research was the development of the CPM by the US Navy
Special Projects Office in 1958 (Kelly 1961). In the 47 years since then, many researchers have looked for ways to improve the scheduling process. However, CPM/PDM are still widely used today probably because it is easy to understand and use. Other proposed techniques have not caught on probably because most require considerably more effort for a small gain in fidelity. The biggest advance has been in the speed of use. The original method was by hand and although rather direct, the effort to schedule a project of any practical size and determine its critical path by hand quickly became overwhelming. Fortunately, computers and, in particular, personal computers came along at the right time and were an ideal match to take away most of the “physical labor” of scheduling. In addition, the profession of project management came into being during this time producing a customer base in need of this product. The software application industry has responded with many products. In Section 2.9 of this chapter the explosion in commercial scheduling software available is discussed and what techniques they are using to develop a project schedule.

Based on these surveys and who is using project scheduling, the remainder of this section is organized. First, the topic of how the CD scheduling problem fits into the overall topic of scheduling is discussed. This is covered in Section 2.2. Next in Section 2.3 the overall approach to project scheduling process is reviewed. Reviewing the five components of project scheduling is next: task identification (Section 2.4), task duration (Section 2.5), networking (Section 2.6), resource loading (Section 2.7) and lastly project tracking (Section 2.8). The results of this literature review are incorporated in Chapter 3 and form the basis for the remainder of this study. Next, Section 2.9 is devoted to reviewing commercial software applications on project scheduling. Lastly in Section
2.10 the work of those professional organizations actively engaged in or encouraging core research into improving project scheduling is reviewed.

2.2 Scheduling

The overall topic of scheduling can be divided into the following four subgroups: project scheduling (usually one of a kind typically large project), production or job shop scheduling (includes lot sizing and batching), timetabling scheduling (reservation systems) and work force scheduling (crew scheduling) (Pinedo and Seshadri 2002). The focus of this research is the CD project schedule, which is a subset of project scheduling which in turn is a subset of scheduling. Each of these subgroups will be briefly reviewed for advancements to set the overall environment for the CD project scheduling problem.

2.2.1 Project Scheduling

Project scheduling is about building a highway, developing an electronic communications black box and building of the International Space Station. A project schedule can be defined in several ways. Taking the words project and schedule separately the definition of project as defined in the Project Management Institute’s (PMI) “A Guide to the Project Management Book of Knowledge PMBOK® Guide 2000 Edition” is “a temporary endeavor undertaken to create an unique product or service.” Another definition given in Effective Project Management (Wysocki, Jr. et al. 2000) is “A project is a sequence of unique, complex, and connected activities having one goal or purpose and that must be completed by a specific time, within budget and according to specification.” The schedule can be defined as the time phase assignment of resources to accomplish the project plan. Therefore, a project schedule is defined as the time phasing of resources to accomplish all the tasks necessary to accomplish a temporary endeavor
undertaken to create a unique product or service. Again in the words of the PMBOK® Guide a project schedule is “The planned dates for performing activities and the planned dates for meeting milestones.” The predecessor to most project schedules involved in project management is the project plan. The project plan lists the what and when of the project. Planning is then the process of developing the plan and scheduling is the process of developing the schedule. The words planning and scheduling are often used interchangeably. Care will be used throughout this study to accurately label an activity as planning or scheduling.

A project is usually one of a kind and typically large. It usually has never been done before nor will it be undertaken quite like this in the future. Hundreds and even thousands of tasks are often needed to accomplish a project. In the electronic concurrent development business, 2000 tasks to develop an electronic communications unit (a black box) are not unusual. There is one dominant technique for determining project duration for these large task projects. That technique started with the Critical Path Method (CPM). It was introduced by Kelly of Remington Rand and Walker of Dupont in 1957 (Kelly 1961) and has not changed much since. The CPM evolved into the Precedence Diagramming Method (PDM), which is used by most software applications. Both are deterministic methods with no attempt made to account for the probabilistic nature of tasks. PDM continues in wide use today due in part because it is easy to understand and to use. The construction industry is not only a big user of PDM it has been studied for improvement. A recent study (Hegazy 2001) integrates PDM, schedule development and detail schedule presentation down to work crews (work force scheduling). Many U.S. government contracts require its use. To use the method simply list all the tasks, list the
durations of each task, arrange the tasks in a logical sequence and assign all the constraints and dependencies associated with each task. The technique then leads you to find the critical path(s) based on the assumed tasks and constraints. When the determined critical path schedule is longer than the program requirements, the tasks, constraints and resources applied need to be challenged to determine if any can reasonably be changed with the result of improving the schedule duration. Schedule generation can be complex on development projects since this may be the first time anything like this project has ever been attempted. The skill of the scheduler along with the program team determines the reasonableness of the developed schedule. A typical scenario to deal with a schedule that does not meet program needs is to require the program team to add/subtract defined/redefined tasks; add resources to tasks on the critical path and to change relationships, constraints and dependencies. The critical path of this new schedule is then found. If this new schedule still does not meet program needs, the cycle is repeated. CPM/PDM is then just a tool used by the program team as they iterate through numerous schedules before arriving at one that will become the baseline. See Section 2.6.1.1 for a review the details of CPM and Section 2.6.1.2 for PDM. Although CPM/PDM have proved to be very useful, many have proposed alternatives. One of the first was called the Program Evaluation and Review Technique (PERT) method (Malcolm, Roseboom et al. 1959), which incorporated variability of tasks (see Section 2.6.1.2). In Section 2.6 the wide range of other techniques that have been developed are reviewed. Many of the techniques for scheduling of projects today simply determine how long a project will take and leave it to the scheduler and program team to find ways to a schedule that meets the
project requirements. There are also wide ranges of techniques to optimize the project schedule. They are also reviewed in Section 2.6.

2.2.2 Job Shop Scheduling

Job shop scheduling or sometimes called production scheduling is closely related to project scheduling. There are two major differences between job shop scheduling and project scheduling. Before addressing the differences, a few words on what kind of work is included in a job shop and what is included in a project. A job shop includes paced production lines, unpaced production lines, lot sizing and batching. With that definition in hand, the first difference between a job shop and a project is that the number of tasks that make up a typical job shop schedule is far fewer than that of a typical project schedule. The result is that different techniques have been developed over time to address the scheduling needs for these two situations. The second major difference is the number of times the shop job/project is repeated. A typical job shop will be done many times. Repeating that job over ten thousand times is not unusual. As a result of this repetition, the range of times to complete a task will become accurately known. In contrast, a typical project is completed only once and often the length of time estimated to do the task is often just a guess. Not surprising, different techniques have been developed to address job shop and projects. In the following sections, a review of some of these techniques is made.

There are a number of techniques for job shop scheduling that are aimed at optimizing throughput or run time or makespan. These techniques include a Decision Support System developed for scheduling of steel fabrication projects but applicable to other type projects (AbouRizk and Karumanasserí 2002). Another technique is the use of
genetic algorithm to address production job shops. A set of researchers (Wang and Zheng 2002) proposed and enhanced genetic algorithm. A new technique (Hino and Moriwaki 2002) called recursive propagation method has been applied decentralized manufacturing systems where each machine only has to notify the change of its plan to other machines, which are directly influenced by the change. Another set of researchers (Satake, Morikawa et al. 1999) used the human scheduler to view a gantt chart of a non-optimal job shop schedule. They found that the human scheduler often finds a better schedule by changing its operation sequence. They then used a simulated annealing method to avoid local minimum states. Other techniques are COMSOAL and Ranked Positional Weight heuristic for line balancing (Askin and Standridge 1993). One of the features of all these techniques is they are limited to jobs that have relatively (when compared to projects) few tasks. In the job shop scheduling environment many jobs can be modeled in a few number of tasks or can be broken down logically into a number of small projects and solutions can be found. Another feature is that over time much becomes known about the task itself, in particular, how long does it takes. Armed with this information the above techniques become quite useful. A recent study (Rom, Tukel et al. 2002) offers up a way to use MRP normally used for production scheduling for the scheduling of job shop projects. This technique adds capacity constraints and variable lead time lengths to adapt MRP to job shop scheduling.

2.2.3 Timetabling Scheduling

Timetabling is the process of scheduling classrooms, meetings or reservation and the like. The problem can be defined as assigning a number of events into a limited number of time periods. Wren (Wren 1996) defines timetabling as follows:
"Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space time, in such a way as to satisfy as nearly as possible a set of desirable objectives."

The university timetabling problem is universal to all universities. There are two main categories: course timetabling and exam timetabling has drawn the attention of a number researchers. A recent set of researchers (Burke and Petrovic 2002) took on the task of university timetabling problem but believe their findings will have wider appeal. These researchers improved current heuristic and meta-heuristic methods for timetabling (where they decomposed large real-world problems), offered a multicriteria approach to timetabling and developed an application of case-based reasoning to timetabling (where you used knowledge gained on the previous problem for the new problem). Others (Asratian and Werrab 2001) have also developed algorithms to solve the university timetabling problem.

Another set of researchers (Zwaneveld, Kroon et al. 2001) took on the task of the scheduling problem of routing trains through a railway station in the Dutch railway system. They described the problem in terms of a weighted node packing problem and developed an algorithm to optimize the routing.

2.2.4 Work Force Scheduling

Work force scheduling includes the process of crew scheduling. One set of researchers (Crocia, Perona et al. 2000) studied the relatively short duration time intervention of the human crew and automated machines. They developed a set of rules on crew size, the way tasks are assigned to operators and the way operators are assigned to machines in the shop. Another set of researchers (Mazzola, Neebe et al. 1998) took on
the problem of multiproduct production planning problem in the presence of work-force learning (MPPL). They developed several formulations to solve this problem from a branch-and-bound algorithm to a nonlinear mixed-integer programming problem to a tabu search technique. They found the tabu search approach provided good results to the MPPL problem. Another set of researchers (Trentesaux, Tahon et al. 1998) considered production activity control systems as centralized, decentralized or hybrid. Their developed hybrid approach used the notion of bottleneck and non-bottleneck resources and only scheduling the bottleneck resource. They assert that their approach support just in time scheduling with good results.

The specific case of work force scheduling of call centers has seen considerable research. One set of researchers (Brusco and Jacobs 2001) considered the number of start times in the work force of a call center. They used a spreadsheet-based program designed approach. They used their approach on a service support center and reported their results as a case study.

2.3 Overall Scheduling Process

Scheduling is a complex endeavor. One way to review the scheduling process is to consider the following four factors. The first factor is to understand the scope of the project to include risk management and to be on the alert for scope crept. The second factor is the organizational structure in which the schedule is being developed to include functional and project management. The third factor is the human side to include motivation, training and talent. The last factor is the technical aspects of scheduling to include the algorithms, tools and techniques to actually develop the schedule. Each of
these factors will be briefly reviewed to set the stage for an in depth review of the technical aspects of scheduling to be addressed in the sections that follow.

2.3.1 Understanding Project Scope

Scheduling is about planning and timing. An area given much attention is inadequate scope definition. One researcher (Uppal 2002) asserts the problem is a management problem in the generation of the scope of work. He further asserts the problem to be in the following areas: “1. Management rarely worries about a project scope of work. 2. Management has preconceived numbers in mind even before the project scope of work is established. 3. Project completion dates are established without a real project scope of work. 4. Management misuses old estimates to fit new conditions with total disregard to the new project scope of work.” One set of researchers (Dumont, Gibson et al. 1997) developed a system to assess the state of the project definition. They developed a project definition rating index (PDRI) to address these problems. The PDRI is a weighted checklist of 70 scope definition elements to measure the level of scope definition.

The subject of project risk management has been subject to much research and rightly so. The realization of a risk can destroy any schedule. One set of researchers (Miller and Lessard 2001) categorized project risks as (1) market-related: demand, financial and supply; (2) completion: technical, construction and operational; (3) institutional: regulatory, social acceptability and sovereign. They go on to address each risk and provide techniques to identify and track the processes to cope with them.

Normally the topic of risk is applied to an individual project. A set of researchers (Baccarini and Archer 2001) took on the task of ranking projects by risk and developed a
technique. They used their technique on projects being managed by the Department of Contract and Management Services (CAMS) - a government agency in Western Australia (WA). Another set of researchers (Ward and Chapman 2002) considered the word risk encourages a threat perspective and made the case for “uncertainty management” as a better way to address project “risks.”

The special topic of software development risk mitigation has seen considerable research. One set of researchers (Houston, Mackulak et al. 2001) developed their own set of software development risk factors (SDRFs) by using a questionnaire. Some have selected over 150 factors but these researchers selected the following from their survey.

1. Creeping user requirements
2. Inaccurate cost estimation
3. Excessive schedule pressure
4. Lack of staff commitment, low morale
5. Instability and lack of continuity in project staffing
6. Lack of senior management commitment
7. Based on these risk factors, they developed a model to assist the risk of the software development.

Understanding the project scope and understanding the risk involved in any project can have a marked impact on the schedule being generated and most importantly, the overall outcome of the project.

### 2.3.2 Organizational Structure of Scheduling

Much has been written on how to best organize successful project completion. The assumption being that you might have the best schedule but with a poor
organizational structure to execute the project, the project is doomed to failure. Here is a review of recent research with sometimes conflicting ideas on what will work, as is the case with the first example. In a recent study (Teasley, Covi et al. 2002) the entire project team was placed in a large room in an arrangement called radical collocation with markedly improved results. Another study (Swink 2002) surveys 132 projects and shows that the common techniques of organizational tactics such as co-location and team isolation produced insignificant results while quality function deployment (QFD) and computerized project scheduling produce good results for accelerated new product development (NPD). Another study (Al-jibouri 2002) undertook the management of a project. Here they considered central management, section management with co-operation between the sections and independent section management as possible ways to manage a project. Section management with co-operation between the sections proved to be the best. Another set researchers (Block and Frame 2001) looked at organization considering the utility of project management offices which oversee/control a number of projects giving reasons for adopting such an organization. Another researcher (Pitagorsky 2001) studying project management determined that the discipline of project management needed a scientific approach similar to Supply Chain Management. This same researcher studied the relationship between project management and functional management (Pitagorsky 1998). He offers ideas to promote collaborative relationships that include “1. Stabilizing project resources, 2. Functional manager involvement in planning, 3. Project manager authority, 4. Accountability, 5. Custodianship versus ownership of functional resources, and 6. the functional manager's role in project
performance and direction.” This is just a sampling of what has been written with the underlying theme that a poor organization structure can doom a project.

2.3.3 Human Aspects

There is a rich source of research material on the human aspect of project success and developing project schedules. A recent study (Thoms and Pinto 1999) placed a great deal of responsibility for project success on the timing of the project manager’s temporal skills. These authors defined the key temporal skills as time warping, creating future vision, chunking time (creating units of future time to be used for scheduling), predicting and recapturing the past. Others (Das 1987) have also studied temporal perspectives and defined individual future time perspective (FTP) as a key element in effective long range planning. This author (Das 1991) later refined this concept into studied planning horizons are key to effective scheduling. In particular, far-future executives are better equipped than near-future executives in developing effective plans. Another study (Milosevic 1999) found that program management language was not universal and can contribute to misunderstanding. They call this the Silent Language of Project Management. Motivation is a major factor in successfully completing a project. The human aspect includes the motivation of the project manager and project team to develop and maintain a realistic schedule to the type of scheduling organization. A recent study (Turner, Utley et al. 1998) examined the motivation of project managers compared to the functional managers. The study identified a major motivation factor to be job satisfaction, which is different for project managers and functional managers. Still another set of researchers (Kliem and Anderson 1996) followed by (Slabey and Austrom 1998) used a style survey tool for program management called Decide-X to facilitate the
team building process. The tool identifies 4 primary styles in how a person approaches relevant work situations. They are: “1. Reactive Stimulator, 2. Logical Processor, 3. Hypothetical Analyzer, and 4. Relational Innovator.” Still another set of researchers (Thomas, Tucker et al. 1999) developed yet another tool called Compass designed to assess the communications project assessment tool by obtaining answers to questions asked of the team members. The tool was validated against 71 Construction Industry Institute (CII) projects. This is a sampling of what has been reported on the human aspects to produce a successful project. There is no doubt a highly motivated team can overcome a flawed schedule but a good schedule with a motivated team can produce outstanding results.

2.3.4 Technical Aspects of Scheduling

Here is reviewed the overall mechanics of actually developing the schedule. As will be shown, many techniques have been developed to produce a schedule. To develop a satisfactory schedule means you want to develop a schedule based on some optimal criteria. To do that, it is most helpful to formulate the scheduling problem into some sort of mathematical model. Most mathematical models involve the following three characteristics. First, there is the number of machines, resources or tasks involved in the project. Then there are the processing requirements and constraints and lastly, there is the objective of what is to be optimized. These models are often formulated as linear programs, integer programs or disjunctive programs (Pinedo and Seshadri 2002). These models will be reviewed in Section 2.6 below.

From a slightly higher view of scheduling and in particular, most techniques used on the CD scheduling problem, the scheduling process can be divided into the following
components: task identification (see Section 2.4), task duration (see Section 2.5), networking (see Section 2.6), resource loading (see Section 2.7) and progress tracking (see Section 2.8). Each of these components is reviewed in the sections as indicated above but before doing that a short review on how these components are interrelated to form a schedule is made. The normal scheduling process can be described as shown in Figure 1.

![Figure 2-1 The Scheduling Process](image)

The process involves making a list of the tasks required to accomplish the project, determining their durations and determining the number of resources assumed to be available to the task. The next step is to network the tasks. In this step the tasks are arranged in a logical sequence in which they are to be performed along with their dependencies and constraints. This step will often uncover tasks that need to be added or consolidated and the scheduler is returned to the first step as the arrow between Network and Tasks shows. Once a reasonable network is produced the schedule step is undertaken. In this step a critical path is usually determined. However, this not always the case depending on the technique being used to develop the schedule. For example, in the Linear Scheduling Module method (see Section 2.6.1.3) rates are used and no critical path is found. Returning to the case where a schedule is required, the initial result will
often not meet the overall program requirements i.e. the critical path will be longer than when the project needs to be completed or there is the question - can we do better in optimizing the schedule? If this is the case, the scheduler returns to the Network step as the arrow shows. One of the questions to be answered is can the tasks be arranged in a different order to shorten the project assuming that minimizing the project duration? That is, can some items be accomplished in parallel? Another question is can additional resources be added to shorten the length of the tasks on the critical path? If additional resources are not available in house, can a task or tasks be subcontracted out? Sometimes tasks can be eliminated or deferred with a small increase in risk i.e. elimination of an intermediate inspection or test point. Up to this point the schedule improvement has been dependent on the skill of the scheduler and program team. There is however a long list of other techniques to optimize the schedule (see Section 2.6). The end result must be a schedule that meets the program needs or else the project should not be undertaken. The scheduling process for a CD project is the above process conducted on each of the concurrent major task branches. These branches are connected together at merge points to produce the overall schedule, which makes the overall scheduling process much more complicated. In the sections to follow each of the components of the scheduling process is reviewed.

An area seeing increased study is the way to increase the speed in sharing of information. One new tool is to make the information web based in a manner that the information can be readily shared. A very recent study (Tserng and Lin 2003) from the construction business attempts to use a variety of currently available information defined by the eXtensible markup language Schema for Scheduling (XSS), the Data Acquisition
Language for Scheduling (DALS), the Hierarchy Searching Algorithm (HSA) and an automatic mechanism called Message Transfer Chain (MTC), an Electronic Acquisition Model for Project Scheduling (e-AMPS). The study develops an information agent, Message Agent (MA) in an eXtensible markup language (XML) format that can be shared by others in the scheduling process.

One last area that needs to be mentioned on reviewing the technical aspects of scheduling development is the explosion of software applications available to aid in the development of project scheduling. This is reviewed in Section 2.9.

This research is directed at the technical aspects of scheduling and leaves the other factors of scheduling as reviewed in Sections 2.3.1 through 2.3.3 above to others to study. In the following sections the various technical aspects of project scheduling are reviewed for ways to reasonably predict the outcome of the project. The focus of this research is on the CD scheduling problem. To further reduce this set of projects, the study was limited to CD schedules in the electronic communication devices industry. There are several reasons to narrow the type of projects studied in this search. First, the electronic communication’s business is a huge business. Many companies are engaged in this type of effort and most could use a more efficient scheduling technique as noted in Chapter 1. Second, the results of this study are probably applicable to a much larger set of projects but that is left to future study. Limiting the study to the stated set of projects will greatly help the many variables that can impact on the success of a project. Lastly, access to raw data of CD project schedules both on going and completed was available. This raw data was used to validate the model developed.
In the next sections the remaining components of the overall CD scheduling process are analyzed. The components are: task identification, task duration, networking, resource loading and progress tracking.

2.4 Task Identification

One of the first steps to take in developing a schedule is the identification of the tasks to be done. Task identification is the listing of the individual tasks to be accomplished. There are many factors that can be considered in establishing the rules to be used selecting how tasks are identified. There are many factors to consider in establishing the rules for task identification. On factor is how many tasks should a project schedule have? In the Pollack-Johnson and Liberatore study in 1998 (Pollack-Johnson and Liberatore 1998), the median size project had 150 tasks, and a large one had over 10,000 tasks. Another factor is the uncertainty of its start time. A set researchers studied (Maniezzo and Mingozzi 1999) the uncertainty of start time and found an optimal solution using integer programming formulation of this problem. Another factor is should the task be functionally identified or project identified. For example, should the task be the detail design by the digital engineer or should the task be the preliminary design of the processor board, which may involve digital, mechanical and component engineers. This question is important in determining ownership of the task. Still another factor is how to handle level of effort (LOE) kinds of tasks such as program management. The issue here is that little can be learned from a LOE task on the overall project duration and seems desirable to minimize. The claim was made at the 1999 Project Management Institute (PMI) Symposium that one reason a project fails is that not all tasks are identified. He presented techniques to ensure tasks are not forgotten. Another presenter
made the case that LOE tasks are often not properly considered and has a tool that ensures they are.

2.4.1 Analyzing “Task”

The definition of “task” is fundamental to any scheduling. Sometimes the word used is activity. There seems no reason to choose one or the other. Task is used throughout this study. It is the building block for every project. There are a number of issues to be addressed in defining what is a task. One issue is how small a block of work should one task cover. The smaller a task, the larger the number of tasks there will be and the more to track. There are projects today that have tens of thousands of tasks. A place to start is with the lowest level task as one defined as a manageable unit of work that has a readily definable start and a readily definable finish. One person or one group of people can accomplish that task. One individual can also manage the task. A task must be able to be assigned to a person or a group of persons and they must be able to understand what they are to do, what constitutes starting and what must be done so they know they are finished or what is their completion criteria. Another issue with the use of tasks is that an understandable hierarchical scheme needs to be in place. A human will have difficulty in integrating more than 20 of anything to include tasks. Therefore, summary tasks must be defined that roll up subordinates’ individual tasks. This hierarchy can and often does go on for many levels. In establishing these higher levels, the supervisor or manager that will be reviewing the work progress must be foremost in their minds. Another aspect is how long will a task take or what is the task duration. In the CD project estimating task duration has a great deal of uncertainty. One set of researchers (Wang 2002) used possibility theory to develop a fuzzy beam search
algorithm to address the uncertainty of task duration. The answer is it depends on a number of factors. One factor is the type and amount of resource that is available to be applied to this task. There are also other aspects such as how do you determine how much resource is required to do a particular task. There are a number of methods that have been offered. One method for early software estimation was offered in a recent study (Huang 2001). Here they offer a two-stage software sizing process and product decomposition technique to develop a schedule in compliance with the International Standard Organization (ISO) ROSE standard.

2.4.2 Impact of Project Size

Different size projects, as defined by the number of tasks required to accomplish the project have different scheduling techniques available to them, which in turn helps select a scheduling technique for a project. Most small projects require only thinking about what is to be done. People plan almost continually with most of that planning never written down. The tasks, durations, constraints and dependencies are entered into their mind and a plan or schedule appears. For example, on my way home I plan to stop for gas and then the grocery store for bread and milk. This will add 15 minutes to my drive home. However, there are occasions when you will want to show projects of a few tasks in writing. As an example, project managers are often called upon to give the status of their project. A schedule is invariably included in the briefing and often it is stated on one chart. That schedule may have 5 to 10 tasks. Even if a detailed schedule is actually in place, the summary schedule will be the one presented. Sometimes simply a list of dates and activities will be used. Often bar charts or Gantt charts will be used. A researcher (Radwan 2000) recently expanded the normal Gantt chart into a dynamic
Gantt chart for maintenance scheduling. The chart is to be used by work crews to accommodate preventive maintenance and breakdowns. Another set of researchers (Davis and Kanet 1997) used color bar charts to develop their production gantt charts. Tasks connected end to end of the same color would not experience a set up cost. One researcher (Abramovici 2000) examined the idea of a “rolling window” technique for any significant size project schedule. He proposes detail planning for 6 to 12 months and milestones for long range.

There are many software applications available to help with the development of the type of schedules generated above. Most graphic and spreadsheet application programs include such a feature as Microsoft Power Point and Microsoft Excel. As the number of tasks increase in the project, a more methodical technique is often sought. As the number of tasks increase, the constraints and dependencies associated with each of these tasks become more important. To accommodate this situation, some type of network is developed. The two most common are the Activity on Node (AON) approach and the Activity on Arc (AOA) approach. There are subtle differences between these two but AON approach is finding greater acceptance because it is easier to program and eliminates the need for “Dummy Activities” sometimes required of the AOA approach to develop a workable schedule. Once this network is developed, the schedule’s critical path(s) is/are found. In summary, the number of tasks used to describe the project will largely determine the type of scheduling techniques used.

2.5 Task Duration

Accurately determining task durations is critical to any schedule development. Some of the techniques developed and their concerns are discussed below. The
techniques will be grouped into how many estimates are obtained per task. The first set of techniques is finding one estimate per task and how best to obtain it. The next set is the three estimates needed to develop PERT (see Section 2.6.2.1) and PNET (see Section 2.6.2.2) schedules will be discussed. Lastly, a few investigators have proposed more than three estimates are needed for each task and they will be discussed. In each of these techniques the accuracy of these estimates is another dimension. Several researchers have addressed the accuracy issue directly. When the number of resources are fixed, the duration of the project is directly related to cost. One researcher (Wang 2002) used a simulation model called COSTCOR for evaluating project costs given correlations among cost items. The scheme takes the top level grandfather level and breaks it down into parents and then children to determine the overall cost risk of the project.

2.5.1 Single Point Estimates

The question is how do you best obtain a single estimate of the task duration? One technique is simply asking the person who is to do the task how long it will take. Without any further instruction to the estimator, a recent researcher (Leach 1999) states that estimators in general believe project managers want a low risk task time so their estimate will be one in which they will have a probability of 80 to 95 percent chance to complete the task within the estimate. Another technique to estimate task duration is to use metrics (i.e. how has this task been taking in the past) if they are available. However, in the CD scheduling problem rarely is this information available. Another is to use a parametric (i.e. the physical size of printed circuit card with a complexity factor) if they are available. Still other techniques are multiple estimates for the same task with a different probability assigned to each estimate. In the CD scheduling problem, task
duration estimated during the hardware software integration phase is especially
challenging since the task to be done probably has never been done before. One
technique (Schmidt and Grossmann 2000) describes task durations as a probability
density function (pdf) which combines Piecewise polynomial segments and Dirac delta
functions. Another set of researchers (Oberlender and Trost 2001) developed an estimate
scoring system of 45 elements. They developed a software application called Estimate
Score Program (ESP) to keep track of the scoring. They tried their technique on 67
projects worth $5.6B to validate their results. The result was that they assert you can
predict the actual cost with high accuracy by determining a project’s accuracy score
developed at the start of the project. Another set of techniques is presented in the article
titled “Want Better Project Estimates? Let’s Get to Work” (Deyoung-Currey and Joan
Knutson 1998). The paper describes a common language, using history and establishing
a repeatable process. A recent case study (Hill, Thomas et al. 2000) was completed with
experts estimating a non-concurrent software development project tasks. The study
showed remarkably accurate results with only a 1 percent under estimation.

2.5.2 Three Point Estimates

Next to be reviewed were those techniques requiring three estimates per task
duration as required by PERT and PNET. For these techniques the three durations are the
most optimistic, the most likely and the most pessimistic. Many texts (Shtub, Bard et al.
1994) give no further definition of the three durations. That is, they simply provide the
three durations without any further instructions on the fidelity of the estimate. What does
the most optimistic and most pessimistic actually mean? Are they end points (the most
optimistic and the most pessimistic duration’s) absolute limits, or are they more than 1
percent and less than 99 percent of the time points respectively, or are they 5 and 95 percent points, or are they something else? The answer can significantly impact the outcome. When PERT is normally used, these points are the absolute end points. In some real life cases they may well be absolute end points but in other cases they could be very extended in the extreme. A second issue with the three PERT durations is the center number. The question asked was what is the most likely duration? This implies mode and that is what the PERT method assumes. The knowledgeable estimator may in fact confuse the difference between median and modal value. The median value may well be what is provided. Lastly, there is an issue of how the data is collected. Does it make a difference in which order the knowledgeable person estimates the three durations? Studies have found that the results are more accurate when the most likely duration is asked for first and then the end points (Selvidge 1980). It seems to have a way of bounding the limits at the start. In summary, the three PERT durations need to be the absolute end points and the center duration to be the mode to obtain the most accurate results. Simply, the knowledgeable person making the estimates needs to understand that this is what is needed when the estimates are made.

2.5.3 Multi-point Estimates

In straight CPM/PDM only one estimate per task duration is required. When PERT and PNET are used three durations are asked of each task. Researchers have been found to question the number the three PERT durations. First, the beta distribution is a four-parameter distribution and one would expect four points would be needed to estimate the four parameters. A set of researchers (AbouRizk, Halpin et al. 1991) studied the distribution functions of construction industry tasks. They initially considered the task to
follow a beta distribution and used a visual means and a software system called VIBES (visual interactive beta estimation system) to estimate the parameters of the distribution. These researchers (AbouRizk, Halpin et al. 1992) further studied the distribution function of construction industry tasks and found after plotting 71 task durations that based on the sample's coefficient of skewness and kurtosis a beta distribution did fit well. This same set of researches (AbouRizk, Harplin et al. 1994) refined their approach once more and found a least-square minimization approach worked best to estimate the parameters for the beta distribution. They used a BetaFit, an interactive, microcomputer-based software package. Another set of researchers (Fente, Schexnayder et al. 2000) also found that construction tasks follow a beta distribution of mode, minimum of 75th percentile and 25th maximum percentile. They presented a method to determine those parameters. The PERT distribution is further analyzed in Section 2.6.2.1. Others have studied taking four estimates and others taking more points. When this is done, one of the first questions is what points do you choose. One set of researchers believe estimators are capable of selecting 7 points such as Lau’s 7 Point Fractile Estimation procedure (Lau and Somarajan 1995). They believe, with experiments to back up his belief, that estimators can be trained to reliably estimate seven points per activity. They recommend the points of 1%, 10%, 25%, 50%, 75%, 90% and 99%. They also found that the three PERT input data points do not do well in estimating a beta distribution. A 7 Point Fractile Estimation procedure is far superior but with considerably more effort required on collecting the input data.
2.6 Networking

Networking is the process of arranging all the project tasks in a logical sequence in accordance with assigned constraints and dependencies and determining how long the project will take or to achieve some project objective. There is a vast array of techniques to accomplish this task and they can often be used in combination. For example, one technique will produce a viable schedule and another technique will find a better solution. These techniques can be grouped a number of different ways. One way is to list all those that develop a schedule as accurately as possible to predict the outcome of the project in one group and list those that optimize the schedule to meet some criteria in the other group. Each group can be further divided. The first group of techniques can be divided into those that account for the task variability and into those that do not. In Section 2.6.1 titled deterministic methods will review the techniques that assume no variability of the tasks. In Section 2.6.2 titled non-deterministic methods a review is made of the techniques that takes into account the variability of the tasks. The second group of optimization techniques can also be divided into two groups. One way is to divide them into those that find an optimal solution and those that find a good solution. An example of the first type is an enumerative approach, which lists all possible solutions. An example of the second type would be a heuristic (an algorithm based on common sense but will not guarantee an optimal solution). The optimal optimization methods are reviewed in Section 2.6.3 and the good or heuristics methods are reviewed in Section 2.6.4. Dividing the networking methods into the four lists mentioned above account for most of the methods, however there are a number of methods that do not neatly fit into
one of these four. A fifth category in Section 2.6.5 titled other is where they are reviewed.

As mentioned above the first two categories of deterministic and non-deterministic methods are primarily designed to produce accurate project schedules that will predict the outcome of the project. In many cases they will also tell you the critical path which is the longest list of tasks through the network but none of these will give you an optimal or even a good solution as the methods reviewed in the other categories attempt to do. In reality most uses of deterministic and non-deterministic methods will want the “best” schedule and will attack that issue something like the following. Once a logical network is developed, the networking effort is often far from over. As is normally the case, the completion date determined by the network analysis is beyond when the product is required. Numerous techniques such as brainstorming are employed to bring the project schedule within the desired delivery date. The result may include “work-arounds” to reordering the way the tasks are networked to adjusting the constraints and dependencies to should the task be done in-house verses contract out to adding more resources if possible to mention some. The schedule is then redone with the modifications made to improve the schedule in a way optimizing the schedule. This entire approach depends on the experience and talent of the scheduler and program team. As will be seen this is the way most project schedules are developed.

Most scheduling optimization problems are extremely hard to solve. They are referred to as NP-hard problems for non-deterministic polynomial time compared to easy problems, which are said to be polynomial time solvable. The idea being that the time to solve NP-hard problem increases exponentially not polynomially as the complexity of the
project increases whereas polynomially time solvable can be solved rather directly. As a result the majority of the techniques are addressing NP-hard scheduling problems. The CD scheduling problem is faced with selecting the best networking technique.

Another way to have been categorizing the optimization algorithms is to say if they are of the constructive type or of the improvement type. That is, the constructive type category builds the schedule. For example, the branch and bound technique (see Section 2.6.3.2) is such a technique and it also attempts to produce an optimal solution. The improvement type takes a completed schedule and seeks to improve the schedule. For example, a local search technique such as the simulated annealing technique (see Section 2.6.4.1) is such a technique. This distinction will be pointed out as the networking methods are reviewed.

2.6.1 Deterministic Methods

As the word deterministic implies these methods do not account for the natural variability of tasks. In most cases deterministic methods are used simply for expediency. As will be shown shortly, deterministic methods are straight forward and easily implemented whereas methods that account for variability require considerably more input data and effort to develop a schedule. However, in many types of projects deterministic methods are more than satisfactory in developing a project schedule that accurately predicts the outcome. For example, projects in which all or most of the tasks have been done many times before and a good set of metrics are available. The construction business would be an example of this type of project. A very popular deterministic method is the Critical Path Method (CPM), which will be described in the next section. The CPM was the first method introduced and since then a number of other
methods have been offered which will also be reviewed. This section will close with a
review of the most recent addition called the Critical Chain Project Management
(CCPM), which has found much appeal in recent years.

2.6.1.1 Critical Path Method (CPM)

A very popular scheduling tool is the CPM. It is straightforward and the critical
path can be determined easily. The CPM came about from a joint venture between
Remington Rand Univac (lead investigator J. Kelly (Kelly 1961)) and the Dupont
Company (lead investigator M. R. Walker) from 1956 to 1959 with their first version
introduced in 1957. Moer, Phillips and Davis (Moder, Phillips et al. 1983) reported that
Walker published his work titled “Project Planning and Scheduling” as Report 6959, E. I.
duPont de Nemours and Co., Wilmington, Delaware, March 1959. Their goal was to
reduce time to perform routine plant overhaul, maintenance and construction work. They
were trying to optimize project duration and project costs. It is a deterministic method
meaning that the tasks are estimated as a fixed length activity. The CPM involves the
following steps: listing all the tasks that need to be done in the project under
consideration, estimating the amount of time it will take to do each task, arranging the
tasks in a logical arrangement, invoking constraints and dependencies on the tasks and
then determine the critical path using the technique of analyzing each task on a forward
pass and then a backward pass. A key feature that distinguishes this technique and PERT
from PD and CCPM is the way activities are diagrammed. The technique used by these
early investigators is referred to as Activity on Arrow (or Arc)(AOA). A simple AOA
network is shown in Figure 2-2.
Here the arrows represent the activities. The nodes represent no effort but only the starting and finish points of activities. The dashed arrow is known as a dummy activity to describe that activity B must be completed before Activity E can start. This technique will determine the longest path through the network or what is called the critical path. That is, if any task on the critical path takes longer that the original estimate, the overall project will be lengthened by that same amount. This technique will also determine the slack time for all tasks not on the critical path. That is, the amount of time the completion of that task can be delayed until it will become a task on the critical path. The CPM is deterministic in that there is no accommodation for the variability of the tasks. For example, stating a task A will take x amount of time plus or minus 10 percent cannot be accommodated with CPM. There are a number of reasons given as to why not add this variability. Probably the biggest reason is that projects often exceed 1000s of tasks and obtaining one estimate per task is a tough enough task not to be adding the burden of collecting additional data for each task. This is particularly challenging in the Concurrent Development (CD) project, which is the focus of my
research. Obtaining this additional information in a more repetitive type project such as a construction project may make sense. However, in a CD project the utility of what was used on the last project is often of marginal value to the current project. The rapid pace of advancing technology makes today’s techniques obsolete tomorrow. CPM has lasted all these years since it is easy to understand and has minimal data input requirements. CPM continues to be studied as evidenced by a recent study (Ari-Pekka Hameri 2002) that uses the time-use within individual tasks time that are on the critical path to ensure project success. The special feature of CPM to use AOA is also being studied today. A recent study (Lin 2002) considered projects with a large number of AOAs. They assumed a fixed duration and fixed budget and that the activities were random integer values. With this they developed an algorithm to generate all longer boundary duration vectors and shorter boundary duration vectors. Project managers can use the resulting information when the project runs into trouble. Branch and bound techniques have been used to find the critical path and is used in a wide range of applications. One recent study (Baber and Mellor 2001) explores multi-modal human-computer interaction using CPM to optimize the results. With the advent of personal computers, the manual chore of the forward and backward pass has been eliminated. Another set of researchers (Nasution 1994) took on the uncertainty of task duration and applied fuzzy numbers to the critical path method. They were able to determine the latest allowable time, the slack for each event as well as the critical path.

2.6.1.2 Precedence Diagram Method (PDM)

It was not until 1959 when the two groups developing CPM and PERT became aware of the other efforts. They were developed independently for different reasons.
However, they were both based on Activity on Arrow (or Arc) (AOA) methodology.

From this PDM was developed that addressed some of the shortcomings of CPM and PERT. It was actually 1958 when Professor J. W. Fondahl of Stanford University first offered up the node scheme to describe activities (Moder, Phillips et al. 1983). However, it wasn’t until 1964 when the node scheme was extended and referred to by J. David Craig as precedence diagramming in the Users Manual for an IBM 1440 computer program. The calculation of early/late and start/finish times for a precedence diagram is the same as an arrow diagram. The technique for forward pass and backward pass to determine these values was actually developed by Professor Keith C. Crandall of the University of California, Berkeley that he published in 1973 (Crandall 1974). The fundamental difference in PDM and PERT/CPM is the way in which the project is diagramed or networked. PDM uses Activity on Node (AON) verses AOA. A simple AON network would look like Figure 2-3.

![Figure 2-3 Activity on Node (AON) Diagram](image)

Here the activities are described on the nodes. The arrows or arcs carry no effort but only show relationships. Information about the activity is often placed in the node box. Common items are the activity number/letter, short title, duration, and dependencies (discussed below), start date, finish date and slack if any. Changing to the AON technique had a number of advantages. First, the only dependency easily allowed with
AOA is Finish to Start (FS). That is, the next activity in the network cannot start until the previous activity has finished. The following dependencies shown in Figure 2-4 are now allowed with AON.

![Diagram of Task Dependencies]

**Figure 2-4 Task Dependencies**

These new dependencies were a great addition since they now allow a project manager to design a network with an improved schedule i.e. reduces the overall length of the project by invoking one or more of these dependencies. In practice using anything other than FS dependency will make analyzing and tracking the progress of the project much more difficult and should not be invoked unless there is a strong need to do so. In the electronics CD projects, these additional dependencies have been frequently employed to help reduce the over project duration but at the cost of losing visibility of the critical path.

In addition to allowing additional dependencies, AON also allows easier statement of constraints. Three common date constraints are the following:

1. No earlier than. This is the earliest date this activity can be completed.
2. No later than. This activity must be completed by this date.

3. On this date. This activity must complete on this exact date.

Another feature of AON is the use of the lag variable. This allows the ability to incorporate a pause in the schedule to wait for something to happen. An example would be allowing time for concrete to cure sufficiently before a load is placed on it.

Still another advantage of AON is there is no requirement for dummy activities to make a viable network. This simply reduces the complexity of the network. Several researchers studied minimizing the number of dummy activities as a way to reduce the complexity of the schedule. One study (Krishnamoorthy and Deo 1979) started with the premise the project completion time is proportional to the number of edges to include dummy activities. They suggested a polynomial time heuristic algorithm to solve the dummy activity problem. The other study (Syslo 1984) developed the polynomial-time algorithm to test if a given activity network requires dummy activities in the event network. PDM and its extra dependency features do on occasion produce anomalous results on slack time and even the critical path (Wiest 1981), which is a sign for some caution when using PDM.

Lastly, the modeling required to implement a schedule using a computer is easier with AON since it lends itself to tables and it doesn’t have dummy activities. Virtually all software applications invoke AON network diagrams, which in essence mean they are using PDM. To confirm this, a survey of the following scheduling applications found that they are all using AON networking:

1. Microsoft® Project 2002 (Microsoft Corporation)

2. Primavera Enterprise® (Primavera Systems, Inc.)
3. Primavera TeamPlay® (Primavera Systems, Inc.)

4. Open Plan® (Welcom)

5. PS 8™ (Scitor Corporation)

In summary, PDM has replaced CPM as a method of choice. PDM does everything that CPM does and more. If the new features were ignored, the results would be the same as with the traditional CPM. CPM uses AOA and PDM uses AON but the end result is the same. A major practical reason for the shift to PDM is that virtually all software scheduling applications are using this method because it is easier to program.

Others have also found PDM useful. The medical professional is using PDM to ensure high quality and cost-efficient health care. A set of researchers (Chu and Cesnik 1998) is attempting to restructure their patient care delivery systems by employing PDM to help solve this problem.

2.6.1.3 Linear Scheduling Model (LSM)

LSM (Johnson 1981) has been around since the 1950’s as an alternative to CPM. It has been used in the construction industry to streamline the complexity of CPM in large projects. Recently a comparison was made with CPM and found LSM still to be very effective (Yamin and Harmelink 2001). However, LSM does not find the critical path, which may explain why it has not achieved wide acceptance. One researcher (Harmelink and Rowings 1998) has added the ability to find a controlling path, which has partly overcome that shortcoming. Since LSM does not have a critical path, it also does not have float, as in CPM/PDM. However, the concept of rate float is more meaningful to construction business and is in sync with the LSM attribute of production rate which is their main attribute (Harmelink 2001).
2.6.1.4 Line of Balance (LOB) Method

Another system called the LOB method (Shtub, Bard et al. 1994) was developed for efforts that have multiple similar products such as an order for a number of ships or buildings. This technique is based on control points or milestones. For example, you might determine to select the following control points for an order of multiple buildings: pour foundation, complete walls and roof, complete interior and complete landscape. Control is then exercised by monitoring progress by keeping track of how many of the buildings pass each milestone. One set of researchers (Suhail and Neale 1994) combined the best merits of CPM and LOB into a composite method. These researchers considered CPM to have difficulty on repetitive projects in changing the order of tasks where LOB had difficulty with projects with a large number of individual tasks. Their composite method selected features of each system that best counter shortcomings. Another set of researchers (Yi, Lee et al. 2002) attacked the idea of network generation on a repetitive project where you can optimize the task linkages. The LOB method does not generate a network, which would be needed to optimize task linkages. These researches developed a model that would. Another set of researchers (Hegazy and Wassef 2001) took on the task of optimizes cost when using the LOB method. They developed a genetic algorithm to optimize the “combination of construction methods, number of crews, and interruptions for each repetitive activity.”

2.6.1.5 Critical Chain Project Management (CCPM)

Dr. Golratt developed the CCPM technique. His first report was in a novel (Goldratt 1984) and later in formal presentation of CCPM in his book titled “Critical Chain” (Golratt 1997). A number of other authors have expanded on this technique to
include (Leach 1999). Another researcher (Steyn 2002) took CCPM from a single project as presented in Dr. Golratt’s work to multiple projects, project cost management and project risk management. CCPM is actually more than just a technique to develop schedules. It is a management tool on how to manage projects. Practitioners of CCPM (Leach 1999) claim projects complete in less than one half the time compared to a schedule developed using the critical path method. One recent user (Hagemann 2001) of CCPM is the NASA, Langley Research Center on a recent wind tunnel test program. They report improvement in performance and employee morale. The acceptance of CCPM as an improved approach to manage project is not universal (Raz, Barnes et al. 2003) but the approach is still relatively new. Another distracter is Professor Trietsch of the University of Auckland who conducted an in depth review and published his findings (Trietsch 2005). He finds shortcomings and makes recommendations for improvement. CCPM is based on the Theory of Constraints (TOC) adapted from managing repetitive production systems. The technique starts as with CPM, then listing all the tasks, arranges the tasks in a logical order add constraints and dependencies and calculate the critical path. Next is to reduce all task times to their 50 percent probability estimate and then find the new critical path length. The difference between this time and the original time is placed in a new task at the end of the project called a project buffer. Next is to look at all chains of tasks leading into a merge point. Those tasks not on the critical path will have slack or float. The technique states you need to add up all the slack of those tasks in that chain and place an additional task in the chain with that length of time. These new tasks are called critical chain feeding buffers. The technique now calls for the project manager to use the buffers as their primary tool to measure and control the project. A
technique given to manage buffers is to first give priority to tasks on the critical path and the resulting project buffer. Next is to take no action as the first third of a buffer is used up. When the project penetrates the middle third of a buffer, the problem is assessed and a plan of action is developed. When the project penetrates the last third, action is initiated. Depending on the project, the buffer penetration is monitored at least monthly but more likely weekly. For CCPM to be a successful, management and team members buying into this technique are critical. Some software vendors who provide scheduling programs are adding features to their programs to accommodate CCPM. This should also help in the acceptance of CCPM. CCPM does require more effort to establish the initial schedule to include the need to determine two task lengths for each task. One is the high assurance completion date and the other is the 50 percent chance of completion date. Obviously, the accuracy of these dates determines the credibility of the resulting schedule. Only time will tell the acceptance of CCPM and its ability to displace CPM as the technique of choice.

2.6.2 Non-deterministic Methods

Accuracy in predicting the project completion date and the path to get there is of vital concern to all program managers. Often project extension means increased costs. Sometimes because of penalties for late delivery and other times, simply the extra cost in maintaining the project core staff for an extended period often referred to as the marching army. An accurate prediction tool is critical not only in the initial project estimation but also in assessing the progress as the project matures. However, accuracy needs to be traded against the time, cost and reasonableness to develop the input data for the prediction tool. It is highly desirable that the prediction tool is not only an accurate
predictor but it needs to be easy to understand and to use. One of first analytic attempts
to estimate project duration was the CPM introduced in 1957 discussed in Section
2.6.1.1. However, it was quickly felt that the probabilistic nature of the activities had to
be taken into account. In 1958, the U.S. Navy Special Projects Office introduced the
Program Evaluation Review Technique (PERT). PERT assumes activities follow a beta
distribution. Simple formulas were developed. They have also become widely accepted.
They are taught at most schools teaching scheduling and widely used in industry. The
next several sections looks at PERT today 44 years later and to see if PERT is a good
method or is there a better method. In Section 2.6.2.1 the PERT method is described
along with its shortcomings. In Section 2.6.2.2 through 2.6.2.7 there is a review of six
other more recent methods, which are studied individually, and then they are compared in
Section 2.6.2.8. In Section 2.6.2.9 there is a discussion on curve fitting collected data. In
particular, how well does the beta distributions do in modeling the PERT three durations
estimates. In Section 2.6.2.10 there is a conclusion on non-deterministic methods.

2.6.2.1 Program Evaluation Review Technique (PERT)

The development of PERT was motivated out of the need to develop the Polaris
missile system in record time. Projects conducted in the early 1950’s have had much to
be desired in achieving cost and schedule performance. The government’s Polaris
Weapons Systems program office took on the challenge. They engaged the prime
contract of the Polaris missile, the Lockheed Aircraft Corporation and a consulting firm,
Booz, Allen and Hamilton to develop a better system. They took the techniques of Line-
of Balance (Turban 1968), Gantt charts and milestone reporting systems and developed
what they first called the Program Evaluation Research Task. By the time of the first
report, the name was changed to Program Evaluation Review Techniques or PERT (Malcolm, Roseboom et al. 1959). PERT added the probabilistic nature of real life activity durations to schedule development. The first step in PERT is to network the project as with CPM. Networking is the process of organizing all the project tasks in a logical sequence, assign constraints and dependencies to each task and determines the critical path. In CPM one number is provided for the duration of each task. However, in PERT three durations for each task are to be provided – the most optimistic, the most likely and the most pessimistic. PERT assumes the tasks are randomly distributed in a beta distribution. Using this assumption, both the expected mean and standard deviation for each activity can be calculated. Using the tasks calculated expected means, the network critical path is found as in CPM. Once the critical path is identified, the standard deviation of the project is determined by adding together the standard deviations calculated for each of the tasks on the critical path.

Step 1. Network the project. List all the activities that make the project into sizes that are intended to be manageable. Determine the interrelationships of the activities. Various techniques have been developed to pictorially display the network (Shtub 1994).

Step 2. For each activity determine (asking a knowledgeable person is one way) the following:

   M  The most likely duration
   a  The most optimistic duration
   b  The most pessimistic duration

Step 3. Calculate the mean, $m$, and the standard deviation, $s$, for each activity using the following equations:
\[ m = \frac{1}{6} (a + 4M + b), \]  

\[ s = \frac{(b-a)}{d}, \]

where \( d \) is a scaling factor. A typical number for construction projects is 3.2 (Moder, Phillips et al. 1983) while 6 is a more general value (Shtub, Bard et al. 1994).

Step 4. Determine the project network’s critical path using the activity means, \( m \)'s, calculated in step 3. The expected project duration is then:

\[ m_T = m_1 + m_2 + \ldots + m_n, \]

where \( n \) is the number of activities on the critical path.

Step 5. Determine the project’s variance using the activity standard deviations, \( s \)'s, calculated in step 3. The project critical path variance is then:

\[ s_T^2 = s_1^2 + s_2^2 + \ldots + s_n^2 \]

Comments. A basic assumption of PERT is that the probabilistic nature of activity duration follows a beta distribution. Many authors have analyzed this assumption. One (Lau and Somarajan 1995) states that the logic is at best marginally defensible only after one recognizes implicit restrictions that are not clearly reasonable. He further says others have implied that equations (1) and (2) are simply illogical and incorrect. Still this approach is widely accepted and taught, most probably because of its simplicity. Putting the beta distribution issue aside, the accuracy of the method most likely depends on a knowledgeable person providing the three durations for each activity. Projects often go into thousands of activities. The task to develop such a set of times can be daunting especially if the activity, or a like one, has not been done before. Even so, if the project is of high value, the effort is often undertaken. One set of researchers (Chen
and Chang 2001) has recently invoked fuzzy logic as a way to address the difficult task of coming up with three precise numbers. Their results produce multiple critical paths. When compared to other methods studied here, PERT gives the most optimistic results as will be shown below, which might also help its popularity.

2.6.2.2 Probabilistic Network Evaluation Technique (PNET)

PNET (Ang 1975) is based heavily on PERT. While PERT considers only the critical path, PNET takes into account more than just the critical path in estimating project duration.

Overall approach. This method analyzes every possible path through the network to select those paths that have the most impact on the overall duration of the project. The selection is accomplished by analyzing the individual path duration variance. This is done by first listing all the possible paths in order of duration from longest to shortest. If two paths have the same duration, the one with the larger variance stays and the one with the smaller variance is eliminated from further consideration. Next, correlation factors are calculated for each pair of paths. Those path pairs that are correlated will then have their variances compared. The one with the larger variance is kept and the other one is eliminated from further consideration. Those path pairs that are not correlated will have both paths kept. The selected paths are then analyzed to determine the predicted length of the program.

Specific steps:

1. Network the project.

2. For each activity calculate estimated duration, m, and the standard deviation, s, as with the PERT method.
3. List all possible paths in order of duration with the longest first.

4. If two or more paths have exactly the same length, select the path with the larger standard deviation. Disregard the other one(s) from further analysis.

5. For every pair of paths calculate its correlation factor, $R_{ij}$, using the following formula,

$$ R_{ij} = \frac{s_1^2 + s_2^2 + \ldots + s_n^2}{\frac{s_i^2}{s_j^2}}, \quad i, j = 1, 2, \ldots, n $$

where $s_1^2, s_2^2, \ldots, s_n^2$ are the variances of activities in common with both paths. For this analysis non-common activities do not add any variance.

6. Assign a 1 if $R_{ij}$ is greater than 0.5 and assign a 0 if $R_{ij}$ is .5 or less.

7. For every pair of paths that have a $R_{ij}$ of 1, select the path that has the larger variance and discard the other. For every pair of paths that have a $R_{ij}$ of 0, keep both paths. The paths selected are then the ones with the most impact on the schedule completion and are further analyzed.

8. Calculate the probability that each path will complete in less time than $T$ by using equation (6). The duration of the path is assumed to be normally distributed based on the Central Limit Theorem. The probability that this path completes in time $T$ is $p(t_i < T)$ and could be found from a standardized normal distribution table. However, equation (6) which was developed by Patel (Patel and Read 1996) is often used. The equation has a maximum error of 0.07%.

$$ p(t_i < T) = \frac{1}{1 + \exp[-1.595769(x)(1 + 0.044715(x^2))]}, \quad 2-6 $$
where \( x = \frac{m_T - T}{s_T} \).

Step 9. Calculate the probability of exceeding a given duration \( T \) using the selected paths as follows:

\[
P(T) = 1 - p(t_1 < T)p(t_2 < T)\ldots p(t_n < T),
\]

where \( n \) equals the number of paths being considered as used above.

Comments. PNET uses the same beta distribution assumption as does PERT. As a result, the basic logic that this represents real world activity durations is suspect. The knowledgeable person input is the same as with PERT so the input data is the same in both methods. However, considerably more effort is required to set up and then calculate the results. As shown by Diaz (1994) the results from using PNET are more pessimistic than with PERT. The real value of PNET is that this method takes into consideration other paths rather than just the critical path in determining the duration of the project. This also explains why the PNET results are more pessimistic than PERT.

2.6.2.3 Narrow Reliability Bounds (NRB) Method

This method was developed by (Ditlevsen 1979) for structural reliability analysis and then applied to scheduling by (Laferriere 1981). NRB is based heavily on PNET. NRB finds upper and lower bounds on the variance found with PNET.

Overall approach. The goal is to determine the lower bound probability (PL) and the upper bound probability (PU) of completing the project in a time longer than \( T \), where \( T \) is at least equal to or longer than the expected time to complete the project. We start out as with PERT by networking and finding the expected duration and standard deviations of each activity. Next, we determine the expected duration and variance of
every path. For the given time $T$, the probability of failure for each path is calculated (i.e., the probability that the duration of that path will exceed time $T$). The paths are then listed in order of failure probabilities from the highest to the lowest. Now, Ditlevsen (1979) determined that the combined probability between paths could be represented as a two dimensional figure with success and failure regions. Further he found the failure region has an upper bound (PU) and a lower bound (PL). He developed a technique to calculate those bounds.

Specific steps.

1. Network the project.

2. For each activity calculate estimated duration, $m$, and the standard deviation, $s$, as was with the PERT method, by using equations (1) and (2).

3. Calculate the estimated duration, $m_T$ and the variance, $s_T^2$ for each path using equations (3) and (4).

4. For each path calculate the probability of failure for time $T$ using equation (6) and then subtracting the results from the number 1. This is the probability that the path will exceed time $T$.

5. Calculate the correlation factors for each pair of paths as with PNET using equation (5).

6. Now two intermediate probabilities ($P_1$ and $P_2$) are found for each pair of paths. The Central Limit Theorem was invoked in developing the following equations:

$$P_1 = \Phi(x_j)\Phi\left[\frac{x_i - R_{ij}(x_j)}{(1 - R_{ij}^2)^{1/2}}\right].$$

2-8
\[ P2 = \Phi(x_j)\Phi\left[ \frac{x_i - R_{ij}(x_j)}{(1 - R_{ij}^2)^{1/2}} \right], \]

where \( \Phi(\bullet) \) is the standard normal distribution function and \( x_i \) and \( x_j \) are the normalized values for each path.

7. Calculate PL and PU by using the following equations:

\[ PL = P(F_1) + \sum_{i=2}^{m} \max\{0, P(F_i) - \sum_{j=1}^{i-1} P(F_i \cap F_j)\}, \]

\[ PU = \sum_{i=1}^{m} P(F_i) - \sum_{i=2}^{m} \max_{j<i} P(F_i \cap F_j), \]

where \( P(F_i) \) is the probability of failure of the \( i \)th path, \( P(F_j) \) the probability of failure of the \( j \)th path, and \( P(F_i \cap F_j) \) is the probability of the intersection of the failure modes \( i \) and \( j \).

Comments. NRB starts with the same beta distribution assumptions as does PERT and PNET. As a result, the basic logic is suspect. The knowledgeable person input is the same as PERT and PNET so that the raw data input is the same. As can be seen this method is more complex to set up and run. With today’s personal computers, the amount of calculations is not much of a consideration. However, the time to set up the network can be. When compared to PNET, the NRB lower bound solution is more optimistic than PNET and the upper bound solution is more pessimistic. This was expected since the method is closely aligned with PNET. When compared to PERT both bound solutions are more pessimistic.
2.6.2.4 Monte Carlo Simulation (MCS) Method

MCSs (Diaz and Hadipriono 1993) have also been used to analyze project durations. Overall approach. The project is networked as before. The network becomes the model for the simulation. Before running the simulation the cumulative distribution function (cdf) of each activity duration needs to be determined. For the methods studied so far in this research, the beta distribution has been assumed. Most any type of distribution could be envisioned for a MCS and implemented. A common MCS distribution is the triangular distribution. It uses the most optimistic, the most likely and the most pessimistic times obtained in the PERT method to determine the shape of the distribution. A typical density function would look like Figure 5 which then could be converted into a cdf for the MCS. Now with the cdf and a random number generator, a duration for each activity can be determined. The duration of each path through the network is determined by adding the individual activity durations. The path with the longest duration is kept. Doing 10,000 iterations is typical for accurate results, although 1000 times will provide satisfactory results (Moder, Phillips et al. 1983). The results are listed in order of duration from the shortest to the longest path. From this the probability of completing in time T can be computed.
Specific steps.

1. Network the project.

2. Determine the cdf for each activity. One way is to ask a knowledgeable person to estimate the most optimistic, the most likely and the most pessimistic durations for each activity as with the PERT method. From this a triangular density can be determined such as is in Figure 1 and in turn the cdf.

3. Use a random number generator and the activity cdf function to generate the duration for each activity. Select the path with the longest duration using CPM.

4. Repeat step 3 10,000 times. List the results in order of duration from shortest to longest.

5. Determine the probability of completing the project in time T as follows:

\[ P = 1 - \frac{n}{N}, \]

where \( n \) is the number of durations that are equal to or less than time T and \( N \) is the total number of durations calculated.
Comments. In the above discussion, a triangular distribution was used for the probability distribution function. No real justification was given that this represents the real world only that it was easy. Its logic is as suspect as the beta distribution. There are many simulation application programs available today that can implement most any distribution that might be felt representative of the case at hand. See section 4 below on curve fitting on using a different distribution. Assuming the quality of the input data is the same, the MCS produces the most pessimistic results so far as will be shown below. This can be explained by realizing all the paths are included in the analysis and can contribute to the overall duration.

2.6.2.5 Simplified Monte Carlo Simulation (SMCS)

This method is very similar to the MCS method (Diaz and Hadipriono 1993). It simplifies calculations with a technique to eliminate activities and paths that have little impact on the outcome of the project.

Overall approach. The method set up is like the MCS. All of the paths are identified and expected durations of each path are determined using the expected durations of each activity. Based on a factor, only those paths that are within a certain percentage of the critical path duration are considered further. The rest are eliminated from further consideration. If the paths remaining do not contain all the activities, those activities are also not considered further. This can dramatically reduce the number of iterations required. Simulation programs are still recommended.

Specific steps.

1. Network the project.
2. Determine the expected duration of each activity.
3. Determine the expected duration of each path.

4. Eliminate those paths that are shorter than $T_{\text{min}}$ determined as follows:

$$T_{\text{min}} = K m_{r},$$

where $K=1$. A typical value for $K$ is $2/3$, which can be adjusted with experience.

5. Complete steps 3 thorough 5 of the MCS method, considering only those activities included in the remaining paths identified in step 4.

Comments. Most of the comments on the MCS method apply here. The main goal of the SMCS method is to reduce the amount of calculations. This is of somewhat dubious value today with the increasing speed of computers. It can however reduce the raw data input required from the knowledgeable person. As mentioned before, projects today can easily go over 1000 activities. When the results of the SMCS method are compared to that by the other methods studied so far, the SMCS produces the most pessimistic results since the shortest paths are eliminated from the analysis. However, it is only slightly more pessimistic than the MCS method.

2.6.2.6 Simulation

There are a variety of other simulation techniques that have been proposed. One such study (Subramanian, Pekny et al. 2001) used simulation to solve a type of projects called R&D Pipeline which is a performance-oriented, resource-constrained, stochastic, discrete-event and dynamic system. They used the computing architecture called SimOpt, which combines mathematical programming and discrete event system simulation to optimize a solution.
2.6.2.7 Perry and Greig Method

The Perry and Greig method (Perry and Greig 1975) based on numerical experiments run by Pearson and Tukey (Pearson and Tukey 1985) to overcome some of the problems that the knowledgeable person may provide in estimating the activity duration.

Overall approach. This method is identical to the PERT method except that slightly different information is asked of the knowledgeable estimator and a different formula is used to calculate the mean. Three durations are still asked for but they are clearly defined as the 5%, 50% and 95% durations.

Specific steps

Step 1. Network the project.

Step 2. Ask the knowledgeable person to provide the following three activity duration estimates: a 5% probability or less of completing this activity ($T_{0.05}$), a 50% probability or less of completing this activity ($T_{0.5}$) and a 95% probability or less of completing this activity ($T_{0.95}$)

Step 3. Use the following equations developed by Perry and Greig to determine the mean, m, and stand deviation, s:

\[
m = T_{0.5} + 0.185(T_{0.05} + T_{0.95} - 2T_{0.5})
\]

\[
s = (T_{0.95} - T_{0.05})/3.25
\]

Step 4. Follow PERT steps 4 and 5.

Comments. Studies by Pearson and Tukey (Pearson and Tukey 1985) showed that equations (14) and (15) are accurate for all bell shaped normal distributions where the
beta distribution is good for only a small subset. As will be discussed in Section 2.6.2.9 on curve fitting there are many more distributions for that this does not hold just like with the beta distribution. This method does have the advantage of precisely describing the durations that the knowledge person is to provide. It has a disadvantage in that it does still requires three duration times per task to collected as does PERT. In summary, the method requires almost the same amount of effort as does PERT but with the advantage that the method will accurately characterize more likely activity durations. No studies were found that compared this Perry and Greig method with the others discussed earlier, but it is likely the optimism or pessimism would mirror the PERT method.

2.6.2.8 Comparison of Non-deterministic Methods

All six methods started with networking the project (i.e., breaking of the project into a manageable number of definable activities and then determining the interrelationships or precedence) to determine the overall project duration. The result is a network that represents the project. This is also what the CPM does. Each of the methods gives a methodology to determine the project duration. The NPET and NRB methods attempt to improve on the basic PERT approach by looking at more than just the critical path. They accept the assumption that the activity durations occur as a beta distribution and from the “three PERT” durations you can determine the key parameters of the distribution (i.e., the mean and standard deviation. The MCS and SMCS methods are not restricted to using a beta distribution and in fact rarely do. A triangular distribution is often used but this assumption can also lead to significant errors. The Perry and Greig method used equations developed from numerical experimentation, and covers more distributions than PERT but large errors can occur for other distributions.
Putting aside the issue of the activities are accurately modeled, the question is how do the methods compare to each other? Dias (1993) ran the first five methods against 31 different types of projects that ranged from 15 to 400 activities. The resulting durations varied greatly. It was not unusual to have the results more than ten percent apart among the five methods. Table 2-1 summarizes those results and attempts to compare the 6 methods.
<table>
<thead>
<tr>
<th>Method</th>
<th>Typical Input Distribution</th>
<th>Distribution</th>
<th>Results Compared to PERT</th>
<th>Ease of Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERT</td>
<td>Beta distribution</td>
<td>Most optimistic, most likely and most pessimistic</td>
<td>Most optimistic results</td>
<td>Easy</td>
</tr>
<tr>
<td>PNET</td>
<td>Beta distribution</td>
<td>Most optimistic, most likely and most pessimistic</td>
<td>More pessimistic than PERT</td>
<td>Fairly complicated</td>
</tr>
<tr>
<td>NRB</td>
<td>Beta distribution</td>
<td>Most optimistic, most likely and most pessimistic</td>
<td>Lower bound more pessimistic than PNET and the upper bound more optimistic than PNET</td>
<td>Most complicated to set up</td>
</tr>
<tr>
<td>MCS</td>
<td>Triangular distribution</td>
<td>Most optimistic, most likely and most pessimistic</td>
<td>More pessimistic than NRB</td>
<td>Straight forward</td>
</tr>
<tr>
<td>SMCS</td>
<td>Triangular distribution</td>
<td>Most optimistic, most likely and most pessimistic</td>
<td>Most pessimistic results</td>
<td>Slightly more involved than MCS but runs faster</td>
</tr>
<tr>
<td>Perry and Grieg</td>
<td>Developed from numerical experiments</td>
<td>T(0.05), T(0.5) and T(0.95)</td>
<td>Expected to be the same as PERT</td>
<td>Easy</td>
</tr>
</tbody>
</table>
The next two subsections will address the issue on how well the tasks are modeled and what data is collected to ensure the task is adequately modeled.

### 2.6.2.9 Curve Fitting

A key question in each of the methods in Sections 2.6.2.1 through 2.6.2.7 was what distribution to use for an activity? This section takes a closer look at the answers to that question. The question has several elements. The task at hand is to describe the duration of the activity in mathematical terms. Rarely will we know the distribution function of the activity, if indeed there is one. This being the case, you might select a general type distribution and try to curve fit with your estimated durations. This is what PERT, PNET and NRB do. PERT selected the beta distribution and PNET and NRB followed. A beta function is a four-parameter distribution. The four parameters are the two end points, the skewness and the kurtosis. The beta density function used in PERT is as follows:

$$f_b(x) = \frac{(x - U)^{p-1}(V - x)^{q-1}}{B(p, q)(V - U)^{p+q-1}}, \quad U=x=V, \quad p>0, \quad q>0,$$

where $B(p,q)$ is the beta function evaluated at (skewness $(p)$, kurtosis $(q)$) and $U$ and $V$ are the end points.

The idea is that this distribution would model real activity duration distribution. That is, these four parameters would be sufficient to describe every type of activity duration distribution. With this assumption made and with some other simplifying assumptions made about the beta distribution, equations (1) and (2) were developed. Over the years many have looked at how well the beta distribution really does at modeling real world activity duration distributions. The research has found that the beta distribution does not
do a very good job at representing most distributions and some have implied that
equations (1) and (2) are simply illogical and incorrect (Lau and Somarajan 1995). K.
Pearson’s (Pearson and Tukey 1985) classic work on four parameter distributions first
stated the fundamental principles for modeling distributions with his “Pearson system of
distributions.” In 1967 Hahn and Shapiro (Hahn and Shapiro 1967) came up with their
“skewness-kurtosis diagram.” All known distributions could be plotted on this diagram.
The beta distribution produces a band across the diagram. However, very few other
distributions fall within that band. Lau (Lau and Somarajan 1995) concluded the beta
distribution is not capable of modeling most distributions. It does well for a very small
subset of normal distributions but little else. Lau looked at a variety of other
distributions. None were all encompassing. He looked at the Ramberg-Schmeiser (RS)
(Ramberg and Schmeiser 1974) distribution developed in 1974 and found that it covers
the area not covered by the beta distribution. A general form of the distribution is given
in equation (17). When plotted on the Shapiro “skewness-kurtosis” diagram, the RS
distribution does a good job of complimenting

\[ R(p) = a + \left[ p^r - (1 - p)^d \right] b, \ 0=p=1, \]

where \( R(p) \) is an inverse cdf with parameters \((e,f,g,h)\) of a beta distribution. The two
distributions overlap slightly but the RS distribution covers the area not covered by the
beta distribution. He recommended that the RS distribution along with the beta
distribution should be used. With these two distributions all possibilities would be
covered. Since he was not able to find a selection criteria ahead of time, he recommends
that both distributions be tried and select the one with the lowest root of mean of
variation or in words, the one with the best fit.
In summary, research has shown that the beta distribution is not very suitable to represent activity durations except for a very small set of normal distributions. With both the beta and RS distributions, all activity duration distributions can be represented. It is important to note that what this is saying is if you know the type of activity duration distribution you can adequately represent it as either a beta or RS distribution. However, in many cases, the type of distribution of an activity duration is not known. It fact, it might not follow any distribution. In a long running production facility, the distribution may be well known. In this case, the beta and RS combination would be a good tool to use for estimation. It would be superior to using just the beta distribution.

2.6.2.10 Non-deterministic Conclusions

PERT is coming up on its 44th birthday. Many studies have been conducted and much has been written. One set of researchers (Lu and AbouRizk 2000) has developed a PERT simulation model, which provides for simplified critical activity identification method. There is much about PERT that researchers have found they do not like. Much is centered around challenging the assumption that real world activity durations follow a beta distribution. Many researchers have found that the beta distribution cover only very few conditions accurately. Lau’s suggestion that the two distributions be used is a good one. However, the type of distribution the activity duration follows is probably unknown which is required if you want to know what to select - the beta or RS distribution. Much is also written on the number and type of input data required of each activity. Obtaining three durations per activity is taxing enough not to mentioning attempting to require seven durations per activity as suggested by Lau. In summary, most ideas to remedy the shortcomings of PERT are complex. PERT is simple and straightforward. This research
finds the method offered by Perry and Greig as simple as and easy as PERT but is an improvement. The Perry Greig method is recommended for use in place of PERT. In reality, PERT is totally ingrained and unless something markedly superior comes along, it probably will not be replaced. Many forces operate on a project and are difficult to model. Such issues are labor issues, money issues and organizational issues. Any one of these can overwhelm the factors discussed in this chapter.

The importance of accurately predicting the outcome of a project continues to be of paramount importance. The study will by necessity continue. Newer techniques and faster computers offer hope. Better prediction methods will be developed. Until some clearly superior method is developed, it looks like PERT will be around for some time to come. To have an effective tool, you not only need a sound technique, you need credible input data. There is hope on both fronts.

2.6.3 Optimal Networking

The first two groups of deterministic and non-deterministic methods were concerned with producing accurate project schedule. The next methods to be reviewed are motivated to produce the optimal schedules. As mentioned above most project schedule are NP-hard which leads most often to heuristic methods to be reviewed in the next schedule. There are however several methods available to obtain an optimal solutions which are discussed below.

2.6.3.1 Enumerative

This method simple says list all possible combinations of the schedule and select the one that optimizes whatever you are trying to optimize. This quickly becomes
overwhelming with any real world project due the size of the number of task and interrelationships.

2.6.3.2 Branch and Bound

The branch and bound technique has the advantage of preventing a local optimization solution and in effect seeks the optimal solution. A recent study (Yan, Wang et al. 2002) using this technique considered the concurrent paths of upstream development design on the downstream process design. They developed a heuristic approach with which they controlled the amount of resources allocated to each phase to improve the efficiency. Another set of researchers (Balasubramanian and Grossmann 2002) used branch and bound techniques to fine the lower bound to the expected makespan by evaluating the over an aggregated probability model.

2.6.4 Heuristic Search Methods

Recently, others (Trietsch 2005) have suggested using such techniques in planning and scheduling.

2.6.4.1 Simulated Annealing (SA)

The SA method is a local search procedure of the improvement type. That is you start out with a complete schedule and try to improve it. This method and the next two of Tabu Search and Genetic Algorithms are well known procedures. Unique to SA the local or neighborhood search is done in a random order. One set of researchers (Gemmill and Tsai 1997) found a simple application of SA that optimizes many sampled project schedules. One recent application (Kuo, Liu et al. 2001) using SA was the optimization
of the irrigation of crops to improve profit. Another (Satake, Morikawa et al. 1999) used SA to minimize the makespan in a job shop.

2.6.4.2 Tabu Search (TS)

The TS method is also a local search procedure of the improvement type. That also starts out with a complete schedule and tries to improve it. The TS technique is very similar to SA technique except it performs its local or neighborhood search is done in a non-random order. A set of researchers (Calhoun, Deckro et al. 2002) took on the task of re-planning and re-scheduling of both project and production setting. They used a Tabu search technique to develop multiple options to update the schedule. They developed their technique in Java to be portable. Another researcher (Tiourine 1999) used a tabu search algorithm to minimize the maximum lateness in a job shop. Another set of researchers (Mazzola, Neebe et al. 1998) took on the problem of multiproduct production planning problem in the presence of work-force learning (MPPL) and used the tabu search technique to obtain good results.

2.6.4.3 Genetic Algorithms (GA)

The GA is perhaps the most well know of the modern heuristics with its origin inspired by population genetics (Rayward-Smith, Osman et al. 1996). The GA method is also a local search procedure of the improvement type. That is, you again start out with a complete schedule and try to improve it. Unique to GA the local or neighborhood search produces multiple new schedules simultaneously using pieces of them to build an ever better schedule. Much has been written using GA to solve scheduling optimization problems. A recent study (Leu and Hung. 2002) proposed a new optimal resource-constrained construction scheduling simulation model using the GA technique that
accounts for duration uncertainty due to such things as weather and resource constraints. They assert their techniques obtain the optimal project duration under resource constraints. Another researcher (Soria 2001) recorded in his dissertation a genetic algorithm he developed to solve a resource constrained project schedule.

2.6.4.4 Fuzzy Logic

Fuzzy set theory deals with the fuzziness arising from humanistic cognitive attributes such as perception. Such concepts as hard and very hard are accommodated. A fuzzy set is a generalized set to which tasks can belong with various degrees of memberships over the interval (0,1). Where 1 means full membership and 0 means full non-membership (Gupta 2002). A recent study (Kumar and Ganesh 1999) was able to invoke fuzzy logic to find optimal schedules in a resource constraint environment. Another recent study (Chen and Chang 2001) invokes fuzzy logic to improve PERT. Another set of researchers (Nasution 1994) was able to generalize the critical path method by accepting imprecise, fuzzy data for the duration of the activities. They based their approach in part on the observation that only the nonnegative part of the fuzzy numbers can have physical interpretation. Lastly, recently a researcher (Liberatore 2002) described an enumerative approach using fuzzy logic to solve a project schedule as an alternative to PERT and Monte Carlo simulation to account for activity uncertainty. The researcher makes the point that fuzzy logic does not assume randomness, as does PERT and Monte Carlo simulation but is concerned with activity ambiguity. Fuzzy logic may be more appropriate when little is known of the activity duration.
2.6.4.5 Petri Nets

Petri nets were developed from the doctoral dissertation of Carl Adam Petri, 
*Kommunikation mit Automaten* (Communication with Automata), at the University of 
Bonn during the early 1960s. A Petri net is an abstract formal model of information flow, 
which uses a graphical language for modeling the interacting components. Models 
implemented with Petri nets consist of independent components that may interact with 
each other in synchronized manner or may carry out their activities simultaneously with 
other components of the system (Mata-Toledo 2002). A recent study (Kumar and Ganesh 
1999) was able to invoke Petri nets to find optimal schedules in a resource constraint 
project.

2.6.4.6 Neural Networks

Neural networks were inspired by studies of the structure and function of the 
human brain. A neural network contains a large number of simple nonlinear processing 
modules, which are connected by elements that have information storage, and 
programming functions. This contrasts with modern computers that typically have one 
complicated linear processing module (DeClaris 2002). A recent study (Siqueira 1999) 
of steel building construction developed an automated cost estimating (ACE) system 
based on neural networks to capitalize on learning gained on previously constructed 
building projects.

2.6.4.7 Analytical Hierarchy Process (AHP)

The AHP is a framework for solving problems, which is a systematic procedure 
for representing the elements or tasks of any problem or project. The AHP organizes the
basic rationality by breaking down a project into its smaller task and then calls for only simple pairwise comparison judgments, to develop priorities in each level. The AHP allows the incorporation of intuitive, rational, and irrational factors in making judgments (Saaty 2000). A set of researchers (Mian and Dai 1999) developed an AHP to form a set of goals, criteria, and alternatives for pairwise comparisons of a project using a commercial Expert Choice software application.

2.6.4.8 Artificial Intelligence

The topic of artificial intelligence is a vast one and is finding applications in project scheduling. A class of artificial intelligence programs called expert systems attempt to exhibit the equivalent performance of humans by acquiring the same knowledge that human experts have is finding application (Buchanan and Newell 2000). A set of researchers (Herroelen and Reyck 1999) used the idea that artificial intelligence has shown that many NP-complete problems exhibit phase transitions. That is, problems can change from NP-complete to easy when certain of their characteristics are modified. Sometimes the transition is sharp and other times it is rather continuous. An interesting result was resources often exhibited a rather sharp transition from hard to easy to hard.

2.6.4.9 Dijkstra’s Algorithm

Dijkstra’s algorithm came out of graph theory. If the points of a graph represent cities and the lines between them are labeled with their distances, Dijkstra’s algorithm is an efficient tool to find the shortest path from one point to another (Gross 2002). A recent study (Adelson-Velsky and Levner 2002) used Dijkstra’s algorithm to minimize project duration.
2.6.5 Other Methods

This last category of methods includes all those methods to produce project schedules that do not neatly fit into one of the previous four.

2.6.5.1 Project Decomposition

One recent study (Sprecher 2002) took the approach to decompose the projects into subprojects and optimize the subprojects. Good results were achieved compared to techniques optimizing the entire project. Another set of researchers that decomposed the project into subprojects was (Schmidt and Grossmann 2000). They considered individual tasks as probability distribution functions (p.d.f.) and combined them to form a cumulative distribution function (c.d.f.) using a graph reduction technique. To help the computational process they divided the project into subprojects.

2.6.5.2 Material Resource Planning (MRP)

The techniques discussed thus far have had a goal of developing a schedule that would minimize the overall length of the project or makespan. However, there is a large set of techniques that just want to track the schedule with no attempt to minimize the overall plan. The whole set of MRP techniques are in this category, which are often used in factory situations. In this case the number of projects are often large with the resulting number of activities immense. Simply keeping track of all the activities is a challenge itself. This does not mean these techniques are simply or even easy to use. These programs sometimes include the time tracking of individual workers for wage determination and also the tracking of materials. These programs will often have many entry points to allow data entry as the various steps of the schedule are accomplished.
The use of bar coding (wanding) is a common technique to not only track the product but also the person performing the task. Often mainframe computers are employed to host these programs due to their size and complexity. No examples were found where MRP was used on a CD project. Since the focus of this study is on the CD scheduling problem, these techniques will not be discussed further.

### 2.6.5.3 Fast-Tracking

Aggressive scheduling or fast tracking scheduling is increasing common particularly in the construction industry. Because of tight budgets, tight schedules and the desire by clients to start operations quickly, fast tracking has become common. However, the concern is the impact of change. A set of researchers (Ibbs, Stephanie A Lee et al. 1998) tried three different fast-tracking techniques to shorten construction schedules. The three techniques were as follows:

1. The construction start is scheduled more and more aggressively and the total cost of design and construction change cost increases significantly compared to the original baseline.

2. The construction start is scheduled more and more aggressively and the total cost of construction change cost increases significantly compared to the original baseline.

3. The construction start is scheduled more and more aggressively and the total cost of design change cost increases significantly compared to the original baseline.

Contrary to expectations, the accelerated schedules did not cost more.

### 2.6.5.4 “S” Curves

One researcher (Murmis 1997) used “S” curves to estimate and track the progress of water distribution and sewage collection in Buenos Aires, Argentina. This researcher
was concerned with not only developing a realistic schedule, he wanted a tool to track progress. He assumed a normal distribution of tasks and developed a set of “S” accumulated progress curves with normal distribution forces to pass fixed points. One set of curves was for tasks that used 10 percent of the time with 5 percent of the progress or what he calls 10T/5P. He tried his “S” curves on 30 projects and when projects deviated much his theoretical curves the project was in deep trouble.

2.6.5.5 Learning Curves

Much has been written on learning curves with most textbooks (Shtub, Bard et al. 1994) on project management covering the subject. The basic idea is whenever you double the amount of times you do a task you will reduce the amount of time it will take to do that task by a fixed percentage or learning factor. A recent study (Amor and Teplitz 1998) applied learning curves to a project schedule. They used approximation methods to greatly reduce the cost of implement pure learning curves to each activity.

2.6.5.6 Queuing Theory

Scheduling multiple projects are particularly challenges. A recent study (Levy and Globerson 1997) invoked queuing theory to cope with a scheduling problem.

2.6.5.7 Look-Ahead Techniques

The Look-Ahead technique starts with forward and backward passes in CPM/PERT and then optimizes with a variety of heuristics to choose from depending on the problem to be solved (Gemmill and Edwards 1999).
2.6.5.8 Manfred’s Distributions

A recent study (Carbno 1999) analyzed both single IT projects and multiple projects within an IT operation. The basic premise was to use approximate means to develop the delay cost curve. Numerous methods of approximating were given to include Manfred distributions to analyze and the impacts of project overruns in situations where critical dates are important.

2.6.6 Networking Conclusions

The number of techniques developed for networking is vast. These techniques have been used with varying degrees of success. Many have only a limited applicability to a small subset of projects. The fact there are so many techniques gives testimony that there is a strong need to develop schedules that actually minimize the overall schedule, accurately predict the outcome and are easy to use. The focus of this study is on the CD scheduling problem. Only a few of the above techniques have found their way to develop schedules for the CD project. By far the most common is the CPM/PDM and that has been in use for decades. A new variant of the CPM/PDM called the CCPM has come on the scene and has been gathering a following. However, pure CPM/PDM is so prevalent it seems unlikely that CPM/PDM will be displaced any time soon.

2.7 Resource Loading

Resource loading is the process of adjusting the networked schedule to account for the amount of resources available. Resources are frequently limited and the initial network will have numerous resources over subscribed. Performance is of primary concern but if the product cannot be delivered on time (schedule) and within the
negotiated cost, the program may well be canceled. With the cancellation, the suppliers will most likely suffer loss of good reputation, see reduced opportunities for future business and may suffer financial loss. A company that does resource loading better will greatly improve its chances for increased market share. Again, numerous techniques have been offered as how to do this resource loading. In the Pollack-Johnson and Liberatore study (Pollack-Johnson and Liberatore 1998) the median number of resources considered is 16. Invariably one or more resources will be critical (i.e. a key person or group will be required more than 100 percent of the time). Again, a number of techniques have been used. First, if the task has slack, the slack is allowed to be consumed. Obtaining resources from other activities within the company may be sought as well as considering contracting the effort out. Contracting out is also often considered. Project “crashing” may be employed where additional resources are added at increasing cost. A recent study (Erenguc, Ahn et al. 2001) has investigated finding an optimal crashing solution using a branch and bound technique. Traditional crashing can be a dynamic process but a feasible schedule must be found for the project to be considered viable. Again, a feasible resource loaded schedule must be found for the project to be considered viable. A paper titled “Automatic Resource Assignment Using Suitability” Narimatsu, by K., Tanaka, T. and Araki, D. of the Toshiba Corporation give a suggestion. These researchers have developed a practical method and algorithm to solve the problem of assigning resources. Another paper titled “Multiple Projects, Limited Resources: Implementing Effective Project Management” written by Alston, B. of 3M Health Information System give more ideas. This presenter shows how this process can be simplified. Lastly, a paper titled “Resource Planning and Management” by Howard, P. of
San Francisco, CA shows how to develop a resource plan and track resource commitment.

2.7.1 Issues with Resource Loading Today

The heart of the issue with scheduling and resource loading of projects is that scheduling is basically a one-dimension activity. That is, tasks are planned for a specific period of time. However, tasks are at least two-dimensional quantities. That is, the amount of time a task will take depends on the amount of some resources that are allocated to the task. For example, one ditch digger can dig a ditch in \( x \) amount of time where two painters could most likely paint a room in half the time. There are other dimensions too. For example, rarely are two resources equal. There may be a synergistic factor to increase their efficiency or there may be an inefficiency factor to reduce their efficiency. For example, that room may be in a confined place and painter will interfere with one another making them less efficient. On the other hand, two painters could move heavy scaffolding where one could not. A common technique to resource loading a project is to schedule a project assuming each task is performed by the known amount of resource available. The schedule is then developed that meets all the constraints and dependencies established. However, the resulting critical path may not meet the program needs. For example, delivery by a certain date or meeting a specific milestone at a certain date may be required. The task of scheduling is to meet all constraints and dependencies so the search will go on to find a schedule that meets the project requirements. One area to be examined is to look at each task to determine if any can be shorten if more resources were applied to them - or increase resource loading. Some tasks may not be possible to shorten. For example, the contract requires the concrete to
cure 7 days before a load is placed on it and placing a load on the concrete is the next task. If a task can be shorten by adding additional resources, a technique called crashing is sometimes used which determines the most cost effective way to add resources (see Section 2.7.2.2 below). Cost is always a consideration and the crashing technique that minimizes the extra cost is an important tool. This technique analyzes all possible combinations by adding resources to the critical path tasks a resource at a time until the resource is exhausted or the overall project schedule is met. As each additional resource is added, the project is “rescheduled” to determine if the new critical path meets the project’s need. If it does, the cost of the additional resources is recorded. The process is repeated for the other tasks on the critical path. When all combinations are analyzed, the lowest cost combination is selected. An important item to be noted is resources are always limited and particular your key assets. One researcher (Bigelow 2001) advocates taking special care of your key resources to ensure they do not leave for other opportunities. Before leaving this topic, there are other reasons for adding resources as discussed next.

2.7.2 Resource Loading Objectives

Resource loading may have different objectives. One set researchers (Vanhoucke, Demeulemeester et al. 2002) define three other objectives. They are the deadline problem where you try to minimize cost to meet a deadline. Second is the budget problem where you try to minimize the duration without exceeding a budget. Lastly, the efficient time/cost profile where to try to find an efficient profile over a set of project durations. One objective that has seen extensive investigation is to maximize the net present value (NVP), which is covered in the next section.
2.7.2.1 Maximizing Net Present Value (NPV)

A project needing to maximize net present value would be one where payment is received based on accomplishing work packages. The idea is to move high value work packages as forward as possible so the payment could be used for other revenue producing ventures. This may well produce a schedule that takes longer than one being optimized to minimized project duration. A set of researchers (Smith-Daniels and Smith-Daniels 1987) found that if you consider material management factors in a NPV evaluation, lower project costs would be realized. Another set of researchers (Reyck and Herroelen 1998) found an optimal solution by using a depth-first branch-and-bound algorithm. Another set of researchers (Vanhoucke, Demeulemeester et al. 2001) used an exact recursive algorithm to find an optimal solution. Another set of researchers (Herroelen, Dommelen et al. 1997) conducted a survey of recent NPV methods and critically reviewed the major contributions of both deterministic and stochastic network models.

2.7.2.2 Minimizing Project Duration

The traditional method to minimize a project’s duration or makespan is to use a technique called crashing. The method involves analyzing a network of tasks to determine if adding resources if available can shorten the overall project duration. Many text books (Shtub, Bard et al. 1994; Wysocki, Jr. et al. 2000) on project management discuss this technique. A number of researchers have studied this technique for improvement. One set of researchers (Deckro and Hebert 1989) used several models to include the Prisker, Walters and Wolfe’s model (1969) and the other the Bowan’s model
(1959) to crash a resource constrained project. Another set of researchers (Ahna and Erengucb 1998) used a multi-pass heuristic procedure to crash the project.

2.7.3 Resource Constrained Project Scheduling

Up to this point the amount of resource has not been a consideration. However, rarely is this the case. Considerable study has been devoted to the Resource Constraint Project Schedule Problem (RCPSP). One recent study uses the Ant Colony Optimization (ACO) approach to set of standard project schedules (Merkle, Middendorf et al. 2002) and found they could find optimal solutions to some schedules that were previously unknown. The authors used the evaluation methods of ants finding new solutions and an elitist ant forgetting the best-found solution. They compared their approach to genetic algorithms, simulated annealing, tabu search, and different sampling methods and found their ACO produced better results on average. Another recent study took 135 standard projects using 6 heuristics and compared their performance to ten selected summary measures: five resource-related measures, four time-related measures, and one-shape measure (Abbasi and Haddadin 2002). Still another develops a new heuristic to solve the RCPSP. These authors used a constraint satisfaction problem solving (CSP) search procedure to resolve conflicts by incrementally removing conflicts of the least constraint against a non-deterministic choice heuristic (Cesta, Oddi et al. 2002). Another set of researchers (Mattila and Abraham 1998) used integer linear programming to level resources in the LSM scheduling problem. Another set of researchers (Slowinski, Soniewicki et al. 1994) used a decision support system (DSS) for multi-objective projects with multiple resource constraints. They used a variety of techniques from parallel priority rules, simulated annealing and branch-and-bound techniques. There was also a
survey (Herroelen, Reyck et al. 1998) conducted of recent resource constrained project scheduling methods with particular attention to depth-first branch-and-bound procedures.

2.7.4 Resource Loading Summary

The very best schedule can be developed but if resources are not loaded in a realistic manner, the schedule is doomed to predict the actual outcome. Resources loading techniques will be different depending on the overall goal. Several goals were discussed with those being minimizing the overall schedule length and maximizing net present value. Resource loading is no easy task. Particularly in the CD project task, resource loading can be very dynamic and really needs to be updated frequently in particular for key resources. The project manager is never willing to pay for a resource until it is needed but on the other hand he wants the resource when it is needed. The key resources are the most challenging since they often are down to an individual. That is, there is only one person within the organization that is an expert in the task being considered and virtually no one else will do. Care indeed needs to be exercised to ensure that a key resource is not overloaded particularly when that resource is needed for your task. It is not unusual to see a key resource loaded at 200 or 300 percent if a careful resource loading is accomplished. Resource techniques are critical in these situations to warn the project manager of the pending conflict in advance. This will allow some time for the project manager to work the priorities of this critical resource with the supervisor of that critical resource and work out a plan that accommodates all the claimants on that resource. If this key resource is a continually being over subscribed, resource loading techniques will warn management in advance that longer range solutions are needed such as contract help or hiring additional persons with that critical talent. Resource loading is
often over looked but if accomplished diligently, will sound the alarm earlier when action can be taken before the situation becomes a crisis.

**2.8 Progress Tracking**

Progress tracking is the process of assessing where the project is against the schedule as the project progresses. A schedule is a plan for the future based on a set of assumptions about the unknown. Assumptions can be wrong and the unexpected happens. The first step is realizing there is a problem as soon as the problem occurs. The schedule can be a great aid if a process is in place to frequently and accurately assess progress against the developed schedule. Many items need to be in place for this to happen. First, the schedule must accurately describe the project at hand. Second, the infrastructure to include the culture needs to be in place to accurately capture status. Third, the ability to update the schedule to account for the new events needs to be in place. Using an Earned Value Management System (EVMS) along with the schedule provides excellent insight into the health of the project. As noted in the review of the commercial software scheduling application, each has a wide variety of ways to present the results.

The EVMS is widely used particularly those dealing with the US government since EVMS is required on many contracts. Not only is much written on how to use EVMS (Taylor 1998; Project Management Institute 2000), research continues to be conducted on its use. One of the fundamental building blocks of EVMS is the schedule. If the schedule does not represent the project or when the actual progress deviates markedly from the schedule, the results produced by EVMS is very suspect. One recent set of
researchers (Kauffmann, Keating et al. 2002) conducted investigation on how to use EVMS to substantiate a claim for scope increase.

One unique way of tracking progress was discussed in Section 2.6.5.4 above titled “S” curves. Here the researcher (Murmis 1997) developed a set of curves to track progress. When the performance deviated markedly from the curves the project was in trouble.

2.9 Commercial Software

Commercial software scheduling packages have gained great acceptance in the five years from 1993 to 1998 by going from 65 percent use to over 90 percent as found in a Pollack-Johnson & Liberatore survey (Pollack-Johnson and Liberatore 1998). This survey was comprehensive and covered a wide range of industries. They noted that in the PM Networks September 1996 issue a total of 63 packages were listed. This is a staggering array of choices with Microsoft’s Project the most used package being used at least once by over 70 percent of the Project Managers surveyed and almost 50 percent of the time the most frequently used package according to the respondents to the above Pollack-Johnson & Liberatore survey. Primavera Project Planner (P3) was second and Timeline came in third. All the remaining packages were used less that 5 percent of the time. The survey also revealed several other factors that could impact on this study. One is almost 80 percent of those using a commercial software scheduling package said they would rate that package as acceptable or better. Another factor was almost 75 percent of the users of commercial software packages enter activity resources while almost 50 percent update their schedules periodically with data as it changes. All this is somewhat surprising considering more than 66 percent of the projects are considered to be in trouble. The whole idea of scheduling out a project is to develop a schedule to monitor
progress, make adjustments as necessary and complete scheduled tasks on time. In summary, the use of project scheduling commercial software packages is well accepted. However, accomplishing that project as scheduled is another matter. In the following sections the most popular commercial scheduling software packages are individually reviewed.

2.9.1 Microsoft® Project 2002 (Microsoft Corporation)

Microsoft’s (http://www.microsoft.com/) project applications are by far the most popular applications in the world. For 2002 there are two basic sets of project software applications offered. First, there is Microsoft Project Standard 2002 for individual business and project managers that will run on a personal computer. The other is Microsoft Solution for Enterprise Project Management for medium and large businesses. This base for this application is Microsoft Project Professional 2002 supported with Microsoft Project Server 2002 which adds web access capability. The packages are designed for a very wide set of applications. Also contributing to its popularity is its relatively low cost compared to the other top applications. Microsoft’s June 2002 web page shows Microsoft Project Standard sells for $599 and Microsoft Project Professional for $999. There are some very low cost alternatives available by very small companies but only time will tell if they are accepted. In the last several years Microsoft has been updating their applications every two years i.e. Project 98 to Project 2000 and now Project 2002. A primary area for enhancement has been web basing entry and reporting. The packages have become very sophisticated in the presentation of the results. However, the basic development of the schedule remains the same. That is, the total number of tasks for the project needs to be identified. Next an estimation of the task
duration is made followed by placing the tasks in a logical sequence to form a network. Constraints and dependencies are then assigned and finally, the critical path is determined by using the CPM. Microsoft Project does allow the entry of three points for each task to run a PERT schedule. All indications are the three points are used to form a Beta distribution for each task. In Section 3.5.2.2 the PERT technique is described which is run in the Microsoft program.

2.9.2 Primavera Enterprise® (Primavera Systems, Inc.)

For the past 18 years Primavera (http://www.primavera.com/) has been developing project software applications. They are second to Microsoft in sales of project scheduling software applications (Pollack-Johnson and Liberatone 1998). Today Primavera has six main software scheduling applications. Their top of the line all-encompassing Primavera Enterprise® is for managing all aspects of multi-projects for an entire enterprise. It is designed for virtually all types of projects from construction through development to professional services. Their Primavera Expedition® suite is tailored for construction projects. Primavera PrimeContact® is for a total e-business construction project and its contractors and suppliers to enter status and in essence manage the project via the internet. Primavera Project Planner® (P3®) is designed for a standalone complex project. SureTrak® Project Manager is designed to be easy to use and an affordable scheduling tool for resource planning and control of small to medium sized projects. Affordable meaning $499 plus shipping and handling compared to 10’s of thousands of dollars for the other programs depending on the features desired and service support required. This section will be reviewing Primavera Enterprise® suite since it is
one of the two packages recommended by Primavera for new product development projects. The other product recommended is Primavera TeamPlay® suite, which will be discussed in the next section. Primavera Enterprise® is the top of the line and as expected the suite is rich in features from all web based to being able to provide data and reports to all the stakeholders in the form that they will find useful. Primavera Enterprise suite is actually a set of programs with the heart of the set being Primavera Project Planner® for the Enterprise (P3e®). “As stated above P3 was designed for a standalone complex project. In this application P3e has been expanded to become a comprehensive, multi-project planning and control software tool. P3e as P3 is built on an Oracle or Microsoft SQL Server relational database for enterprise-wide project management scalability. As will most of commercially available software applications the start of all project planning is the identification of the tasks, estimating how long each task will take, arrange the tasks in a logical sequence, assign any constraints as necessary, assign resources and determine a critical path by running a CPM. Some of the supporting application software is designed to make this process easier and more accurate. More of the supporting application software however, is to give visibility of the project status to the various stakeholders. The intent is to give these stakeholders information so they can mitigate problems on the horizon.

2.9.3 Primavera TeamPlay® (Primavera Systems, Inc.)

As mentioned in the section above Primavera is second to Microsoft in project scheduling software application sales. Primavera now offers 6 scheduling applications and of those, two were recommended by Primavera for new product development of which concurrent development is a subset. One of the two products is Primavera
Enterprise®, which was analyzed in the section above. In the present section the other recommended product, which is Primavera TeamPlay®, will be analyzed. Primavera states TeamPlay® “makes managing projects a repeatable, predictable and positive experience of everyone involved.” As with Primavera Enterprise the TeamPlay® application is web based and can be used on either an Oracle or Microsoft SQL Server which allows for an unlimited number of projects and resources to be entered enterprise wide. This centralized database, which contains all of the project history and knowledge, is not only for all of the current projects but also develops metrics for future projects. Another feature is the centralized resource management, which is designed to ensure the right person is assigned to each task. A goal of TeamPlay® is to allow team players to plan their workday consistent with the project priorities. Thresholds can be established to cause TeamPlay® to send out e-mails when those thresholds are met. As with Primavera Enterprise® project scheduling starts with the identification of activities, estimating task duration, arranging the tasks in a logical sequence, assigning resources and determining the critical path with CPM. No mention is made of anything other than the need for point estimates to be entered for activity duration nor is any kind of Monte Carlo simulation available for risk management assessment. However, TeamPlay® is rich in features to track progress so that the appropriate stakeholder can take action to remedy any pending problem. Being web based, the response time can be far quicker to optimize resources, mitigate risks and to allow collaboration of team members to work out project issues.
2.9.4 Open Plan® (Welcom)

Welcom (http://www.welcom.com/) was established in 1983 and provides project and cost management solutions. This includes application software, consulting, training and technical support. Their main stay application software program is called Open Plan®. It has all the features of an up to date project scheduling software application program. Their focus is on multi-project and enterprise wide applications. This means it has features to allow easy integration between not only their capstone software such as Cobra® but also applications such as those from Microsoft® such as Microsoft Project® scheduling software. Cobra® is their top application program, which integrates all their enterprise software applications. Welcom has attracted some big name corporations to use this application to include Lockheed Martin, BAE Systems and General Motors. The US Navy is also using it. They have also established a footing in France and England.

As with other scheduling software application they have incorporated CPM in their basic scheduling tool to determine the Critical Path. They do have a Risk Management feature that allows entry of three point estimates for some or all activities. These estimates are used to feed normal, beta, triangular or uniform activity distributions and then are used to run Monte Carlo simulations. Open Plan has a fairly extensive list of resource management features to include allowing the resources available to be entered in a variety of distributions. It also has a feature, which allows allocating resources over all of the enterprise projects. Welcom has also included web based reporting to include automatic e-mails when certain predefined conditions are met and entry of performance data. This allows all the enterprise stakeholders from the people performing the task to the company’s CEO to have access to the project data in a form tailored to their needs. In
summary, Open Plan is a modern application with many features however, project scheduling start with identifying the activities, arrange them in a logical sequence, assigning restraints and running CPM to determine the critical path.

2.9.5 PS 8\textsuperscript{TM} (Scitor Corporation)

PS8 is the latest scheduling tool by the Scitor (http://www.scitor.com/) Corporation. The Scitor Corporation was formed in 1979 and offered their first product for project management in 1982. They have been adding enhancements over the years keeping pace with the other major suppliers of scheduling software applications. Actually, PS8 is part of Scitor’s capstone application program called Scitor PS Suite\textsuperscript{TM}. This suite includes not only PS8, which is the heart of the suite, but also Project Communicator which adds web base communications between stakeholders and PSI which is the system interface between PS8 and SAP R/3\textsuperscript{®}, a cost accounting system. Scitor PS Suite is an enterprise-wide application package for multi-project, resource leveling tools and web based. However, PS8 is what all else is built on and at the center of PS8 scheduling is the CPM. In the newest release of PS8 Scitor added Critical Chain Program Management (CCPM) as an alternate to the basic CPM but as discussed in Section 2.6.1.5, CCPM is founded on CPM. A review of the features determines the basic task duration entry is a single point entry. There is an option to enter 3 points per task for running a PERT schedule, which is interpreted to mean that tasks will follow a Beta distribution. There is no option to run a Monte Carlo simulation for any risk determination. Scitor has become one of the leading scheduling applications providers by providing a rather wide range of applications from a single project application on a personal computer for $1000 to a typical divisional enterprise-wide application running
on a server for a little over $63,000. The prices are from their web in June 2002. In summary, the basic development of schedules is the very similar to the other top applications reviewed. That is, tasks are identified, task durations are estimated, tasks are arranged into a logical sequence or a network, constraints and dependencies are assigned and critical path is determined using CPM.

2.9.6 Commercial Software Comparison

The top four suppliers of scheduling software applications were reviewed with those being Microsoft Corporation, Primavera Systems, Welcom and Scitor Corporation. All four have been producing and improving their scheduling offerings for 19 or more years and have worldwide markets. They all offer a range of applications to satisfy the needs of an individual project to multi-projects. They also have offerings for the small business to medium to large enterprise-wide business. All four offer most all the same features with the most recent enhancements being centered on adding web features to increase collaboration and communications among the stakeholders. There are a few unique features that separate the suppliers. For example, Scitor is the only one to offer Critical Chain Program Management (CCPM) as a means to develop a schedule. Another example is Primavera and Welcom where they are the only ones reviewed that offer Monte Carlo simulation as part of their basic program. However, the basic generation of schedule is the same across all the applications reviewed. That is, all the tasks for the project are listed and entered in the program. Next or concurrently the duration of each task is estimated and entered into the program. Virtually all programs have, as there default position single point entries for task duration. All did allow entry of three point estimates for task duration so that a PERT analysis could be down. The tasks are then
linked together by assigning dependencies and constraints to one another. Lastly, the critical path is determined by using the CPM. Scitor allows CCPM be run as an alternative/addition to CPM. These methods are assumed valid and will produce schedules that will accurately predict the outcome of a project. However, as reviewed in Section 1.2 rarely do projects come in on the initial schedule. The focus of this study is to understand why.

2.10 Organizations Conducting/Encouraging Scheduling Research

There are a number of professional organizations that conduct and/or encourage research in project scheduling. Their results are often published in magazines and journals and presented at conferences and conventions. The ones that have a strong interest in project scheduling are the following.

2.10.1 Project Management Institute (PMI)

Since being established in 1969, PMI (http://www.pmi.org) has grown to a worldwide organization of over 90,000 members of primarily project management professionals. This non-profit organization has local chapters around the world to include China to Ireland to Chile to name a few. PMI offers a full range of services from Training and Development, Annual Seminars and Symposiums, a Bookstore, a set of PMI Standards Program Products to include their well know Project Management Book Of Knowledge (PMBOK®), a Knowledge and Wisdom Center, a PMI Corporate Council to work with corporations to further project management and an Awards Program. A cornerstone of PMI is their Project Manager Certification program, which more and more companies are requiring as a prerequisite to becoming a project manager. To be
certified as a Project Management Professional (PMP®), the candidate must pass a comprehensive examination and have over 3 years of documented experience. In addition, continuing annual education is required to maintain the PMP designation. Four periodic publications are produced which includes a quarterly Project Management Journal for detail examination of critical project management issues, a monthly PM Network for project management topics in an easy to read magazine format, a monthly PM Today a newspaper like supplement to PM Network and a periodic The PMI Project Management Fact Book to include a listing of current books on project management. A particular good source for material on scheduling is the annual seminars and symposiums sponsored by PMI where hundreds of papers are presented. Copies of the papers are available through their bookstore. PMI also has a very active research program where they encourage volunteer research as well as funding a limited number of projects. All the information is available in their Knowledge and Wisdom Center. A new addition to the research program is their Research Conferences, which are designed for the researcher and academics in project management. A review of this paper’s bibliography will find more source material was obtained from PMI than any place else.

2.10.2 Institute Industrial Engineering (IIE)

The IIE (http://www.iienet.org/) started as the American Institute of Industrial Engineers or AIIE in 1948. By 1966 the organization had grown to be international so the A was dropped and simply became the Institute of Industrial Engineers. Today they have over 17,000 members in 150 chapters worldwide dedicated to serving the industrial engineer. IIE provides a wide range of services typical of many professional organizations from education to an annual convention to research. They publish two
monthly periodicals. The first is IEE Transactions, which has four focus areas from
design and manufacturing to scheduling and logistics to quality and reliability
engineering to operations engineering. A search of all articles published found many on
scheduling and one rather recent on critical paths. The article is titled “Robustness to
Variability in Project Networks” (Gutierrez, McCombs School of Business et al. 2001).
These investigators found that increased variability in tasks at the start of a project had
about the same impact on the overall project duration as increased variability on tasks
toward the end of the project. However, if a project has dominant critical paths,
variability of early stage tasks had a greater impact on the overall duration than increased
variability in late stage tasks. This knowledge was used in structuring project teams.
The other publication published by IIE is IIE solutions. It is in a magazine format. A
review of articles found the cover story in the October 2001 edition titled “Broken
Promises” (Yu-Lee and Lorenzl 2001). The authors examine why business cases made at
the beginning of a project often go awry. They give three reasons. First, the justification
tools used to identify and reduce costs do not reflect the bottom line. Second, the
cost/benefit analyses require additional action that is overlooked or not taken. Third,
focusing on cost reduction independently of profits may lead to profit-limiting behavior.
These authors were more concerned with a production line than a concurrent
development but they make a good point that management will be highly motivated to
reduce costs and forget about the business case. The result is the project loses money.
In summary, IIE is another good source of information on what is being done to
improving project scheduling.
2.10.3 International Council on Systems Engineering (INCOSE)

INCOSE is a not-for-profit organization that promotes “the application of an interdisciplinary approach and means to enable the realization of successful systems.” It is an international organization with 32% of its membership being non-US. INCOSE is a relatively new organization being established in 1990 but offers a full range of features of a professional organization. They include a newsletter titled Insight, a journal titled Journal of Systems Engineering, a yearly symposium and a sponsored research program to name a few. This organization is of particular interest to project scheduling in that systems engineers and practitioners are concerned with the overall success of the project.

A review was made of the articles published in INCOSE Insight newsletter. An article published in the Winter 1999 newsletter titled “Before requirements: what, who, Where, When, why and How” (Gaasbeek 1999) makes this point very well. The first step in a project schedule development is task identification. This article makes the point that for a project to be successful a set of validated requirements is an absolute. He also makes the point that all product development projects must define the same types of information to define the problem. The articles published in the Journal of Systems Engineering were also reviewed for application to study of schedules. An article was published on line in April 2002 journal titled “Why Projects Often Fail, Even with High Cost-Contingencies” (Kujawski 2002). This author makes the point of maintaining a project-wide contingency verse allocating the entire contingency to each of the individual subsystems. This is close to what is advocated in the Critical Chain Project Management (CCPM) approach to project scheduling discussed elsewhere in this paper. Lastly, the articles presented at the annual symposiums were reviewed. As an example of one presented at the 2002
symposium, their 12th was titled “Toward a Mathematical Theory of Systems Engineering Management” (Honour 2002). Here the author trades a list of project variables to include schedule, cost and risk and explores the benefits and drawbacks of each relationship. INCOSE reinforces the needed for studying and improving scheduling.

2.10.4 Software Program Managers Network (SPMN)

The SPMN (http://www.spmn.com/) was established in 1992 by the Assistant Secretary of the Navy as a result of continuing over run of cost and schedule of large software intensive programs. In early 2002 the lead was transferred to The Deputy Under Secretary of Defense for Science and Technology (DUSD)(S&T). Software Intensive Systems Directorate to emphasize its applicability to the tri-services. The initial goal of SPMN was to identify proven industry and government software best practices that addressed the underlying cost and schedule drivers that have caused many software programs to be delivered over budget, behind schedule and along with other significant performance shortfalls. The result was the SPMN 16 Critical Software Practices™ (http://spmn.com/16CSP.html) for performance based management. Today the mission of SPMN as stated on their web site is “To seek out proven industry and government software best practices and convey them to managers of large-scale DoD software intensive acquisition programs.” To help them in this mission the SPMN has contracted with Integrated Computer Engineering, Inc (ICE) (http://www.iceincusa.com). They claim to have helped over 250 DoD programs by providing on-site assessments, risk assessments, software tools, guidebooks and training. Of particular interest to this study is the second and third of the 16 Critical Software Practices. The 16 Critical Software
Practices are divided into three groups, which are Project Integrity, Construction Integrity and Product Stability and Integrity. Best Practice number 2 in the Project Integrity group is to “Estimate cost and schedule empirically.” This best practice states that task estimate at the start of a project should be looked as a high risk venture due the lack of definitive detail of what the task really is. The best practice recommends both a top down estimate such as metrics and bottoms up engineering estimate should be conducted. A sanity check such as industry standards should also be performed and finally the estimates need to be approved. At every program review these estimates need to be reviewed and updated with the latest information. Best Practice number 3 also in the Project Integrity group is to “Use metrics to manage.” The best practice is to clearly identify the metrics to be tracked at the start of a program to include limits when action needs to be taken. Not only must they be identified, a system needs to be in place to collect the metrics in a timely manner. In summary, two of the 16 best practices deal with the quality of the input data to schedule development and the need to track and update as the project proceeds. In other words, the input data to a schedule needs to be reality if there is any hope that the final schedule will be.

2.10.5 ProjectWorld

Imark Communications conducts expositions and conferences for a number of professional groups with one being for those responsible for projects and wanting to improve their skills to manage their projects for on-time, on-schedule and on-expectation performance. Imark Communications calls these expositions and conferences ProjectWorld with 5 expositions and conferences scheduled in 2002. They have a long list of sponsors to include the Project Management Institute (PMI), Microsoft, Primavera
Systems Inc and Scitor Corporation. Imark Communications also started publishing in
the last 2 years a magazine called Project@Work designed for the project management
professional. Lastly, ProjectWorld has a Knowledge Center that is a repository of papers
on project management topics that is web accessible. Since these expositions and
conferences, the Project@Work magazine and Knowledge Center are targeted to the front
line workers in project management and as a result, scheduling, this should also be a
good area to review to determine what are the issues with project scheduling today. A
review of the articles published in Project@Work found a recent article titled “On
Schedule: Scope It Out” (Curtis 2002) where the author makes the point that without a
well defined scope statement the project is at risk. This scope needs to be well
documented in writing to ward off scope creep. Any scope change needs to be formally
changed in the scope statement document. A review was also made of the proceedings of
recent conferences. One presentation that is titled “Design Better Projects Using
Dependency Structure Matrices” (Denker 1999) is a good example where the author
makes the strong case for how the tasks are arranged is critical to a better project. He
uses what he calls Dependency Structure Matrices (DSM) to logically lay out the
relationships. In summary, Imark Communications with ProjectWorld provides good
insight into what is troubling project management practitioners.

2.11 Summary

This chapter on the CD Scheduling Problem analyzed the scheduling process from
the overall process to task identification to networking techniques to resource loading
techniques. Much has been written and studied on how to develop a project schedule that
will actually predict the outcome of the project. In a way this is a testament to the strong
need for scheduling techniques that accurately predict the outcome of a project. The focus of this study is on the CD scheduling problem and to find ways to improve scheduling techniques. As all the elements of scheduling process were analyzed, the uniqueness of the CD scheduling problem was compared to the elements of scheduling to find ways that might improve the CD scheduling process. First, task identification was analyzed. The sections on task identification underline the importance of this building block step. If all the tasks are not identified or miss identified, there is no hope that no matter what networking technique is used, the resulting schedule will be flawed. Once the tasks are identified, the task durations need to be estimated. In the CD scheduling problem, this step is particularly challenging since many of the tasks may have never been done before. There are many aspects to task duration determination. First, what is the credibility of the estimator? Next, what does the estimator assume as to what they are providing? Does the estimator assume high assurance or is it 50 percent chance of making that estimate or is it something else? If there is no stated guidance, each estimator will provide their own estimate of what is required. As evidenced by the most commercial scheduling tools available, a point estimate is simply all that is required. No assumptions are required to be stated or assume. That is, this task will take for example 32 hours and as a result, someone reviewing the final schedule will be at a loss as to what confidence level to place on the schedule. One last element is what is assumed as the task distribution of the possible outcomes. For example is it normal, triangle, beta or something else? For the most part, CD project estimators will have only a general idea since this task has never been done before. As will be shown in the preliminary results, this is a fertile area for adding structure to a most often uncontrolled aspect of schedule
development. After the tasks are identified and an estimate is made of their duration and
sometimes an estimate on distribution of tasks, the networking process commences. In
the above sections, numerous techniques were examined for applicability to the CD
scheduling problem. Many required a considerable amount of effort to develop the
schedule. The cost benefit of this extra effort for an improved estimate is difficult to
judge other than to conclude that the method of choice today by far is CPM. CPM is
used in virtually all commercial software scheduling applications. A question that should
be asked is with this almost universal acceptance of CPM, how good is it at predicting the
actual outcome of a project? The preliminary results show that at least in the CD
scheduling problem the results are almost if not always overly optimistic.
Chapter 3

The Concurrent Development Scheduling Problem (CDSP)

3.1 Introduction

The CD Scheduling Problem (CDSP) is defined as most all CD baseline project schedules being developed today turn out to be overly optimistic. Two approaches have been identified to solve this problem. One approach is to develop a better scheduling technique. The other approach is to accept the baseline schedule being developed today or to develop a technique to assess the optimism of the baseline schedule. Each approach is briefly discussed below.

3.1.1 New Scheduling Technique

A baseline project schedule should show an expected duration that reliably predicts the actual project duration outcome. This schedule must also allow tracking of actual progress and alert the project manager of a problem early so corrective action can be taken. Many techniques and tools are available to schedule a project as reviewed in Chapter 3 but only a few of them have found acceptance and are being used in practice today with Critical Path Method (CPM) and Precedence Diagramming Method (PDM) being the most notable. However, CPM/PDM are the methods most used today and these baseline schedules are the ones that are optimistic. A better technique is desperately needed to schedule CD projects that will produce repeatable on-time-delivery schedules. This literature review looked at the techniques and tools proposed to date to determine
what works under what circumstances and where are the errors coming from. To be useful the technique had to be easy to understand, easy to obtain results with a reasonable amount of effort and easy to update as the project progresses. Many have proposed enhancements to CPM/PDM but most require much more effort with little or no perceived improvement in quality of the schedule. Also, no totally different technique from CPM/PDM was found that would help solve or partially solve the CDSP.

3.1.2 Assessing Optimism

Another approach to solving the CDSP is to develop a technique to judge a baseline schedule for its accuracy. That is, accepting the baseline schedule as developed today but has a technique to judge its optimism. One way to do this is to develop a model that can be applied to any CD project schedule and determine its optimism or possibly pessimism and by how much. Chapter 4 develops that model. The parameters to be included in the model were derived from the literature review. The literature review was organized around the six components of the CD baseline scheduling development process, which are: overall process, task identification, task duration, networking, resource loading and progress tracking. A section in Chapter 3 was devoted to each area. Each component is discussed below as a result of the literature review on how they can contribute to the optimism of the overall schedule. Chapter 4 takes these parameters and develops a model to judge the optimism of a CD schedule.

3.2 Overall Process

The overall scheduling process can be divided into a structural component and a non-structural one. Here the structural component means the overall technique used to develop the schedule such as Ghant charts, CPM, PDM, Critical Chain Project
Management (CCPM) or Milestones to name a few. Most of these techniques are used with the assumption that the logic behind the method is sound. However, these methods are not without logic flaws and can contribute to the overall accuracy of the schedule as will be shown below under task duration and networking. The non-structural component includes such items, as is concurrent development even a good idea? Might it not be better to mature the hardware before software is applied? With this approach, hardware and software problems would more easily be isolated. Applying software early presents the situation where a problem can’t be readily identified as a hardware or software problem. Another factor in the overall process is who does the scheduling and how much time should be allocated to the development of a schedule? Still another, the schedule may not be the problem but the real problem may be the lack of discipline in living up to the schedule. All these factors can contribute to a poor initial schedule. The question is what can be done to improve? This study focused on the structural aspect of scheduling and leaves the non-structural aspects to others.

3.3 Task Identification

Task identification is the listing of the individual tasks to be accomplished. There are many factors that can be considered in establishing the rules to be used in selecting how tasks are identified. They include how many tasks should there be i.e. how small should each task be and should the tasks be functional i.e., tasks to be accomplished by a digital engineer or should the tasks be projects i.e., tasks to be accomplished by a cross-functional team of people to complete a sub unit. Task identification produces the building blocks in a schedule development. Clear understanding on the assumptions in task identification is a must to fully understand the quality of the schedule that is
developed. Poor task identification will likely produce an initial schedule with no hope of predicting the outcome. As it turns out the length of an individual task in relationship to the other tasks in a CD project schedule does impact optimism. This influence is considered in the final model. Also the total number of tasks in a project schedule impacts the project duration. This influence is also included in the final model.

3.4 Task Duration

Task duration is the time it takes to perform a task. Accurately determining task durations is critical to any schedule development. Many techniques have been used with varying degrees of success. One technique is to simply ask the person who is to do the task how long it will take? Some use a parametric or a metric developed from historical data if available. However, in most cases a point estimate is given and that being the expected duration of the task. Since most tasks have a duration distribution function, an error is possible. The investigation has shown the typical CD task duration distribution will cause an optimistic overall schedule. In summary, the accuracy in task duration determination is directly related to the quality of the final schedule. The types of task duration distribution, the skewness of the task distribution and the uncertainty of the estimate all have an impact on optimism. All these influences are included in the final model.

3.5 Networking

Networking is the process of connecting all the tasks together in a logical manner and determining how long the project will take to complete. Over time many networking techniques have been proposed. By far the most popular is the CPM/PDM. Both are straightforward and PDM can be easily implemented on a computer, which can also
handle the mathematics necessary to determine the critical path. Once a logical network is developed, the networking effort is often far from over. As is normally the case, the completion date determined by the network is beyond the date when the product is required. Numerous techniques such as crashing are employed to bring the project schedule within the desired delivery date. On the surface the CPM/PDM looks straightforward and a logical way to develop a schedule. However, shortcomings were found in the CPM/PDM method that can contribute to optimistic schedules. One shortcoming is the merge point phenomenon, which is present in every CD project by its very nature. Preliminary investigation showed that the more parallel paths into a merge point the more optimistic a schedule. The model includes the impact of merge points.

3.6 Resource Loading

Resource loading is the process of adjusting the networked schedule to account for the amount of resources available usually human or machine or cash flow. Resources are almost always limited and the initial network will have numerous resources over subscribed. Again, numerous techniques have been offered as how to do this resource loading. These techniques depend heavily on the assumptions made, all of which will contribute mightily to schedule accuracy. The management of critical resources is the key to resource loading. This was considered a non-structural and was not included in the model present. Further research may find a way to incorporate resource loading into the model.

3.7 Progress Tracking

Progress tracking is the process of assessing where the project is against the schedule as the project progresses. There are several dimensions to progress tracking
such as frequency of reporting and numerous techniques used such as earned value
developed to assess progress. There are also a number of techniques on how progress is
assessed. Obviously, if the schedule doesn’t reflect reality, schedule variance will
quickly appear. The developed model was designed to assess the baseline schedule
however, can be used periodically throughout the project to assess its optimism.

3.8 Summary

The CD scheduling problem is complex with many opportunities for the baseline
schedule to be in error. There is much evidence to show that this baseline schedule is
rarely right in predicting the actual outcome. In the above, the problem was divided into
six components with each discussed to show some how errors may arise in the schedule
development. Six structural problems were identified in the way most CD schedules are
developed today which gives rise to optimistic schedules. These six factors are the merge
point phenomenon, the number of tasks in a project, the task duration distribution
function, the variability of individual task durations in relationship to one to another, the
shapeness of the task distribution function and the uncertainty of the estimate of the task
duration. Experiments were devised to understand their interrelationships and their
impact on the overall schedule optimism. All six of these factors are in the final model as
developed in Chapter 4. The end result was the development of a concurrent
development scheduling model (CDSM) that can be used on any CD baseline schedule to
assess its optimism. Since the model is mathematical, the model will also show what is
driving the optimism and by how much. Chapter 5 gives the details of the experiments
that developed the model. Chapter 6 states the findings and gives suggestions on how the
optimism may be minimized.
Chapter 4

Methodology

4.1 Introduction

The research progressed through three phases. However, as the research progressed, the research required resetting to an earlier phase on several accounts. The three phases were:

1. Model Development
2. Comparing the Proposed Model with Typical CD Schedules
3. Comparing the Proposed Model with Completed Real Life CD Schedules

The methodology used during each of these three phases is discussed in the chapter. The actual results are presented in Chapter 5 and the findings and recommendations for further research are presented in Chapter 6.

4.2 Fundamentals for Model Development

In this section the fundamentals or the building blocks for model development are presented. These are the foundations on which this research is built. They include the following:

1. Dependent and independent variables
2. Statistics used
3. Confidence interval along with sample size
4. Distribution functions to include normal, beta and triangular distributions
Each of these is discussed below. A constant concern was to ensure the results are unbiased. The process was iterative. Data was collected and analyzed. As the model took shape the above building blocks were ever present giving direction to the research.

### 4.2.1 Dependent and Independent Variables

Project execution involves an almost countless number of variables from scheduling techniques to human emotion. No two projects are ever alike so what worked one time may not work the next time. This is particularly true in CD projects with the rapid advancement of technology making each new project as something never done before. For this research a clear statement was made as to what were the independent and dependent variables.

In virtually all cases, the dependent variable used in this research is the project expected duration. In most cases a data set was obtained of the project durations. This data set was usually analyzed to determine its relationship to the independent variable(s). To conduct most analyses the data was normalized.

Independent variables chosen were from all the parameters that could be entered into a CD schedule. The independent variables investigated as having a possible impact on optimism/pessimism were the following:

1. Length of the task expected duration
2. Distribution type of the task expected duration
3. Skewness of the task expected duration
4. Confidence level of the task expected duration
5. Number of tasks in a path
6. Number of concurrent paths into a merge point
7. Durations of the concurrent paths as they are related to one another

This research made the assumption that all other variables were considered fixed over the
period of investigation. The purpose of this investigation was to determine the impact of
these independent variables on optimism. It is also important to note that each of the
factors listed above are often under some kind of control by the decision maker and can
be modified if it was known the impact they are having on the schedule.

4.2.2 Statistics Used

Analysis of Variance (ANOVA) tables were generated in steps one and three of
the model development. To judge if the factors or interaction of factors were
significance, the standard F-distribution was used. The statistic generated was the ratio
of the sum of squares of the factor or the interaction of factors in question to the error
sum of squares. For this to be valid the F-distribution’s sums of squares are assumed to
follow a chi-square distribution. This is further based on the sample set being
independent and identically distributed random variables. The data used in these
ANOVA tables consisted of independently run simulations. A random number generator
with a random seed was used to initialize each simulation run. The conditions for the F-
distribution to be valid were met. The result was that significance was determined based
on the ratio of the sum of squares. The desired degree of significance of 0.99 was
initially used. This of course showed high significance. In several cases an interaction of
two factors didn’t make 0.99 of significance but did at 0.95 that showed a weaker
interaction and was helpful in determining the next step.
4.2.3 Confidence Interval

In this research over a 1600 simulations were run with each one producing a mean. We know with certainty the mean of the sample distribution but the question is how close is it to the real distribution. The confidence interval is a measure of how good the mean of the sample distribution is to the real distribution. The confidence interval is added and subtracted to the sample mean to determine its range. This is interpreted to be that if this experiment were run many times the mean of the real distribution would lie within the confidence interval to the established confidence level. The confidence interval is dependent on the following three factors: the confidence level desired, the standard deviation of the real distribution and the sample size. Our impact on each of these factors was as follows:

1. Confidence level. Typical confidence levels are 90%, 95% and 99%. As will be discussed in Section 4.4.1 “Simulation Techniques”, the Risk+ add on application to Microsoft Project was used to conduct the simulations. Risk + sets the confidence level to 95% which is satisfactory for this research.

2. Standard deviation. In our case we used the standard distributions of normal, beta, triangular and uniform distributions. In Section 5.4 “Comparing the Proposed Model with Actual CD Schedules” the developed scheduling model is compared to completed CD projects. The results show that the beta distribution, which was used for most of the simulations, maps fairly well to actual collected data. As will be discussed in Section 4.3.5 “Distribution Functions” the beta distribution has the smallest standard deviation of the four standard deviations stated above.
3. Sample size. The confidence interval decreases as the number of samples taken increases. The relationship is the confidence interval decreases as the square root of the number of samples. That is, if the number of samples increases 100 fold, the confidence internal would decrease by a factor of 10. The initial sample size of 1000 iterations was initially chosen. In virtually every analysis, three simulation runs were made. To reduce the variability, i.e. to decrease the confidence interval, the sample size was increased to 10,000 iterations per simulation. Most simulations with 10,000 iterations ran in less than a minute on a standard desktop computer, which made the extra run time a small penalty.

4.2.4 Distribution Functions

A goal of this research is to give decision makers rules of thumb that suggest adjustments to achieve a more realistic schedule from a deterministic CPM generated schedule and the resulting critical path. A key parameter in determining the optimism of a CD schedule is the task duration distribution needs to be understood as to its impact. To investigate this parameter the four standard or frequently used distributions were considered. They are the normal distribution, beta distribution, triangular distribution and uniform distribution. Preliminary research (Lau and Somarajan 1995) has shown that none of these distributions (normal, beta and triangular) mirror real life very closely. However, they have understood mathematics, which makes them ideal for incorporating in any model. If a scheduling application program allows entry of any task distributions, these four will most likely be offered. The first three distributions are briefly discussed below to see the applicability to the CD scheduling problem. The uniform distribution finds little use in CD projects and is not discussed.
4.2.4.1 Normal Distribution

The normal distribution is probably the most popular distribution of all with its universal appeal in many scientific endeavors. However, the paper mentioned above (Lau and Somarajan 1995), showed that rarely is the actual task expected duration distribution normal. However, normal distributions are easy to deal with mathematically. With a true normal distribution the mean, mode and median are all equal and their upper and lower bounds are infinity. For particular matters, a normal distribution is often truncated at plus and minus at some point like three standard deviations or 99.7% of all values. However, the primary shortcoming of the normal distribution for CD schedules is that they are symmetrical. Preliminary data and the results of actual completed CD projects in Section 5.4 “Comparing the CDSM with Actual Schedules” verify that the task duration distributions of CD projects are highly skewed. The normal distribution does not have the ability to reflect this skewness. As a result, the normal distribution was not used in any of the investigations in this research. However, the final model should work quite well with task duration distributions that are determined to be normal but can be assumed to be beta distribution with no skewness.

4.2.4.2 Beta Distribution

The beta and triangular distributions over come the shortcoming symmetrical nature of the normal distribution by being able to assign the mean any where within the distribution range. By design both the beta and triangular have a lower and upper bound without having to truncate the distribution. Another difference between the distributions is that the beta distribution has a narrower standard deviation than the normal distribution where the triangular distribution standard deviation is wider. This comes in useful if you
have a sense of how good your estimate is. For example, if you are fairly confident your
estimate is good; the beta distribution would be the distribution of choice. As used in this
research the beta distribution is defined as follows (from MathWorld by Wolfram
Research)

With the domain (0,1), the beta probability function $P(x)$ is:

$$P(x) = \frac{(1-x)^{\beta-1}x^{\alpha-1}}{B(\alpha, \beta)}$$

$$\alpha, \beta > 0$$

where $\alpha$ and $\beta$ are parameters defining the shape of the beta distribution and

$B(\alpha, \beta)$ is the beta function and defined as follows:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}$$

where $\Gamma(a)$ and $\Gamma(\beta)$ are gamma function that can be found as follows:

$$\Gamma(m) = \int_{0}^{\infty} x^{m-1} e^{-x} dx = (p-1)!$$

if $m$ is an integer:

$$\Gamma(m) = (m-1)!$$

and introducing back into $B(\alpha, \beta)$:

$$B(\alpha, \beta) = \frac{(\alpha - 1)!(\beta - 1)!}{(\alpha + \beta - 1)!}$$

the mean is:

$$\mu = \frac{\alpha}{\alpha + \beta}$$

the mode:
\[
\hat{x} = \frac{\alpha - 1}{\alpha + \beta - 2}
\] 4-7

The variance:

\[
\sigma^2 = \frac{\alpha \beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)}
\] 4-8

The numbers for “a” and “ß” are the key beta distribution parameters. The smaller the parameter “a” is in relationship to “ß” the closer to zero is the mode and the distribution is skewed to the right or more optimistic. That is, the project is less likely to be completed on time or the schedule is optimistic. The smaller “ß” is in relationship to “a” the closer the mode is to one and the distribution is skewed to the left or more pessimistic. Also, the larger the “a” and “ß” parameters are, the tighter or smaller the standard deviation of the distribution. The software application program Risk + used for this analysis assigns the number 6 to one of these parameters and finds the other one based on other information entered into the program but in no case is either number bigger than 6. This is intended to represent real life task duration distributions. Larger values of “a” or “ß” make the standard deviation smaller.

The equation 4-8 for variance is an exact solution for a beta distribution but a simpler way was needed to find the variance of a beta distribution of a real task duration distribution. In fact, it was desired that a method for finding variance be even more general than just a beta distribution since we will not really know for certainty the exact distribution of the task expected duration. The answer was found in a relationship commonly known in statistics for many unimodal distributions. That is, the standard deviation is equal to one-sixth the range of the distribution (Moder, Phillips et al. 1983) or in our case:
As stated above the beta distribution is only defined in the domain $[0,1]$. If a beta distribution is selected to represent a task duration distribution, a conversion or mapping is needed to “scale” the beta distribution to the real distribution. Conversely if a set of task expected duration data is obtained from a project and it is desired to curve fit this data to a beta function, a mapping method is needed. The following equation was used through out this research to do that mapping (Risk + by C/S Solutions):\[ X = a + (b - a)P(x) \] Here $X$ is the distribution of the real task expected duration. It has 4 parameters ($a$, $b$, $a$ and $\beta$). The parameter $a$ and $b$ are the lower and upper bound respectively of the real distribution and $P(x)$ is the beta distribution which has a value of $[0,1]$ determined by the beta distribution parameters $a$ and $\beta$. Or going the other way from a real task expected duration to a beta distribution: \[ P(x) = \frac{X - a}{b - a} \]

### 4.2.4.3 Triangular Distribution

The triangular distribution as did the beta distribution has the advantage over the normal distribution in that the mean can be placed any where in the range by the scheduler. Here the triangular distribution is not confined to the domain $[0,1]$. The differences between triangular and beta distributions for this research are considered minor. In Table 4-2 below a model was generated that analyzed several distributions to include triangular and beta. The standard deviation was slightly bigger for the triangular
distribution than normal. The triangular distribution is however, piece wise continuous requiring two equations to describe the function. For this reason, the beta distribution was used almost exclusively throughout this research.

4.3 Special Tools and Techniques for Model Development

Several special tools and techniques were used throughout this research in the development of the model. They were a simulation technique, non-linear regression, curve fitting and determining a correlation coefficient. Each is discussed in the following sections.

4.3.1 Simulation Techniques

A tool was needed to accomplish the following:

1. Analyze the inter relationships between the factors impacting optimism. For example in step 4 the activity was to determine the impact of varying the number of tasks in a schedule. Simulations runs were used to understand that relationship.

2. Determine the effectiveness of models as they matured in each advancing step.

The tool was needed to analyze dynamic problems. The driving functions of each analysis and each model were task duration distributions. Two tools were considered. They were the Monte Carlo Method and Latin Hypercube Sampling. The techniques and advantages of each are discussed in the following sections.

4.3.1.1 Monte Carlo Method

Monte Carlo Method is named after the city in the Principality of Monaco. Monte Carlo is of course known for gambling and in particular, the roulette table which in effect
is a random number generator. The Monte Carlo method has developed into a method to give approximate solutions to intractable mathematical problems. A number of people contributed to the development of the method. The first person was a student (W.S. Gosset) in 1908 but it wasn’t until 1944 when the Monte Carlo method became a research tool in the development of the atomic bomb. The probabilistic problem in understanding diffusion in fissionable material was an intractable problem. For example, Harris and Herman Kahn [Pllana, year unk #214] used the Monte Carlo method to find eigenvalues estimates to the Schrodinger equation. Further it was found the error in estimate decreased by one over the square root of the number of samples, which helped, bound the answers produced by the method.

The CD scheduling question becomes what is the distribution of the overall project duration outcomes when all the tasks in the schedule have probabilistic distribution values. This CD schedule problem described with task probabilistic durations distributions is an intractable mathematical problem. However, the Monte Carlo Method can find an approximate solution to the problem. These are the steps:

1. Determine the cumulative distribution function (cdf) of the task expected duration distribution under consideration. See Section 4.3.4 “Distribution Functions” for a discussion on the type of distribution to be used. Scale the task expected duration from zero to one. The result is a probability verses duration curve with both axes going from zero to one.

2. Generate a number with a random number generator from a range of numbers that can be converted to a scale of zero to one. If more than one run is planned of the
same schedule and conditions, a random seed is needed to start the simulations. Risk +
has that as a feature. If not used, all the results will be the same producing no insight.

3. Use the random number that has been converted to zero to one and enter this
as the probability into the cdf generated in step 1. This corresponds to a task duration
after the zero to one which is reconverted back to the actual duration for this case.

4. Steps 2 and 3 are performed on every task in the schedule.

5. The task durations determined in steps 3 and 4 are entered into the CD
schedule. A critical path analysis is run on this schedule to determine projected project
duration.

6. Steps two through five are run many times. In this research the number of
iterations was set at 10,000 for each simulation run.

7. The results will produce a set of data and in our case 10,000 outcomes. Using
a histogram approach, this produces a distribution of possible outcomes. The mean and
standard deviation is calculated. The mean is then the expected project duration.

A number of options were available to run simulations using the Monte Carlo
Method from add on to stand-alone applications. The one chosen for this research was
Risk + by C/S Solutions, Inc which is an add-on to Microsoft Project. Over 1800
simulations were run with most runs at 10,000 iterations. The results of these simulations
were used to analyze the inter relationships between the factors impacting optimism and
to determine the effectiveness of models as they matured in each advancing step.

4.3.1.2 Latin Hypercube Sampling

The advantage Latin Hypercube Sampling (LHS) has over Monte Carlo
simulation is that the number of iterations can be reduced yet still attains the same degree
of accuracy. Reducing the number of iterations may be important if each iteration is
difficult to take. LHS also has the advantage of ensuring that distributions with “long
tails” of low probability are accurately considered. On the negative side each iteration
will probably take longer which will counter some of the time saving in reducing the
number of iterations.

The basic LHS approach for a CD scheduling problem is to divide the task
duration cdf into an equal number of intervals. Then with a random number generator
produce an equal number of samples for each internal. Here are the steps in using the
LHS technique on a CD schedule:

1. Determine the cdf for the task duration as was done with the Monte Carlo
   method.

2. Divide the probability interval into equal subintervals such as 20.

3. Determine the number of samples N to be taken. The number needs to be a
   multiple of the number of subintervals. When completed each subinterval will have the
   same number of samples.

4. With a random number generator find a number in each subinterval. The idea
   is to continue looking for a value for each subinterval before a second value is added to
   any subinterval. If a random number generator produces a number in a subinterval
   already used, that number will be discarded.

5. When each subinterval has one sample, a second sample is then added to each
   subinterval. This continues until all N values are found.

6. The results are analyzed as with the Monte Carlo approach.
As with the Monte Carlo method a number of software applications are available that implements LHS. The one selected for this research was Risk + which is an add-on to Microsoft Project. This is the same package selected for the Monte Carlo method as stated above.

The two methods of Monte Carlo method and LHS were considered almost identical for the CD scheduling problem. The research conducted simulations on relatively well-behaved distributions and the size of the schedules were relatively modest for which we might have to select LHS. Today’s desktop computers are more than up to the task of running 10,000 iteration simulations. Most of the simulations in this research ran in less than one minute. To add credibility to this claim a sample project was generated and the two methods were run on the schedule. The results are shown in Table 4-1. The Monte Carlo method was selected and used through out primarily because it is better known and LHS did not offer any particular advantage over the Monte Carlo method for this research.
### Table 4-1 Monte Carlo Method Verses LHS

**Conditions:** Each model condition is exactly the same except for simulation technique and number of partitions. A +/- 50% Beta distribution is used for all tasks.

<table>
<thead>
<tr>
<th></th>
<th>Arabic Hypercube</th>
<th></th>
<th>Monte Carlo</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ave Sim</td>
<td>Ave Percent</td>
<td>Ave Std Dev</td>
<td>Ave Sim</td>
</tr>
<tr>
<td>20 Partitions</td>
<td>3.65</td>
<td>2.10</td>
<td>2.13</td>
<td>3.65</td>
</tr>
<tr>
<td>Random Seed</td>
<td>3.65</td>
<td>2.15</td>
<td>2.15</td>
<td>3.65</td>
</tr>
<tr>
<td></td>
<td>3.63</td>
<td>2.15</td>
<td>2.15</td>
<td>3.65</td>
</tr>
<tr>
<td>40 Partitions</td>
<td>3.63</td>
<td>2.13</td>
<td>2.13</td>
<td>3.70</td>
</tr>
<tr>
<td>Random Seed</td>
<td>3.63</td>
<td>2.12</td>
<td>2.12</td>
<td>3.63</td>
</tr>
<tr>
<td></td>
<td>3.63</td>
<td>2.13</td>
<td>2.13</td>
<td>3.60</td>
</tr>
<tr>
<td>100 Partitions</td>
<td>3.63</td>
<td>2.14</td>
<td>2.14</td>
<td>3.63</td>
</tr>
<tr>
<td>Random Seed</td>
<td>3.65</td>
<td>2.14</td>
<td>2.14</td>
<td>3.63</td>
</tr>
<tr>
<td></td>
<td>3.63</td>
<td>2.14</td>
<td>2.14</td>
<td>3.63</td>
</tr>
</tbody>
</table>

**Conclusions:** Number of partitions and/or type of simulation Arabic Hypercube or Monte Carlo seem not to impact amount of optimism.
4.3.2 Non-Linear Regression

A technique was needed to describe the relationships when varying one or more parameters and holding all else constant. The entire CD scheduling problem is intractable which eliminated any kind of linear analysis. A non-linear regression technique was needed. The non-linear regression technique in the Statistical Tool box in MATLAB by The MathWorks, Inc. was found to work quite nicely on the data sets generated through out this research. MATLAB uses the Gauss-Newton method, which uses a Taylor series expansion eliminating high order terms. The general approach is least squares, which is a mathematical optimization technique to find the best fit of data by minimizing the sum of squares of the differences between the suggested function and the data presented. The Gauss-Newton method is an iterative method meaning you need to provide an initial set of values [Wikipedia, year unk #215]. False results or non-convergence can occur if the initial values are far from the real answer. MATLAB modifies the Gauss-Newton method with one developed by Levenberg-Marquardt to enhance global convergence. This tool was used extensively in the initial steps in the development of the model. In particular, the data derived from the full factorial designs were subjected to this technique to arrive at the models present in step one and three.

4.3.3 Curve Fitting

Completed real CD schedules were analyzed. The initial baseline schedules were obtained and compared to complete schedules. The data was normalized meaning that the completed task durations were compared the initial schedule task duration. A value of one meant that task completed with a duration exactly equaled to the duration estimated at the start of the project. This resulting data set needed to be subjected to a
curve-fitting algorithm. MATLAB was also used here. In the MATLAB statistical
toolbox there is a curve-fitting tool. The tool used least squares as in the non-linear
regression tool discussed above. Here however, you need to provide a suggested function
and the curve-fitting tool will provide the best-fit parameters for the function. MATLAB
provides as standard entries beta, normal, triangular and uniform distributions. There is
an option to provide any other distribution. Since all of the data analysis was highly
skewed, the normal and uniform distributions were eliminated from considerations. The
two viable distributions were beta and triangular without having to resort to a special
distribution function. The challenge was what distribution to choose. A schedule was
generated to study the impact on the various distributions. The results are shown in Table
4-2. Here optimism is measured as before. That is, the extra duration due to the
variability of the tasks divided by the critical path length expressed as a percent.
Table 4-2 Distributions Comparisons

*Condition:* Each model is exactly the same except for task distribution and the percent variability of tasks. The beta distribution is used as the standard. This is Model 1. Other distributions have their variability varied to achieve the desired standard deviation of the standard. Models 2 through 13. Each task is 10 days long.

<table>
<thead>
<tr>
<th>Model</th>
<th>Variability</th>
<th>Distribution</th>
<th>Ave Sim</th>
<th>Ave Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>+/- 50%</td>
<td>Beta</td>
<td>3.70</td>
<td>2.173</td>
</tr>
<tr>
<td>Model 2</td>
<td>+/- 50%</td>
<td>Normal</td>
<td>4.78</td>
<td>2.720</td>
</tr>
<tr>
<td>Model 3</td>
<td>+/- 40%</td>
<td>Normal</td>
<td>3.69</td>
<td>2.393</td>
</tr>
<tr>
<td>Model 4</td>
<td>+/- 30%</td>
<td>Normal</td>
<td>3.66</td>
<td>2.087</td>
</tr>
<tr>
<td>Model 5</td>
<td>+/- 50%</td>
<td>Triangular</td>
<td>5.69</td>
<td>3.400</td>
</tr>
<tr>
<td>Model 6</td>
<td>+/- 30%</td>
<td>Triangular</td>
<td>3.53</td>
<td>2.017</td>
</tr>
<tr>
<td>Model 7</td>
<td>+/- 31%</td>
<td>Triangular</td>
<td>3.60</td>
<td>2.077</td>
</tr>
<tr>
<td>Model 8</td>
<td>+/- 32%</td>
<td>Triangular</td>
<td>3.63</td>
<td>2.133</td>
</tr>
<tr>
<td>Model 9</td>
<td>+/- 50%</td>
<td>Uniform</td>
<td>8.08</td>
<td>4.720</td>
</tr>
<tr>
<td>Model 10</td>
<td>+/- 20%</td>
<td>Uniform</td>
<td>3.20</td>
<td>1.917</td>
</tr>
<tr>
<td>Model 11</td>
<td>+/- 24%</td>
<td>Uniform</td>
<td>3.99</td>
<td>2.273</td>
</tr>
<tr>
<td>Model 12</td>
<td>+/- 23%</td>
<td>Uniform</td>
<td>3.70</td>
<td>2.200</td>
</tr>
<tr>
<td>Model 13</td>
<td>+/- 22%</td>
<td>Uniform</td>
<td>3.54</td>
<td>2.063</td>
</tr>
</tbody>
</table>

*Conclusions:* Optimism seems independent of the task distribution if they all have the same standard deviation.
The following conclusions were drawn of the data:

1. The optimism was totally determined by the standard deviations. That is, all distributions performed exactly the same once a particular standard deviation was selected.

2. For a given optimism and corresponding standard deviation, the beta distribution has the greatest variability followed by normal, triangular and uniform distributions. Or stated differently for a given variability the optimism gets larger going from beta to normal to triangular to uniform. This in effect means that beta distribution has the narrowest distribution and the uniform distribution the widest.

In Chapter 5 it will be shown that a beta distribution matches well with real data and was used almost exclusively during the model development.

### 4.3.4 Correlation Coefficient Determination

A method was needed to determine the goodness fit of the various suggested equations that were attempting to model the collected data. The correlation coefficient sometimes called the product-moment coefficient of correlation or Pearson’s correlation was used. The correlation coefficient is often used to give a value to dispersion of data around the derived least square equation for the data under study. In our case we have two sets of data. One data set derived from Monte Carlo simulations and the other data set from the suggested model. That is the data sets were Monte Carlo simulations verses optimism and the other the model under study verses optimism. In essence we found two straight lines representing each set of data and compared the two straight lines. The following set of equations was used as outlined in MathWorld from Workfram Research:

\[ s_{xx}^2 = \sum (x_i - \bar{x})^2 \]
Where $ss_{xx}$ is the sum of squares for the simulation data verses optimism.

$$ss_{xx} \equiv \sum (y_i - \bar{y})^2$$  \hspace{1cm} 4-13

Where $ss_{yy}$ is the sum of squares for the model data verses optimism.

$$ss_{yy} \equiv \sum (x_i - \bar{x})(y_i - \bar{y})$$  \hspace{1cm} 4-14

Where $ss_{xy}$ is the sum of squares of the interaction between the two data sets.

$$r^2 = \frac{ss_{xy}^2}{ss_{xx}ss_{yy}}$$  \hspace{1cm} 4-15

Where $r$ is the correlation coefficient and was used to compare one suggested model to another. When $r$ equals one the correlation is perfect.

### 4.4 Model Development

The plan was to propose a mathematical model that closely matches a dynamic CD schedule. The idea was to gain insights as to what parameters drive optimism and by how much. The baseline for comparison of the research results was the deterministic Critical Path Method (CPM)/Precedence Diagramming Method (PDM) generated schedule. Although there is a wide range of software application programs available, Microsoft Project was used exclusively in this research due to its wide spread use. Optimism is defined as the percentage difference between the time the project actually takes to the time initially predicted by CPM/PDM or the critical path length divided by the critical path length. The concept is if a project finishes later than originally predicted, the original schedule was optimistic. Optimism can be a negative number, which is interpreted as pessimism. Here the project completed earlier than expected and the original schedule was pessimistic. During this research the optimism of every analyzed
completed project was a positive number. All results were normalized. The goal was to propose a model that would determine a percent of optimism of the determined baseline deterministic critical path. This then could be used on any project schedule.

The mathematical model evolved through eight steps. The model development started with a set of simple and idealist models with a list of assumptions. This list of assumptions reduced the variables driving optimism but they also took the models further away from real world schedules. As the model was developed, these assumptions were eliminated producing a model representative of many real world CD schedules. The assumptions were:

1. All merge points have the same number of paths into them.
2. Each path has the same overall length.
3. Each path has the same number of tasks.
4. Each task has the same duration.
5. The task variance is limited the task duration variance of the percent under the mode plus the percent over the mode always equaling 100 percent.

**4.4.1 Full Factorial Design with 4 Factors and 3 Treatments**

To begin the construction of the model, four factors that showed a marked influence on project schedule optimism in preliminary investigations were examined in greater detail. Those factors were the number of merge points (M), the number of paths (P) into a merge point, the total number of tasks (T) and the shapeness (S) of the task duration. Shapeness was defined as the percent over the mode divided by the sum of the percent under the mode and the percent over the mode. For example, 75 percent over the mode and 25 percent under the mode would have a shapeness of 0.75. The goal was to
determine if these factors were actually of significance in determining schedule optimism as well as was there any interactions of these factors that also contributed to optimism? A full factorial design experiment was selected.

A sample CD project is shown in Figure 4.1. This particular project has a total of 32 tasks \((T = 32)\), 2 merger points \((M = 2)\) and 4 paths \((P = 4)\) leading into each merger point. Note that in Figure 4.1, SW and HW stand for software and hardware tasks, respectively.

Several considerations went into selecting the treatment levels for each factor. One was to ensure each path had the exact same number of tasks to ensure the resulting matrix would be symmetrical. Drawing conclusions would be more complicated if they were not. Second, more than 2 treatment levels were desired to aid in the detection of any nonlinearity. Third, increasing the treatment levels quickly increases the number of simulations to be run. Lastly, the treatment levels needed to be real world. Three treatment levels were selected. The treatment levels selected for each of factors were as follows:

1. Number of merge points: 1, 2 and 4
2. Number of paths: 2, 4 and 8
3. Number of tasks: 32, 64 and 96
4. Task shapeness (under mode/over mode): 25%/75%, 50%/50% and 75%/25%

This resulted in a full factorial design with four factors and three treatments for each factor to answer the questions stated at the start of this step. That is, what impact do these factors have on the project schedule optimism and are there any interactions between the factors?
There are a total of $3 \times 3 \times 3 \times 3 = 81$ combinations. For each combination, three repetitions were run. This leads to a total of 243 simulation runs. Monte Carlo simulations were run on each combination of M, T, P and S. The results are shown in Table 5.1. Reviewing the data shows that optimism ranged from a $-3.87\%$ to a $+26.25\%$. 
Sample Network:

- 2 Merge points (M)
- 4 Paths per merge point (P)
- 32 Total number of tasks (T)

Assumptions:
- All tasks have the same duration length
- All tasks distributions are the same

Figure 4-1 CD Scheduling Sample
An analysis of variance (ANOVA) table was then prepared from the simulation data to determine the significance of the factors and any interactions. The ANOVA table is shown in Table 5.3. Non-linear regression analysis was used on the significant factors and the interactions to find a model that fit the data and predict the outcome of the project. The correlation coefficient was calculated on each suggested model to determine the goodness of fit. The final result of step one was a good model that predicts the optimism of the project schedule under the stated assumptions.

The model after step 1 was the following:

\[ \text{Optimism} = 1.36M + 1.42P - 0.053T + 20.9S - 10.7 \]  

4.4.2 Merge Point Contribution to Optimism

The second step takes advantage of the observation that each merge point section can usually be considered independent of one another, which allows each merge point section to be analyzed individually. Using this concept the optimism of each merge point section could be found by using the model proposed in equation 4-16. To complete this step a method had to be determined on how to combine the results from each merge point section. It was observed that merge point sections that have a longer critical path should have a greater impact on the overall optimism of a project. A weighted average equation as shown as equation 4-17 was developed that took into consideration each merge point section’s optimism and its critical path length. An experiment was devised to show that this relationship is true. Those results are in Chapter 5.

\[ \text{Overall Optimism} = \alpha = \frac{L_1\alpha_1 + L_2\alpha_2 + \cdots + L_n\alpha_n}{L_1 + L_2 + \cdots + L_1 + \cdots + L_n} \]
Where \( L_i \) = Critical Path Length into each merge point and \( \alpha_i \) = Optimism into each merge point.

### 4.4.3 Full Factorial Design 3 Factors and 3 Treatments

Section 4.4.2 determined that each merge point section could be analyzed by itself and a method was developed on how to combine the results of each merge point section. This section re-looks at the model developed in Section 4.4.1 in that there is one less factor to consider (i.e., the number of merger points \( M \) equals one). As a result, the full factorial design in step one was reduced to 3 factors and 3 treatments full factorial design. The treatment levels of the 3 remaining factors are kept the same. The three samples of each combination were maintained. The resulting data set for the 3-factor 3-treatment design was obtained by selecting the appropriate cells from the 4-factor 3-treatment design. Again a non-linear regression analysis was run on this data set to arrive at a revised model for a merge point section. A good fit was found in equation 4.18. Section 5.2.3 give the details in the development of this equation.

\[
\text{Optimism} = 0.97P - 0.036T + 20.25S - 8.16
\]

### 4.4.4 Number of Tasks Impact

This section takes another look at how the number of tasks in a project impacts the project’s optimism. The model proposed so far (equation 4-18) is a linear combination of the main factors. The model proposed simply has the number of tasks in a merge point section multiplied by a negative constant (the term \(-0.036T\)). That is, as the number of tasks increase, the optimism goes down. This may seem reasonable over the selected number of tasks chosen to develop the model but as the number of tasks
increases, it doesn’t seem reasonable that this linear relationship holds. An extended set of simulations was run to cover an extended range of the number of tasks. The original sets of tasks were 32, 64 and 96. Two additional sets of total number of tasks were added. They were 8 total tasks and 16 total tasks. The same number of parallel paths as before was used. They were two parallel paths, four parallel paths and eight parallel paths. All of the other variables were held constant to include one merge point; all tasks with the same distribution; all tasks have the same length; and all tasks with the same shapeness. The results were plotted and indeed the relationship is not linear. The previous non-linear regression technique was used on the simulation data to determine a good fit. The plotted results suggested a power curve. A number of models with a power term were tried. The results showed that optimism followed a power curve to a power of -0.49, -0.50 and -0.49 or the reciprocal of the square root of the number of tasks. These curves follow so closely to the simulation data that when plotted they lay virtually on top of one another. Using the finding that optimism in a schedule follows the reciprocal square root of the total number of tasks, a non-linear regression analysis was run on the data to produce the revised model. The one producing very good results is shown in equation 4-19.

\[
Optimism = 0.97P + \frac{30.7}{\sqrt{T}} + 20.2S - 14.6
\]

In Chapter 5 the data from the experiments run that determined the reciprocal square root nature of the number of tasks is shown. This clearly shows the reciprocal square root nature of the total number of tasks in a schedule. The revised model was then compared to the Monte Carlo simulation data. This data is also shown in Chapter 5.
4.4.5 Interaction Between Tasks and Paths

In this section the interaction between the number of paths and the number of tasks was investigated further. It was determined in Section 4.4.1 that there was an inter-relationship between these two parameters. Using data from Section 4.4.4 a number of suggested relationships was investigated. The same technique of non-linear regression as used in previous steps was used to determine a good fit. A number of models were tried. The entire set of the Full Factorial 3-Factor 3-Treatment data set was used. A model with P divided by the square root of T provided a very good fit. That result is equation 4-20.

\[
Optimism = \frac{6.91P}{\sqrt{T}} + 20.3S - 10.4
\]

Examination of this equation finds that some adjustment is necessary to cover merge point sections that have only one path. Investigation up to now has shown the drivers of optimism are the merge point phenomenon, shapeness of the task durations and the number of tasks. When the shapeness is symmetrical e.g. 25% under mode and 25% over mode or \( S = 0.5 \), Equation 4-20 shows the optimism is driven by the number of merge points and number of tasks. However, we have also observed that when there is no merge point or when \( P = 1 \) and when \( S = 0.5 \), there is no optimism. Assuming this merge point section also has only one task, equation 4-20 predicts an optimism of about 6.91%, which we know is not true.

Two adjustments were made to the model to account for these low-end conditions. One was to replace the P term with a P-1 term and then run the non-linear regression analysis. This will ensure that when \( P = 1 \), that term is zero. This also allows shapeness to continue to have its influence on merge point sections of one path, which we
have observed is indeed the case. The second adjustment is an observation obtained from Section 4.4.4 during the investigation of the number of tasks impact. That investigation clearly shows that the optimism in a schedule follows reciprocal of the square root of the number of tasks. However, the factor being multiple by this relationship represents the impact of the number of paths. Up to this point that relationship had been assumed to be linear. Plotting these factors shows that this relationship is slightly non-linear. A non-linear regression analysis was conducted on these multiplying factors, which showed this relationship was actually at power function of \( P \). That is, optimism increases in relationship to a power of \( P \). Making these two adjustments and rerunning the non-linear regression analysis on the Full Factorial 3 Factors and 3 Treatments data set a revised model was produced as shown as Equation 4-21.

\[
Optimism = 12.00 \frac{(P - 1)^{0.79}}{\sqrt{T}} + 20.24S - 10.23
\]

4-21

In Chapter 5 the results of the experiments that show the non-linear nature of the number of paths are given. Also, there is a comparison of Equation 4-20 to Equation 4-21, which shows a good improvement of Equation of 4-21 over 4-20.

4.4.6 Varying Lengths of Parallel Paths

The next two sections incorporate the impact of unequal parallel path lengths, unequal task durations and a different number of tasks per path. The approach taken was to account for these situations by calculating an effective number of paths or an “effective \( P \)” in the model. The approach resulted from observing that optimism was the greatest when all parallel paths were of equal length with the same number of tasks and each task in a particular path was of the same duration. As a parallel path was shortened in

132
duration from the critical path, the optimism decreased. This may seem intuitive but in
Chapter 5 Monte Carlo simulation data will show how dramatic this effect is. Uneven
task durations have the opposite effect on optimism. Optimism is the least when all task
durations in a parallel path have the same duration. As the individual task durations
increased in variance, the optimism increased. This may not seem as intuitive but results
will be shown in Section 4.4.7 as to its impact. In this section a method was found to
incorporate uneven parallel paths into the model by calculating an effective number of
paths or an “effective P” for each merge point section of the CD schedule. In Section
4.4.7 uneven task duration lengths and differing number of tasks per path are
incorporated into the model by further adjusting the effective number of paths or an
“effective P.”

In real life, having all paths of a CD project connect into a merge point with the
same length would be rare. An experiment was design to analyze this effect. Monte
Carlo simulations were run on two parallel paths into one merge point with one path
gradually being reduced. It was found that the optimism not only greatly dependent on
the length difference between the two paths but also on the uncertainty of the estimate of
the task duration length. Three sets of uncertainty were subjected to Monte Carlo
simulations and plotted. They were +/- 50%, +/- 25% and +/- 10%. Here uncertainty is
defined as the sum of the absolute values of the uncertainty under the mode and the
uncertainty of the over the mode divided by 100. That is, an uncertainty of +/- 25%
would have a value of 0.5. The data obtained from the simulations was plotted which
suggested a number of possible models. These models were subjected to nonlinear
regression analysis as discussed in Section 4.3.2. The results of the non-linear regression

133
analysis found that each set closely followed a negative power of “e.” A model providing
a good fit to the data is shown as equation 4-22.

\[ Effective\ Path = 1.02e^{-8.73x/y} \]  

Where \( x \) = the percent reduction in length of parallel path from the critical path
\( y \) = The percent uncertainty in the estimate of the task duration

Plotting this equation against the simulation data shows parallel paths of different
lengths falls off quickly as the uncertainty of the task duration estimate is reduced. In the
limit if there is no uncertainty in the path length duration estimate, there would no
contribution to optimism. The critical path length would determine the outcome. The
details of the analysis are given in Chapter 5. The conclusion is that when a parallel path
is more than 10% less than the critical path, that path can usually be discounted as
contributing to any optimism. Note also that the factor in front of “e” in Equation 4-22 is
1.02 or very close to 1.0 and as result, will be used in the final model.

The above analysis was on two parallel paths. In a more than two parallel paths
merge point section; each parallel path can be analyzed compared to the critical path to
determine an effective path for those two parallel paths using Equation 4-22. The next
task was to combine the results of each of these calculations to determine an effective
path length for the merge point section under consideration. A number of models were
suggested and the one providing a good fit is shown as Equation 4-23. This was
suggested from as early as Equation 4-16 where the optimism contributed by parallel
paths was determined by the term 0.97\( P \) or simply a one to one relationship between the
number of paths and optimism.

\[ P_i = R_{i1} + R_{i2} + \cdots + R_{ij} + \cdots + R_{ip} \]  

4-23
Where $P_i = \text{The effective path length of all parallel paths p into merge point i}$

$R_{ij} = \text{The contribution that parallel path j makes to the effective path length compared to the critical path.} \ R_i$ is reserved for the critical path and always equal to 1. $R_{ij}$ is determined from Equation 4-2.

In summary, this section eliminates the assumption that all parallel paths into a merger point needs to be the same length.

4.4.7 Varying Task Durations

In this section the assumption that all task duration lengths have to be the same and all paths with an equal number of tasks was challenged. In real life of CD projects having all task durations of the same duration would be rare. As shown in Chapter 5 investigations of two parallel paths finds that optimism is the least when both parallel paths are exactly the same number of tasks and each task is the same length. Optimism increases when reducing the number of tasks of one path verses the other or changing the individual task lengths while keeping the total length the same.

To further investigate the impact of varying task duration lengths a set of Monte Carlo simulations was conducted in which two equal length paths had a variety of individual task durations. Those results were plotted and analyzed. It was found that the optimism varied in direct relationship to the square root of the sum of the variances of all the tasks under consideration or the standard deviation of the set of tasks under consideration. Further, once the data is normalized, the standard deviation of all the tasks in the pair of paths under consideration is essentially equal to the optimism of the pair. This data was then plotted. One was the simulation data and the other was the normalized standard deviation of the tasks under consideration. Not only are the
relationship between the simulation and normalized standard deviation linear, the two are essentially equal.

The normalization process requires finding the least standard deviation of the pair of paths and comparing it to the set under consideration. The least standard deviation occurs when all the tasks are of equal length. This is found by dividing the duration length of the path under consideration by the number of tasks in that path. This makes all the tasks equal and the standard deviation can be calculated from this set. The result is the smallest the standard deviation will ever be. These relationships are shown in equations 4-24 and 4-25. See Section 4.3.4.2 “Beta Distribution“ for a further explanation. Equation 4-24 shows the percent increase a particular path has from what it would be if all the tasks were of equal duration. Note that the path identified as path one is always the critical path for the merge point section under consideration.

\[ D_{ij} = \frac{stddev}{stddev_{\text{min}}} \]  

\[ stddev = \left[ \sum_{i=1}^{t_i} \left( \frac{t_{\text{max},i} - t_{\text{min},i}}{6} \right)^2 \right]^{\frac{1}{2}} \]  

Where \( D_{ij} \) = the percent increase in optimism over a path with all tasks of equal length. Subscript \( i \) is the merge point section under investigation. Subscript \( j \) is the task duration number of the path under investigation. \( D_{i1} \) is reserved for the critical path.

\( \text{stddev}_{\text{min}} = \) Standard deviation when all tasks are of equal length

\( t_{\text{max}} = \) Maximum task length \( i \)

\( t_{\text{min}} = \) Minimum task length \( i \)
Although a beta distribution was used in the simulation runs, Equation 4-25 is of a general nature and has been used successfully on a variety of distributions. In our case using the percent over and the percent under the mode finds \( t_{\text{max}} \) and \( t_{\text{min}} \). This then is the impact on optimism for that particular path for varying task durations. The divisor in Equation 4-25 is 6 since by definition we have defined \( t_{\text{max}} \) and \( t_{\text{min}} \) as 100% and 0% estimates. The number 6 has been the historic number and was used in the original offering of PERT. However, some (Moder, Phillips et al. 1983) argue that this number should be 3.2 in light that estimates are more likely 5% and 95% numbers but throughout this research 100% and 0% estimates have been assumed and a divisor of 6 seems appropriate for our model.

The results from this Section and Section 4.4.6 are now combined. The goal stated at the start of Section 4.4.6 was to determine an effective number of parallel paths or an “effective P” to be used in the final model. In Section 4.4.6 we showed that reducing the length of parallel from the critical path follows a negative “e” curve. The total impact to the merge point section under investigation is found by equation 4-23. In the first part of this section we showed that the normalized standard deviation of uneven task durations is directly equal to the percent of optimism. The task was to combine these two results. A number of combinations were considered but the one in Equation 4-26 was straightforward and provided good results.

\[
\begin{align*}
P_i &= D_{i1} + R_{i2}D_{i2} + \cdots + R_{ij}D_{ij} + \cdots + R_{ip}D_{ip} \\
\end{align*}
\]

Where \( P_i \) = Effective path length of all parallel paths p into merge point i

\( D_{ij} \) = Contribution of each parallel path because of its length
\( R_{ij} \) = Contribution of each parallel path because of its varying task durations.

Note that the D’s account for the uneven task durations in a particular path and the R’s account for that particular path being shorter than the critical path. The derived number of P in Equation 4-26 is therefore an effective number of paths or the “effective P” into the merge point section under consideration.

In summary, this section eliminates the assumption that all parallel paths have an equal number of tasks and they all had to be of the same length.

4.4.8 Scaling the Task Duration Distribution

Up to this point the percent under the mode plus the percent over the mode always equaled 100 percent. For example, in the initial four and three way full factorial analyses both had three treatments of 25%/75%, 50%/50% and 75%/25%. This step eliminates that restriction. To analyze this impact a two parallel path project with 4 equal tasks in each path was selected. All parameters were kept the same except for the variability of the tasks. It was observed that the most optimistic condition occurs when the under mode percent was zero and the most pessimistic condition occurs when the under mode percent was 100%. These two along with 50% were set as the lower bound and the over mode percent was varied. The details are shown in Chapter 5. The plots show clearly that there is a linear relationship between optimism and the sum of the under mode and over mode percents. Not surprisingly the zero under mode curve passed through zero. After all, when both the under mode and over mode percentages equals zero there is no optimism.

The following conclusions were drawn from this data:
1. When $U$ was set to 0% (the least possible), the normalized optimism directly equaled $(O + U)/100$.

2. When $U$ was set to 100% (the maximum possible), optimism followed directly $(O + U)/100$ but offset by 25.7%. In other words the two conditions are parallel but offset.

3. When $U$ was set to 50%, the results were half way between the two extremes.

4. The above suggested the factor in Equation 4-27 should be added to the model to cover task durations that do not have $O$ and $U$ that adds to 100%. This factor will be labeled $B$.

$$B = \frac{(O + U)}{100}$$  \hspace{1cm} 4-27

In summary, this section eliminates the assumption that task duration variance of the percent under the mode plus the percent over the mode always equaled 100 percent.

### 4.4.9 Final CD Scheduling Model (CDSM)

The final CDSM is a culmination of the previous eight sections. The final model no longer has any restrictions listed in Section 4.4 which were:

1. All merge points have the same number of paths into them (Section 4.4.2).
2. Each path has the same overall length (Section 4.4.6).
3. Each path has the same number of tasks (Section 4.4.7).
4. Each task has the same duration (Section 4.4.8).
5. The task variance is not limited the task duration variance of the percent under the mode plus the percent over the mode always equaled 100 percent.
The final CDSM is shown in Figure 4-2. This model was then tested against a varied number of typical CD projects and also used on completed real CD projects to predict the outcome. Those results are in Sections 4.5 and 4.6 and the details are in Chapter 5. To use the model, follow the steps recommended after Figure 4-2.
Find overall critical path (CP) length $L_{CP}$ assuming no variability

Locate the merge points (MP) along the CP and analyze the paths into each MP as follows:

1. Determine the length $L$ of each path into the MP with $L_1$ being the CP length.
2. Determine the range in percent of the lower and upper bounds $B$ of tasks into the MP.
3. Compare $L_1$ with $L_2$ and determine the percent reduction $R$ compared to two equal $L_1$'s.
4. Determine the $R$ value for each path $L_3$ through $L_n$ as above.
5. Determine the skewness $S$ of the tasks under consideration.

Where:

- $B = \frac{|L_1 - L_2|}{100}$
- $S = \frac{|O|}{|O| + |U|}$

Where $S$ is a number from 0 to 1 indicating Mode location.

$R_{ij} = e^{17.46 x_{ij} / B}$

Where: $R_{ij}$ = Reduction of path 2 compared to two $L_1$'s into $MP_1$
$x_{ij}$ = Percent less of $L_1$ compared to $L_1$

Figure 4-2 CDSM Part 1
Determine impact of distribution of task lengths $D$ for each path into the MP

$$D_{ij} = \frac{stddev_{ij}}{stddev_{\text{min}}}$$

Determine the total number of tasks $T$ in the paths under consideration

$$P_i = \sum_{j=1}^{p} R_{ij} D_{ij}$$

Where:
- $D_{ij}$ = Impact of task length distribution in path $i$ into MP$_1$
- $stddev = \left[ \sum_{i=1}^{p} \frac{(t_{\text{max}} - t_{\text{min}})}{6} \right]^{1/2}$

$Stddev_{\text{min}}$ = Equation above with all tasks equal length
- $t_{\text{max}}$ = Maximum task length
- $t_{\text{min}}$ = Minimum task length
- $p$ = Number of parallel paths into MP

Where $P_i$ = Optimism reduced when all paths are compared to all paths equal to $L_1$
- $p$ = Number of parallel paths

Determine the effective number of parallel paths $P$ into the MP

Determine the optimism $\alpha$ of this merge point

$$\alpha_i = (12.00 \cdot \frac{(P_i - 1)}{T_i^{0.49}} + 20.24 S - 10.23)B$$

Combine results of each merge point for overall value

$$\alpha = \frac{L_1 \alpha_1 + L_2 \alpha_2 + \cdots + L_n \alpha_n}{L_1 + L_2 + \cdots + L_n}$$

Predict the actual length

$$L_{\text{Predicted}} = L_{CP} (1 + \alpha)$$

Where:
- $\alpha_1$ = Optimism of MP$_1$
- $L_1$ = Critical Path Length into each merge point
- $\alpha_i$ = Optimism into each merge point

Figure 4-3 CDSM Part 2
4.4.9.1 Examine Baseline Schedule

Examine the CD baseline schedule under investigation. It needs to have one task as its start and one task as its finish. All tasks that are not Finish to Start need to be carefully analyzed to understand the impact on the critical path and the development of the reduced schedule. The CDSM was built assuming all tasks were Finish to Start. In general, adding non-Finish to Start tasks to a schedule makes examining that schedule a challenge but it does not invalidate the model.

4.4.9.2 Produce Network Diagram

Produce a PERT (network) diagram on the schedule and determine the critical path. This can be done manually but a software application program like Microsoft Project will make the task on any project schedule particularly a large schedule go much faster. Identify the merge points along the critical path along with all the parallel paths ending into that merge point. Any parallel path that is less than 10 percent of the critical path in a particular section can be eliminated from the investigation as not having a significant impact on optimism.

4.4.9.3 Produce Sub-Schedule of CD Schedule

Produce a sub-schedule of the overall CD schedule, which includes just the critical path and all parallel paths into merge points along the critical path that have not been discarded because of the 10 percent rule. In Section 5.5 “Comparing CDSM with Actual CD Schedule” on average, only a little over six percent of all tasks are critical. This sub-schedule can now be further broken down into sections with each section
containing a merge point on the critical path and all the parallel paths leading into that merge point. The CDSM will be applied to each of these merge point sections.

4.4.9.4 Use CDSM on Each Merge Point Section

Use the CDSM on each merge point section of the sub-schedule. The approach now is to determine the optimism contribution each merge point section makes to the overall schedule optimism. Following the sequence of tasks from top to bottom in Figure 4-2 does this. A number of factors in Figure 4-2 have a double subscript. The first subscript number identifies the merge point section and the second subscript number is usually the path number of the path under investigation or task number of a task in a path. Path number 1 is always reserved for the critical path into the merge point. The desired result by examining each merge point section is two numbers. They are the length of the critical path of that merge point section or $L_i$ and the optimistic contribution of that merge point section, which is identified as $\alpha_i$.

4.4.9.5 Determine Overall Optimism

The overall CD schedule optimism is then determined by combining the results from each section by using Equation 4-17 and also shown second from the bottom of Figure 4-2. From this, the overall predicted schedule duration is found by using the equation at the bottom of Figure 4-2 and Equation 4-28.

$$L_{Predicted} = L_{CP}(1 + \alpha)$$  \hspace{1cm} 4-28

The CDSM has now determined the structural optimism of the project. This is important information for the scheduler and the manager. Assuming the answer is a positive number, which in most cases it will be, an obvious first question is what can be
done about it if anything. They may choose to add a buffer or buffers or if the end is a firm date, they may choose not to continue with the venture. They also may use the suggestions in Section 6.1, which gives ideas of what might be done to restructure the schedule to reduce the optimism along with the magnitude these ideas can have on the delivery date.

4.5 Comparing the Proposed Model with Typical CD Projects

The proposed model was used on a variety of projects that were intended to represent the range of CD projects. The model results were compared with Monte Carlo simulations of the same project. Again three replicates with 10,000 iterations were run and then averaged before comparison. One rather complex project is recorded in this report. The others are summarized in a table. The models ranged from one to multiple merge points, from one single path to multiple paths, from one task per path to a variety of tasks with varying task durations and to a variety of shapeness. The intent was to test the robustness of the proposed model.

4.6 Comparing the Proposed Model on Real Life CD Schedules

The real proof of the model is how well the model works on real projects. Can the model predict the outcome at the start of the project? The approach taken was to locate a series of completed CD projects and obtain a copy of the baseline schedule and compare it to the final completed schedule. All of the selected projects were completed by the same design facility, which helped to prevent introducing other variables. A key parameter in the proposed model is the shapeness of the task durations. This proved to be a powerful influence on the optimism of the baseline schedule. All of the tasks on the initial critical path were analyzed for their shapeness. The results were averaged and
used as the shapeness. A representative project was selected. The proposed model was
used on this project using the computed shapeness. The results were compared to a
Monte Carlo simulation runs on the same project and also to the actual completion
duration of the project. The Monte Carlo simulation was run three times with 10,000
iterations each and then averaged to do the comparisons.

4.7 Methodology Summary

This Chapter was divided into three sections. The first section gave the overall
approach used in this research. The research was divided into five phases. Each of these
phases were discussed which were:

Model Development. This was further divided into eight steps that were
generally sequential but some steps were accomplished concurrently. In some cases a
step was repeated based on data collected in a later step. The approach used in each step
was discussed. The last step culminates with the final model.

Comparing the Proposed Model with Typical CD Schedules.

Comparing the Proposed Model with Real Life CD Schedules.

State Findings and Recommendations. The key value of this research is
intended for decision makers who will be given a more realistic picture of what it will
take to complete the project under consideration. Also, a set of rules of thumb was to be
developed which not only identify the sources of optimism but also the predicted
magnitude of the optimism and what might be done to reduce the impact.

Recommend Future Research.

The actual results of the 5 phases are presented in Chapter 5.
The second section in this chapter was devoted to the ground rules and the assumptions used in the research. This included a description of the dependent and independent variables, the statistics used a discussion on confidence interval and a discussion on the normal, beta and triangular distributions used.

The third section was devoted to the special tools used in the research and in particular, the tools used for the development of the model and the conduct of the project analyses. They included the simulation technique used, non-linear regression, curve fitting and correlation coefficient determination.
Chapter 5

Model Development and Simulation Experiments

5.1 Introduction

In Chapter 4, the overall methodology for the research was described. In this chapter the results of the research are given. In review, the research was conducted in three phases. In Phase 1 a mathematical model was developed that predicts the optimism of a CD schedule. As stated in Chapter 4, optimism is defined as the percentage difference between the time the project actually takes to the time initially predicted by CPM/PDM. The concept is if a project finishes later than originally predicted, the original schedule was optimistic. Optimism can be a negative number, which is interpreted as pessimism. Here the project was completed earlier than expected and the original schedule was pessimistic. In Phase 2 the proposed model was used on typical CD schedules. The results are compared to Monte Carlo simulations of the same schedule. In Phase 3 completed real world CD project schedules are examined and compared to the results of the model. In Chapter 6 the insights derived from this research are listed along with recommendations for future research.

5.2 Model Development

The model development went through an evolution of 8 steps to arrive at the full up model as described in Chapter 4. Here the specifics of the model development are given.
5.2.1 Full Factorial Design 4 Factors and 3 Treatments

In this first step a full factorial design with 4 factors and 3 treatments was run. The four factors were:

1. Number of merge points (M)
2. Number of paths (P) into a merge point
3. Total number of tasks (T)
4. Shapeness (S) of the task duration. Shapeness is defined as the percent over the mode divided by the sum of the percent under the mode and the percent over the mode. For example, 75 percent over the mode and 25 percent under the mode would have a shapeness of 0.75.

The treatment levels selected for each of factors respectively are as follows:

1. Number of merge points: 1, 2 and 4
2. Number of paths: 2, 4 and 8
3. Number of tasks: 32, 64 and 96
4. Task Shapeness (under mode/over mode): 25%/75%, 50%/50% and 75%/25%

The results of the design experiment are shown in Table 5.1. The resulting ANOVA table based on these results is shown in Table 5.2. The analysis shows the four main factors are highly significant as expected. Two interactions (MP and PT) were found significant but to a much lesser degree. MP is between the number of merge points and the number of paths, and PT is between the number of paths and the number of tasks but only when the degree of significance dropped to 95% significance. These two interactions were considered but did not make the initial model. They are again considered in Steps 2 and 4 in the model development and made part of the model.
Table 5-1 Full Factorial Design 4-Factors 3-Treatments Results

<table>
<thead>
<tr>
<th>$M$</th>
<th>$a - 1$</th>
<th>$b - 2$</th>
<th>$c - 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d = 2$</td>
<td>$e = 4$</td>
<td>$f = 8$</td>
<td>$d = 2$</td>
</tr>
<tr>
<td>$p$</td>
<td>$p$</td>
<td>$p$</td>
<td>$p$</td>
</tr>
<tr>
<td>$g = \frac{1}{2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.84</td>
<td>6.25</td>
<td>7.76</td>
<td>12.18</td>
</tr>
<tr>
<td>1.84</td>
<td>6.93</td>
<td>7.69</td>
<td>12.33</td>
</tr>
<tr>
<td>1.85</td>
<td>6.94</td>
<td>7.85</td>
<td>12.35</td>
</tr>
</tbody>
</table>

$h = \frac{1}{4}$ |  |  |  |
| 1.24 | 6.33 | 7.00 | 10.10 | 12.25 | 12.25 |
| 1.33 | 6.36 | 8.65 | 12.33 | 15.15 | 18.75 |
| 1.26 | 6.46 | 7.01 | 10.06 | 15.30 | 7.70 |

$i = \frac{1}{8}$ |  |  |  |
| 1.09 | 6.19 | 7.90 | 10.85 | 13.30 | 10.96 |
| 1.09 | 6.17 | 7.84 | 10.80 | 13.17 | 16.78 |
| 1.13 | 6.16 | 8.03 | 11.00 | 13.10 | 16.93 |
Table 5-2 ANOVA Full Factorial Design 4 Factors 3 Treatments

<table>
<thead>
<tr>
<th>Sources of Error</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>F</th>
<th>( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>78.02</td>
<td>2</td>
<td>39.01</td>
<td>0.34</td>
<td>Significant at 0.01 if ( F \geq 4.61 )</td>
</tr>
<tr>
<td>P</td>
<td>338.46</td>
<td>2</td>
<td>169.23</td>
<td>1.48</td>
<td>Significant at 0.01 if ( F \geq 4.61 )</td>
</tr>
<tr>
<td>T</td>
<td>53.55</td>
<td>2</td>
<td>26.77</td>
<td>0.23</td>
<td>Significant at 0.01 if ( F \geq 4.61 )</td>
</tr>
<tr>
<td>S</td>
<td>493.88</td>
<td>2</td>
<td>246.94</td>
<td>2.16</td>
<td>Significant at 0.01 if ( F \geq 4.61 )</td>
</tr>
<tr>
<td>MP</td>
<td>25.24</td>
<td>4</td>
<td>6.31</td>
<td>0.06</td>
<td>Significant at 0.01 if ( F \geq 3.32 )</td>
</tr>
<tr>
<td>MT</td>
<td>4.24</td>
<td>4</td>
<td>1.06</td>
<td>0.01</td>
<td>ns</td>
</tr>
<tr>
<td>MS</td>
<td>0.43</td>
<td>4</td>
<td>0.11</td>
<td>0.00</td>
<td>ns</td>
</tr>
<tr>
<td>PT</td>
<td>17.85</td>
<td>4</td>
<td>4.46</td>
<td>0.04</td>
<td>Significant at 0.05 if ( F \geq 2.37 )</td>
</tr>
<tr>
<td>PS</td>
<td>1.61</td>
<td>4</td>
<td>0.40</td>
<td>0.00</td>
<td>ns</td>
</tr>
<tr>
<td>ST</td>
<td>0.25</td>
<td>4</td>
<td>0.06</td>
<td>0.00</td>
<td>ns</td>
</tr>
<tr>
<td>MPT</td>
<td>1.18</td>
<td>8</td>
<td>0.15</td>
<td>0.00</td>
<td>ns</td>
</tr>
<tr>
<td>MPS</td>
<td>0.41</td>
<td>8</td>
<td>0.05</td>
<td>0.00</td>
<td>ns</td>
</tr>
<tr>
<td>MTS</td>
<td>0.13</td>
<td>8</td>
<td>0.02</td>
<td>0.00</td>
<td>ns</td>
</tr>
<tr>
<td>PTS</td>
<td>0.40</td>
<td>8</td>
<td>0.05</td>
<td>0.00</td>
<td>ns</td>
</tr>
<tr>
<td>MPTS</td>
<td>0.10</td>
<td>16</td>
<td>0.01</td>
<td>0.00</td>
<td>ns</td>
</tr>
<tr>
<td>Error</td>
<td>18,530.54</td>
<td>$\frac{162}{242}$</td>
<td>114.39</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Where: M = Number of merge points  
\( P \) = Number of paths into a merge point  
T = Total number of tasks  
S = Task shapeness

Armed with the results of the ANOVA table, a model was developed using the non-linear regression technique discussed in Chapter 4. A range of models was tried to include incorporating the interactions between M and P and between P and T identified in the ANOVA table. The correlation coefficient was calculated for each of the tried models to determine their goodness of fit to the data. Several more complicated models provided slightly better correlation coefficients but a simple linear combination provided very good results. The model chosen is shown in equation 5.1.
Optimism = 1.36M + 1.42P − 0.053T + 20.9S − 10.7

Calculation of the correlation coefficient of Equation 5-1 is shown in Table 5-3. Note the differences (delta) between the model and Monte Carlo simulations show a very good match between the model and the simulations.

The model was applied to several project schedules and the results are shown in Table 5-4. The model provides a fairly good match to the Monte Carlo simulations. Equation 5-1 was used to calculate the optimism in the columns titled “Model.” Monte Carlo simulations were run on each of the models and recorded in the column titled “Sim” and averaged in the column titled average percentage. As shown, each model was run three times and averaged to determine the amount of optimism.
Table 5-3 Correlation Coefficient of Step 1 Model

\[ z = x_1 \cdot b_1 + x_2 \cdot b_2 + x_3 \cdot b_3 + x_4 \cdot b_4 + b_5 \]

<table>
<thead>
<tr>
<th>Model</th>
<th>Paths</th>
<th>Tasks</th>
<th>Skewn-b4</th>
<th>Error</th>
<th>b1</th>
<th>b2</th>
<th>b3</th>
<th>b4</th>
<th>b5</th>
<th>Model-b1</th>
<th>Model-b2</th>
<th>Model-b3</th>
<th>Model-b4</th>
<th>Model-b5</th>
<th>Model-b6</th>
<th>Sim-b1</th>
<th>Sim-b2</th>
<th>Sim-b3</th>
<th>Sim-b4</th>
<th>Sim-b5</th>
<th>Sim-b6</th>
<th>Delta-b1</th>
<th>Delta-b2</th>
<th>Delta-b3</th>
<th>Delta-b4</th>
<th>Delta-b5</th>
<th>Delta-b6</th>
<th>81.89</th>
<th>83.4096</th>
<th>80.394</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merge</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.36</td>
<td>1.42</td>
<td>-0.05</td>
<td>20.9</td>
<td>-10.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.08</td>
<td>4.73</td>
<td>-0.34</td>
<td>-1.07</td>
<td>-1.20</td>
<td>1.28</td>
<td>1.13877</td>
<td>1.4392</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paths</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skewn-b4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>32</td>
<td>0.25</td>
<td>2</td>
<td>-2.99</td>
<td>-3.03</td>
<td>-0.04</td>
<td>-9.13</td>
<td>-8.97</td>
<td>81.89</td>
<td>83.4096</td>
<td>80.394</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>32</td>
<td>0.50</td>
<td>5</td>
<td>5.08</td>
<td>4.73</td>
<td>-0.34</td>
<td>-1.07</td>
<td>-1.20</td>
<td>1.28</td>
<td>1.13877</td>
<td>1.4392</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>32</td>
<td>0.75</td>
<td>8</td>
<td>15.97</td>
<td>15.18</td>
<td>-0.79</td>
<td>9.83</td>
<td>9.25</td>
<td>90.94</td>
<td>96.645</td>
<td>85.569</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>64</td>
<td>0.25</td>
<td>2</td>
<td>-3.32</td>
<td>-3.00</td>
<td>0.31</td>
<td>-9.46</td>
<td>-8.94</td>
<td>84.56</td>
<td>89.5329</td>
<td>79.857</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>64</td>
<td>0.50</td>
<td>4</td>
<td>4.75</td>
<td>4.72</td>
<td>-0.03</td>
<td>-1.40</td>
<td>-1.22</td>
<td>1.70</td>
<td>1.95003</td>
<td>1.4793</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>64</td>
<td>0.75</td>
<td>8</td>
<td>15.65</td>
<td>15.23</td>
<td>-0.42</td>
<td>9.50</td>
<td>9.29</td>
<td>88.30</td>
<td>90.2788</td>
<td>86.373</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>96</td>
<td>0.25</td>
<td>4</td>
<td>-2.29</td>
<td>-2.74</td>
<td>-0.45</td>
<td>-8.43</td>
<td>-8.67</td>
<td>73.12</td>
<td>71.1388</td>
<td>75.163</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>96</td>
<td>0.50</td>
<td>4</td>
<td>5.78</td>
<td>5.46</td>
<td>-0.32</td>
<td>-0.37</td>
<td>-0.47</td>
<td>0.17</td>
<td>0.13589</td>
<td>0.2237</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>96</td>
<td>0.75</td>
<td>8</td>
<td>16.67</td>
<td>16.85</td>
<td>0.18</td>
<td>10.53</td>
<td>10.92</td>
<td>114.95</td>
<td>110.867</td>
<td>119.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \text{Correlation Coefficient} = 0.99844 \]
Table 5-4 Step 1 Model Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Merge</th>
<th>Paths</th>
<th>Tasks</th>
<th>Shapeopen Factor</th>
<th>Optimism Simulation</th>
<th>Model</th>
<th>Optimism Simulation</th>
<th>Model</th>
<th>Optimism Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>75/25</td>
<td>0.25</td>
<td>-1.45</td>
<td>-1.73</td>
<td>3.73</td>
<td>3.59</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>25/75</td>
<td>0.75</td>
<td>0.83</td>
<td>0.72</td>
<td>8.87</td>
<td>10.57</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>16</td>
<td>75/25</td>
<td>0.25</td>
<td>-0.35</td>
<td>-0.20</td>
<td>3.64</td>
<td>4.42</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>16</td>
<td>25/75</td>
<td>0.75</td>
<td>0.92</td>
<td>9.68</td>
<td>9.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>24</td>
<td>75/25</td>
<td>0.25</td>
<td>-1.03</td>
<td>0.12</td>
<td>3.71</td>
<td>5.35</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>24</td>
<td>25/75</td>
<td>0.75</td>
<td>0.87</td>
<td>10.57</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5.2.2 Merge Point Contribution to Optimism

An observation was made that the overall schedule may be decomposed into merge point sections that could be analyzed and then recombined for the total impact if the sections were independent of each other. With the assumption that the paths and tasks into a merge point were independent of the paths and tasks of any other merge point, which is a good assumption in real world CD projects, the following equation was proposed to compute the optimism.

\[ \text{Overall Optimism} = \alpha = \frac{L_1\alpha_1 + L_2\alpha_2 + \ldots + L_i\alpha_i + \ldots + L_n\alpha_n}{L_1 + L_2 + \ldots + L_i + \ldots + L_n} \]

5-2

Where \( L_i \) = Critical Path Length into each merge point section and \( \alpha_i \) = Optimism into each merge point.

The net result of the above is the overall optimism of a project schedule is found by analyzing the parallel paths into each merge point along the critical path as a stand-alone project. The task is then for each merge point section: find the length of its critical path or \( L_i \) and the overall optimism or \( \alpha_i \). The overall project optimism can then be found by using equation 5.2.

An experiment with a variety of project schedules was developed to add credence that this relationship is true. Monte Carlo simulations were conducted on these schedules. The schedules are shown in Figure 5-1. Those results are shown in Table 5.5. The figure and table shows seven projects. Projects 1, 2A, 3A and 4A have a single merge point each. The critical path lengths are easily computed. Using Monte Carlo simulations, the amount of optimism was found. These are the building blocks for equation 5-2. Each of projects 2, 3 and 4 has multiple merge points. From here the
critical path lengths was found and with Monte Carlo simulations the amount of optimism was determined. Below these figures of optimism in outlined boxes are the optimisms obtained from using Equation 5-2. There is a very close match between the model and simulation. This gives confidence that Equation 5-2 can be used.

5.2.3 Full Factorial Design 3-Factor 3-Treatment

Since merge points are now accounted for in the decomposed schedule, the factorial design was reduced to 3 factors by eliminating the merge point contribution. Table 5-1 was reduced to the left side of Table 5-6. As in Section 5.2.1 a non-linear approach was used on the data in Table 5-6 to find a relationship between the main factors. Again the interaction between the number of paths and the total number of tasks were considered but they didn’t improve the model much and were not incorporated in the model at this step. Note the interaction between number of merge points and number of parallel paths was dealt with in Section 5.2.2. The correlation coefficient of each model was calculated to determine their goodness of fit to the data. The best model found is shown as equation 5-3. This equation along with equation 5-2 made up the model after Step 3.

\[
\text{Optimism} = 0.97P - 0.036T + 20.25S - 8.16
\]

The model was compared to the simulation results. The results are shown on the right side of Table 5-6.
Figure 5-1 Merge Point Contribution (Models)
<table>
<thead>
<tr>
<th>Merge</th>
<th>Paths</th>
<th>Tasks</th>
<th>Shapeless</th>
<th>Ave Percent</th>
<th>Ave Percent</th>
<th>Ave Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>75/25</td>
<td>0.25</td>
<td>-1.15</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Day Tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>50/50</td>
<td>0.50</td>
<td>3.73</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>25/75</td>
<td>0.75</td>
<td>8.83</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>16</td>
<td>75/25</td>
<td>0.25</td>
<td>-1.02</td>
<td>-1.06</td>
</tr>
<tr>
<td>Model 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/6 Day Tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>16</td>
<td>50/50</td>
<td>0.50</td>
<td>3.69</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>16</td>
<td>25/75</td>
<td>0.75</td>
<td>9.00</td>
<td>8.96</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>24</td>
<td>75/25</td>
<td>0.25</td>
<td>-1.05</td>
<td>-1.04</td>
</tr>
<tr>
<td>Model 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/6/4 Day Tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>24</td>
<td>50/50</td>
<td>0.50</td>
<td>3.64</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>24</td>
<td>25/75</td>
<td>0.75</td>
<td>8.93</td>
<td>8.92</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>32</td>
<td>75/25</td>
<td>0.25</td>
<td>-1.08</td>
<td>-1.03</td>
</tr>
<tr>
<td>Model 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/6/4/3 Day Tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>32</td>
<td>50/50</td>
<td>0.50</td>
<td>3.66</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>32</td>
<td>25/75</td>
<td>0.75</td>
<td>8.92</td>
<td>8.90</td>
</tr>
</tbody>
</table>
**Table 5-6 Full Factorial Design 3-Factor 3-Treatment Results**

<table>
<thead>
<tr>
<th>Factors</th>
<th>Simulation</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>d = 2</td>
<td>e = 4</td>
</tr>
<tr>
<td>g = 32, h = 64, i = 96</td>
<td>j = 75/25</td>
<td>k = 50/50</td>
</tr>
<tr>
<td>j = 75/25</td>
<td>-3.03</td>
<td>0.34</td>
</tr>
<tr>
<td>k = 50/50</td>
<td>1.84</td>
<td>4.73</td>
</tr>
<tr>
<td>l = 25/75</td>
<td>6.70</td>
<td>10.04</td>
</tr>
<tr>
<td>j = 75/25</td>
<td>-3.87</td>
<td>-2.11</td>
</tr>
<tr>
<td>k = 50/50</td>
<td>1.09</td>
<td>2.77</td>
</tr>
<tr>
<td>l = 25/75</td>
<td>6.17</td>
<td>7.92</td>
</tr>
</tbody>
</table>

**Factors**
- $P$ = Number of Paths per Merge Points
- $T$ = Number of Tasks
- $S$ = Skewness

**Formula**

$$Optimism = 0.97P - 0.036T + 20.3S - 8.16$$
5.2.4 Number of Tasks Impact

This section took another look at how the number of tasks in a project impacts the project’s optimism. The model proposed in Equation 5-3 simply has the total number of tasks in a merge point section multiplied by a negative constant (the –0.036T factor). That is, as the number of tasks increase, the optimism goes down. This may seem reasonable over the selected number of tasks chosen to develop the model but as the number of tasks increases this assumption appears unlikely. A new set of experiments was run to investigate this nature. The approach taken was to start with the set of data in Table 5-6 and add two more treatments to the total number of tasks. Table 5-6 has treatments for 32, 64 and 96 total number of tasks. Treatments of 8 and 16 tasks were added. All of the other variables were held constant. Monte Carlo simulations were run on the new treatments. The results are recorded in Table 5-7 and the data was plotted. Indeed the relationship was not linear. Armed with these results a non-linear regression analysis was conducted on this data to determine the relationship between optimism and the number of tasks. A number of relationships were tried but the one that produced an excellent fit was the reciprocal of the square root of the number of total tasks. The actual powers to the total number of tasks are -0.49, -0.50 and -0.49 as shown on Table 5-7. These relationships are also plotted on Table 5-7. They are such a close match that they lay almost on top of one another.

Using this finding, a new model was proposed using the non-linear regression analysis on the Full Factorial 3 Factors and 3 Treatments data set. The relationship that showed a good fit is shown as equation 5-4.
This equation was then compared to the simulation runs in Table 5-8. It needs to be noted that the simulations runs in Table 5-7 to determine the relationship of optimism over a wider range of the number of task assumed a constant shapeness. The simulations used to determine an updated model was done on the entire set of conditions to include three levels of shapeness. This is tabulated in Table 5.8 to include the three levels of shapeness to give a better indication as to how effective this updated model is.
Table 5-7 Analyzing Increasing Numbers of Tasks

P = Number of Paths per Merge Points
T = Number of Tasks
S = Shapeness Beta 50%<50%

**Optimism**

<table>
<thead>
<tr>
<th>d = 2</th>
<th>e = 4</th>
<th>f = 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.63</td>
<td>9.50</td>
<td>18.27</td>
</tr>
<tr>
<td>2.62</td>
<td>6.65</td>
<td>12.97</td>
</tr>
<tr>
<td>1.84</td>
<td>4.73</td>
<td>9.27</td>
</tr>
<tr>
<td>1.28</td>
<td>3.40</td>
<td>6.49</td>
</tr>
<tr>
<td>1.09</td>
<td>2.77</td>
<td>5.36</td>
</tr>
</tbody>
</table>

Note: Each cell run 3 times and averaged.

**Regression Analysis Results**

For 2 paths

\[ \text{Optimism} = 10.13T^{-0.49} \]

For 4 paths

\[ \text{Optimism} = 26.65T^{-0.50} \]

For 8 paths

\[ \text{Optimism} = 51.05T^{-0.49} \]
**Table 5-8 Incorporating a Wider Range of Number of Tasks**

Factors:
- $P$: Number of Paths per Merge Points
- $T$: Number of Tasks
- $S$: Shapeness

<table>
<thead>
<tr>
<th>Simulation</th>
<th>$P$</th>
<th>$d = 2$</th>
<th>$e = 4$</th>
<th>$f = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$g = 32$</td>
<td>$i = 64$</td>
<td>$i = 96$</td>
<td>$i = 64$</td>
</tr>
<tr>
<td></td>
<td>$j = 75/25$</td>
<td>$k = 50/50$</td>
<td>$l = 25/75$</td>
<td>$j = 75/25$</td>
</tr>
<tr>
<td></td>
<td>$-3.03$</td>
<td>$0.34$</td>
<td>$4.38$</td>
<td>$-3.59$</td>
</tr>
<tr>
<td></td>
<td>$1.84$</td>
<td>$4.73$</td>
<td>$9.27$</td>
<td>$1.28$</td>
</tr>
<tr>
<td></td>
<td>$6.70$</td>
<td>$10.04$</td>
<td>$15.18$</td>
<td>$6.38$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>$P$</th>
<th>$d = 2$</th>
<th>$e = 4$</th>
<th>$f = 8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$g = 32$</td>
<td>$i = 64$</td>
<td>$i = 96$</td>
<td>$i = 64$</td>
</tr>
<tr>
<td></td>
<td>$j = 75/25$</td>
<td>$k = 50/50$</td>
<td>$l = 25/75$</td>
<td>$j = 75/25$</td>
</tr>
<tr>
<td></td>
<td>$-2.16$</td>
<td>$-0.21$</td>
<td>$3.67$</td>
<td>$-3.74$</td>
</tr>
<tr>
<td></td>
<td>$2.91$</td>
<td>$4.85$</td>
<td>$8.73$</td>
<td>$1.32$</td>
</tr>
<tr>
<td></td>
<td>$7.97$</td>
<td>$9.91$</td>
<td>$13.79$</td>
<td>$6.38$</td>
</tr>
</tbody>
</table>

Note: Each cell run 3 times

Optimization: $0.97P + \frac{30.7}{\sqrt{T}} + 20.2S - 14.6$
5.2.5 Interaction Between Tasks and Paths

In this section the interaction between the number of paths and the number of tasks was investigated further. In Chapter 4 an initial improvement model was given as Equation 5.5.

\[
Optimism = \frac{6.91P}{\sqrt{T}} + 20.3S - 10.4
\]  

A further improvement was proposed in Chapter 4 to accommodate low-end performance of the model. The first adjustment was to replace the P factor with a P-1 factor to ensure the model reports no optimism from the merge point phenomena when there is a single path merge point section. The second adjustment comes from a further study of Table 5-7 where the increasing number of tasks was analyzed. The results are shown in Table 5-9. Up to now the relationship between number of paths and optimism had been assumed to be linear. Table 5-9 shows that there is a slight non-linearity to the relationship. Using these results a new non-linear regression analysis was run on the Full Factorial 3 Factors and 3 Treatments data set to arrive at a revised model shown in Equation 5-6.

\[
Optimism = 12.00 \frac{(P-1)^{0.79}}{\sqrt{T}} + 20.24S - 10.23
\]

This data was tabularized in Table 5-9. Both Equations 5-5 and 5-6 were used to calculate the optimism and then their answers were compared to the Monte Carlo simulations. The variation of Equation 5-5 from the simulation varied from 0.0% to 1.27% where the variation of Equation 5-6 from the simulation varied from 0.01% to 0.43% or a good improvement over Equation 5-5.
Table 5-9 Optimism Versus Number of Paths

<table>
<thead>
<tr>
<th>Number of Paths</th>
<th>Optimism (Simulation)</th>
<th>Optimism (Regression Analysis Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>10.13</td>
<td>10.80</td>
</tr>
<tr>
<td>4</td>
<td>26.65</td>
<td>26.01</td>
</tr>
<tr>
<td>8</td>
<td>51.05</td>
<td>51.23</td>
</tr>
</tbody>
</table>

Optimism = 10.80(P - 1) ^ 0.80
Table 5-10 Incorporating Interactions between Tasks and Paths

<table>
<thead>
<tr>
<th>Factors</th>
<th>Simulation</th>
<th>Model</th>
<th>Difference From Simulation</th>
<th>Model</th>
<th>Difference From Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$d = 2$</td>
<td>$e = 4$</td>
<td>$f = 8$</td>
<td>$d = 2$</td>
<td>$e = 4$</td>
</tr>
<tr>
<td>$P$</td>
<td>$k = 75/25$</td>
<td>$j = 50/50$</td>
<td>$l = 25/75$</td>
<td>$k = 50/50$</td>
<td>$l = 25/75$</td>
</tr>
<tr>
<td>$S$</td>
<td>$g = 32$</td>
<td>$h = 64$</td>
<td>$i = 96$</td>
<td>$g = 32$</td>
<td>$h = 64$</td>
</tr>
<tr>
<td></td>
<td>-3.03</td>
<td>0.34</td>
<td>4.38</td>
<td>-2.16</td>
<td>-0.21</td>
</tr>
<tr>
<td></td>
<td>1.84</td>
<td>4.73</td>
<td>9.27</td>
<td>2.91</td>
<td>4.85</td>
</tr>
<tr>
<td></td>
<td>6.70</td>
<td>10.04</td>
<td>15.18</td>
<td>7.97</td>
<td>9.91</td>
</tr>
<tr>
<td></td>
<td>-3.59</td>
<td>-1.43</td>
<td>1.78</td>
<td>-3.74</td>
<td>-1.80</td>
</tr>
<tr>
<td></td>
<td>1.28</td>
<td>3.40</td>
<td>6.49</td>
<td>1.32</td>
<td>3.26</td>
</tr>
<tr>
<td></td>
<td>6.38</td>
<td>8.65</td>
<td>12.25</td>
<td>6.38</td>
<td>8.32</td>
</tr>
<tr>
<td></td>
<td>6.17</td>
<td>7.92</td>
<td>10.88</td>
<td>5.68</td>
<td>7.62</td>
</tr>
</tbody>
</table>

$Optimism = 0.97P + \frac{30.7}{\sqrt{I}} + 20.2S - 14.6$

$Optimism = 12.00 \left( \frac{P-1}{\sqrt{I}} \right)^8 + 20.24S - 10.23$
5.2.6 Varying Lengths of Parallel Paths

The next two sections incorporate the impact of unequal parallel path lengths, unequal task durations and a different number of tasks per path. The approach taken was to account for these conditions in the model by calculating an effective number of paths or an “effective P.” Section 4.4.6 gave the methodology for incorporating unequal parallel paths and stated the Equation 5-7 quantified the impact.

$$ EffectivePath = 1.02e^{-8.73x/y} \quad 5-7 $$

Where $x$ = the percent reduction in length of parallel path from the critical path

$y$ = The percent uncertainty in the estimate of the task duration

Section 4.4.6 also gave the methodology for combining multiple parallel paths into one merge point as shown in Equation 5-8.

$$ P_i = R_{i1} + R_{i2} + \cdots + R_{ij} + \cdots + R_{ip} \quad 5-8 $$

Where $P_i$ = The effective path length of all parallel paths $p$ into merge point $i$

$R_{ij}$ = The contribution that parallel path $j$ makes to the effective path length compared to the critical path. $R_{i1}$ is reserved for the critical path and always equal to 1. $R_{ij}$ is determined from Equation 4-2.

Equation 5-7 was developed as a result of an experiment designed to investigate the impact of uneven parallel paths into one merge point. The experiment started with two parallel paths with an equal number of tasks and initially the same overall length. The experimental set up is shown in Figure 5-2.
Monte Carlo simulations were run on the two parallel paths into one merge point with one path gradually being reduced. It was found that the optimism not only greatly depended on the length difference between the two paths but also on the uncertainty of the estimate of the task duration length. Three sets of uncertainty were subjected to Monte Carlo simulations and plotted as shown in Tables 5-11, 5-12 and 5-13. They were +/− 50%, +/− 25% and +/− 10%. Here uncertainty is defined as the sum of the absolute values of the uncertainty under the mode and the uncertainty of the over the mode divided by 100. That is, an uncertainty of +/− 25% would have a value of 0.5. The data obtained from the simulations was plotted which suggested a number of possible models. These models were subjected to nonlinear regression analysis and produced Equation 5-7. This equation is also plotted against the simulation data in the tables below. Note that the impact of parallel paths of different lengths falls off quickly as the uncertainty of the task duration estimate is reduced. In the limit if there is no uncertainty in the path length duration estimate, there would no contribution to optimism. The critical path length would determine the outcome. Each of the three charts has an arrow pointing to a point...
on the abscissa that represents a 90 percent reduction in optimism from where the two path lengths were of equal length. They range from 2.7% to 12.7%. The conclusion is that when a parallel path is more than 10% less than the critical path, that path can usually be discounted as contributing to any optimism.

Table 5-11 Varying Lengths of Parallel Paths with Shapeness +/- 50%

<table>
<thead>
<tr>
<th>Length of Tasks in Path 2</th>
<th>Percent Reduction of Path 2/1</th>
<th>Optimism Normalized</th>
<th>Least Square Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>0%</td>
<td>3.63</td>
<td>1.00</td>
</tr>
<tr>
<td>9.8</td>
<td>2%</td>
<td>2.78</td>
<td>0.77</td>
</tr>
<tr>
<td>9.5</td>
<td>5%</td>
<td>1.61</td>
<td>0.44</td>
</tr>
<tr>
<td>9.0</td>
<td>10%</td>
<td>0.44</td>
<td>0.12</td>
</tr>
<tr>
<td>8.5</td>
<td>15%</td>
<td>0.13</td>
<td>0.04</td>
</tr>
<tr>
<td>8.0</td>
<td>20%</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>7.5</td>
<td>25%</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>7.0</td>
<td>30%</td>
<td>0.02</td>
<td>0.00</td>
</tr>
</tbody>
</table>

\[ x = 4 \pm 50\% \]

Note: Asterisked number is the point where optimism drops to 10% of two equal path durations.

\[ \text{Effective Path} = 1.02e^{-8.73x/y} \]
### Table 5-12 Varying Lengths of Parallel Paths with Shapenes +/- 25%

<table>
<thead>
<tr>
<th>Length of Tasks in Path 1</th>
<th>Percent Reduction in Path 2/1</th>
<th>Optimism</th>
<th>Normalized Optim</th>
<th>Least Square Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>0%</td>
<td>1.82</td>
<td>1.00</td>
<td>1.02</td>
</tr>
<tr>
<td>9.8</td>
<td>2%</td>
<td>0.98</td>
<td>0.54</td>
<td>0.50</td>
</tr>
<tr>
<td>9.5</td>
<td>5%</td>
<td>0.30</td>
<td>0.17</td>
<td>0.18</td>
</tr>
<tr>
<td>9.0</td>
<td>10%</td>
<td>0.03</td>
<td>0.01</td>
<td>0.03</td>
</tr>
<tr>
<td>8.5</td>
<td>15%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
</tr>
<tr>
<td>8.0</td>
<td>20%</td>
<td>0.03</td>
<td>0.02</td>
<td>0.00</td>
</tr>
<tr>
<td>7.5</td>
<td>25%</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>7.0</td>
<td>30%</td>
<td>0.02</td>
<td>0.01</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Note: Asterisked number is the point when optimism drops to 10% of two equal path durations.

**Effective Path = \(1.02e^{-0.73x/\gamma}\)**
The above analysis was on two parallel paths. In a more than two parallel paths merge point section; each parallel path can be analyzed compared to the critical path to determine an effective path for those two parallel paths using Equation 5-7. Equation 5-8 was suggested from Equation 5-1, which accounts for multiple paths simply by a 0.97P term or simply a one to one relationship between the number of paths and optimism.
5.2.7 Varying Task Durations

In this section the assumption that all task duration lengths has to be the same and all paths with an equal number of tasks was challenged. Chapter 4 explained the methodology that derived the following relationships to account for a variety of task durations and a number of tasks in a parallel path unequal to the number in the critical path. The relationship that accounts optimism of a pair parallel paths with one being the critical path are repeated here as Equations 5-9 and 5-10.

\[ D_{ij} = \frac{\text{stddev}_i}{\text{stddev}_{\min}} \]  

\[ \text{stddev}_i = \left[ \sum_{j=1}^{t} \left( \frac{t_{\text{max}_i} - t_{\text{min}_i}}{6} \right)^2 \right]^{\frac{1}{2}} \]

Where \( D_{ij} \) is the percent increase in optimism over a path with all tasks of equal length. Subscript \( i \) is the merge point section under investigation. Subscript \( j \) is the task duration number of the path under investigation. \( D_{11} \) is reserved for the critical path.

\( \text{stddev}_{\min} \) = Standard deviation when all tasks are of equal length

\( t_{\text{max}_i} \) = Maximum task length \( i \)

\( t_{\text{min}_i} \) = Minimum task length \( i \)

The approach taken to arrive at this conclusion was to design an experiment with two paths leading into one merge point project where the number of tasks on both paths is different but the overall length of each parallel path remained the same. This in effect varied task duration. Monte Carlo simulations were run on the project schedule with varying number of tasks. The computational results indicate that the relationship between task lengths is non-linear. Furthermore, the impact on optimism was the least
when all the tasks were of the same length. A variety of parameters of the task duration were considered when it was found that the normalized standard deviation of the paths directly impacted the optimism. Those results were plotted and analyzed as shown in Table 5-14.

Upon further investigation it was found that the optimism varied in direct relationship to the square root of the sum of the variances of all the tasks under consideration or the standard deviation of the set of tasks under consideration. The experiment used to investigate this relationship is shown in Figure 5-3.

![Figure 5-3 Varying Task Durations](image)

**Conditions:** All task durations are beta distributions 25% under/25% over mode
Each simulation cell was run 3 times 10,000 iterations and then averaged
All path lengths are 20 days long

The experimental results are shown in Tables 5-15 and 5-16.
Conditions: Each path in each model has the same number of days e.g. each path in Model 1 has 50 days.
- HW1 tasks varied from 10 to 50 days and all SW1 tasks are 10 days. All conditions are the same except varying the number of tasks in path 1.
Variability of each task is the same.
  e.g. Model 1 path 1 +/- 50% of 50 is 25 to 75; path 2 is 5 times +/- 50% of 10 which is 5 to 15 or a total of 25 to 75.

Figure 5-4 Varying the Number of Tasks in a Path
Table 5-14 Varying the Number of Tasks in a Path

<table>
<thead>
<tr>
<th>Model</th>
<th>Merge</th>
<th>Paths</th>
<th>Tasks</th>
<th>Shapeness</th>
<th>Number of Tasks Path 1</th>
<th>Simulation Ave Percent Optimism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>50/50</td>
<td>1</td>
<td>5.78</td>
</tr>
<tr>
<td>Model 2</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>50/50</td>
<td>2</td>
<td>4.49</td>
</tr>
<tr>
<td>Model 3</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>50/50</td>
<td>3</td>
<td>3.81</td>
</tr>
<tr>
<td>Model 4</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>50/50</td>
<td>4</td>
<td>3.45</td>
</tr>
<tr>
<td>Model 5</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td>50/50</td>
<td>5</td>
<td>3.35</td>
</tr>
</tbody>
</table>

*Note:* Each Model simulation run 3 times and averaged.

*Conclusions:* Optimism is the least when the number of tasks per path are the same. Optimism is the greatest when one path has only one task. Optimism is the greatest when the standard deviation is the greatest.
Table 5-15 Varying Task Durations Part 1 of 2

<table>
<thead>
<tr>
<th>Group</th>
<th>Ave Sim Optimism</th>
<th>Ave Sim Std Dev</th>
<th>Var 1</th>
<th>Var 2</th>
<th>Var 3</th>
<th>Var 4</th>
<th>Total Var</th>
<th>Std Dev</th>
<th>Sim Optimism</th>
<th>Sim Std Dev</th>
<th>Total Var</th>
<th>Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Path 1</td>
<td>5 5 5 5 5</td>
<td>1.82</td>
<td>0.53</td>
<td>0.17</td>
<td>0.17</td>
<td>0.17</td>
<td>0.69</td>
<td>0.83</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Path 2</td>
<td>2 5 6 7</td>
<td>1.95</td>
<td>0.56</td>
<td>0.03</td>
<td>0.17</td>
<td>0.25</td>
<td>0.69</td>
<td>0.89</td>
<td>1.07</td>
<td>1.06</td>
<td>1.14</td>
<td>1.07</td>
</tr>
<tr>
<td>Path 1</td>
<td>1 4 5 10</td>
<td>2.20</td>
<td>0.63</td>
<td>0.01</td>
<td>0.11</td>
<td>0.17</td>
<td>0.69</td>
<td>0.99</td>
<td>1.21</td>
<td>1.19</td>
<td>1.42</td>
<td>1.19</td>
</tr>
<tr>
<td>Path 2</td>
<td>1 4 5 10</td>
<td>2.65</td>
<td>0.76</td>
<td>0.01</td>
<td>0.03</td>
<td>0.06</td>
<td>1.36</td>
<td>1.46</td>
<td>1.21</td>
<td>1.46</td>
<td>2.10</td>
<td>1.45</td>
</tr>
<tr>
<td>Path 1</td>
<td>1 2 3 14</td>
<td>3.13</td>
<td>0.91</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>2.01</td>
<td>2.03</td>
<td>1.42</td>
<td>1.72</td>
<td>1.72</td>
<td>2.92</td>
</tr>
<tr>
<td>Path 2</td>
<td>1 2 3 14</td>
<td>3.57</td>
<td>1.03</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>2.01</td>
<td>2.03</td>
<td>1.42</td>
<td>1.72</td>
<td>1.72</td>
<td>2.92</td>
</tr>
</tbody>
</table>

![Normalized Std Dev vs. Normalized Standard Deviation](image)
Table 5-16 Varying Task Durations Part 2 of 2

<table>
<thead>
<tr>
<th></th>
<th>Ave Sim Optimism</th>
<th>Ave Sim Std Dev</th>
<th>Calculated</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Group 2</td>
<td></td>
<td>Var 1</td>
<td>Var 2</td>
</tr>
<tr>
<td>Path 1</td>
<td>5 5 5 5 5</td>
<td>1.83 0.53</td>
<td>0.17 0.17</td>
<td>0.17 0.17</td>
</tr>
<tr>
<td>Path 2</td>
<td>5 5 5 5 5</td>
<td>1.90 0.55</td>
<td>0.17 0.17</td>
<td>0.17 0.17</td>
</tr>
<tr>
<td></td>
<td>2 5 6 7</td>
<td></td>
<td>0.03 0.17</td>
<td>0.25 0.34</td>
</tr>
<tr>
<td></td>
<td>1 4 5 10</td>
<td></td>
<td>0.01 0.11</td>
<td>0.17 0.69</td>
</tr>
<tr>
<td>Path 1</td>
<td>5 5 5 5 5</td>
<td>2.05 0.59</td>
<td>0.17 0.17</td>
<td>0.17 0.17</td>
</tr>
<tr>
<td>Path 2</td>
<td>1 2 3 14</td>
<td>2.30 0.66</td>
<td>0.17 0.17</td>
<td>0.17 0.17</td>
</tr>
<tr>
<td></td>
<td>1 1 1 17</td>
<td>2.57 0.75</td>
<td>0.01 0.01</td>
<td>0.01 0.01</td>
</tr>
</tbody>
</table>

![Graph showing relationship between Normalized Optimism and Normalized Standard Deviation](image)
These tables show that once the data is normalized, the standard deviation of all the tasks in the paths under consideration is essentially equal to the optimism of the pair. Two sets of curves are shown in each of the two graphs. One is the simulation data and the other is the normalized standard deviation of the tasks under consideration. Not only is the relationship between the simulation and normalized standard deviation linear, the two are essentially equal. The results from this section and Section 5.2.6 are now combined using Equation 5-11 as explained in Chapter 4.

\[ P_i = D_{i1} + R_{i1} D_{i2} + \cdots + R_{ij} D_{ij} + \cdots + R_{ip} D_{ip} \] 5-11

Where \( P_i \) = Effective path length of all parallel paths p into merge point i

\( D_{ij} \) = Contribution of each parallel path because of its length

\( R_{ij} \) = Contribution of each parallel path because of its varying task durations.

This then is the effective number of parallel paths or an “effective P” to be used in the final model. The D’s account for the uneven task durations in a particular path and the R’s account for that particular path being shorter than the critical path. The derived number of P in Equation 5-11 is therefore an effective number of paths into the merge point section under consideration.

5.2.8 Scaling the Task Duration Distribution

Up to this point, all the Monte Carlo simulations were run with the restriction that the percent estimated for under the mode \(|U|\) plus the percent estimated for over the mode \(|O|\) was always equaled to 100 percent. The percentages are in absolute values.
Chapter 4 developed the methodology to eliminate this restriction by multiplying the model by a factor called $B$. This factor is restated here as Equation 5-12.

$$B = \frac{|\theta| + |\nu|}{100}$$  \hspace{1cm} \text{5-12}

To develop this factor a two parallel path project with 4 equal tasks in each path was selected. See Figure 5-5.

![Figure 5-5 Scaling Task Duration Distribution](image)

**Conditions:**  Each model is exactly the same except for task duration shapeness.  Here "U" means percent under or lower bound in task duration.  Here "O" means percent over or upper bound in task duration.  Here "O" and "U" are absolute values.  All previous simulations were made so that the sum of the absolute values of "O" and "U" always equaled to 100.

All parameters were kept the same except for the variability of the tasks. It was observed that the most optimistic condition occurs when the under mode percent was zero and the most pessimistic condition occurs when the under mode percent was 100%. These two conditions along with 50% were set as the lower bound and the over mode percent were varied. The results are plotted in Table 5-17.
The plots show clearly that there is a linear relationship between optimism and the sum of the under mode and over mode percent. Not surprisingly, the zero under mode curve passed through zero. After all, when both the under mode and over mode percentages equals zero there is no optimism.
In summary, this section eliminates the assumption that task duration variance of
the percent under the mode plus the percent over the mode always equaled 100 percent.

5.2.9 Final CD Scheduling Model (CDSM)

The findings in the previous steps went into the final CD Scheduling Model, which
will be referred to as the CDSM from here on. The CDSM is shown in Figure 4-2 and
Figure 4-3. Section 4.4.9 gives a recommendation on how to use the model. In the next
two sections (5.3 and 5.4) the model was applied to a variety of CD schedules and a set of
competed CD schedules. The results will be compared to Monte Carlo simulations and
where applicable against actual completion results.

5.3 Comparing the CDSM to Typical CD Schedules

The CDSM was tried on typical multi merge point CD project schedules using
Monte Carlo simulations to verify the results. The general overall plan in using the
CDSM is to run a critical path analysis on the overall CD schedule and then identify all
the merge points on the critical path and all the paths leading into those merge points.
The paths leading into merge points or merge point sections are then examined. Any
paths that are clearly much shorter (generally greater than 10% shorter) can be ignored.
At this point the rest of the schedule can also be ignored. A reduced CD schedule is then
produced and in our example project, that schedule is shown in Figure 5-6. In this
example the schedule assumes a task duration shapeness of 50% under the mode to 100%
over the mode. The CDSM from Figure 4-2 and Figure 4-3 was used on this network
with its results shown in Table 5-18 and Table 5-19. The CDSM shows that this schedule
has an optimism of 10.12 % while the Monte Carlo simulation shows an optimism of
9.94%. The CDSM results matches well with the Monte Carlo simulation but in either
In this case, the CDSM and Monte Carlo simulations is showing this project to complete in 289 days compared to the baseline of 263 days or 27 workdays or over 5 weeks late.

Two more projects were selected to mirror the full spectrum of CD projects to include the length of the project, the number of parallel paths of hardware and software development, the number of tasks and task duration shapeness. The range of projects attempted to capture the “normal” CD project from the very simple to the more complex. Each of these project schedules was tested with the following three sets of task duration shapeness: 25%/25%, 50%/100% and 82%/272% to represent estimates that are believed to be well understood to those having a high degree of uncertainty. The two additional project schedules are shown in Figure 5-7. The CDSM was applied to each of these schedules with the results compared to Monte Carlo simulations. The results are shown in Table 5-20.
Figure 5-6 Typical CD Schedule

Start-Up:
- RQ 1-1 → RQ 1-2 → RQ 1-3 → RQ 1-4 → RQ 1-5 → RQ 1-6
- RQ 3-1 → RQ 3-2 → RQ 3-3

Path 1: Critical Path 6 tasks (9/15/16/16/16/16) = 38 days
Path 1: 8 tasks (12/16/25/3/2/3/6/9) = 76 days
Path 3: 3 tasks (2/13/60) = 75 days

Merge Section 1

PD 1-1 → PD 1-2
PD 2-1 → PD 2-2 → PD 2-3
PD 3-1 → PD 3-2
PD 4-1
PD 5-1

Path 1: Critical path 2 tasks (40/45) = 85 days
Path 2: 3 tasks (15/40/15) = 70 days
Path 3: 2 tasks (15/40) = 55 days
Path 4: 1 tasks (50) = 50 days
Path 5: 1 tasks (75) = 75 days

Merge Section 2

CD 1-1 → CD 1-2
CD 2-1 → CD 2-2
CD 3-1 → CD 3-2 → CD 3-3
CD 4-1 → CD 4-2 → CD 4-3

Path 1: Critical path 2 tasks (45/45) = 90 days
Path 2: Critical path 2 tasks (45/45) = 90 days
Path 3: 3 task (18/12/35) = 65 days
Path 4: 3 tasks (18/10/45) = 73 days

Merge Section 3
Table 5-18 Typical CD Schedule Merge Section 1

Path 1: Critical Path 6 tasks (9/15/16/16/16/16) = 88 days
Path 1: 8 tasks (12/16/25/3/2/3/6/9) = 76 days
Path 3: 3 tasks (2/13/60) = 75 days

Upper bound = 100%
Lower bound = 50%
S = 0.67
B = 1.50

\[
S = \frac{|U|}{|U| + |L|}
\]

\[
B = \frac{|O| + |U|}{100}
\]

\[
R_{ij} = e^{-17.46 x_{ij} / B}
\]

<table>
<thead>
<tr>
<th>Percent Reduction</th>
<th>Std Dev</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_{i1}</td>
<td>9117.29</td>
<td>1.01512</td>
</tr>
<tr>
<td>D_{i2}</td>
<td>6.71751</td>
<td>1</td>
</tr>
<tr>
<td>R_{11}</td>
<td>0.14</td>
<td>0.20448</td>
</tr>
<tr>
<td>R_{12}</td>
<td>0.20448</td>
<td>8.52936</td>
</tr>
<tr>
<td>D_{i3}</td>
<td>10.82532</td>
<td>1</td>
</tr>
<tr>
<td>R_{13}</td>
<td>0.15</td>
<td>0.17915</td>
</tr>
<tr>
<td>D_{i3}</td>
<td>15.35619</td>
<td>1.41854</td>
</tr>
</tbody>
</table>

\[
\alpha_1 = \frac{12.00 (\bar{r} - 1)^{0.69}}{\mathbf{R}} + 20.24 (c_i - 10.23)^{0.69}
\]

Merge Section 1 (SDR)
### Table 5-19 Typical CD Schedule Merge Point Section 2 and 3

**Merge Section 2 (PDR)**

<table>
<thead>
<tr>
<th>Percent Reduction</th>
<th>Std Dev</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>15.02602</td>
<td>15.05190</td>
</tr>
<tr>
<td>1.18</td>
<td>10.19363</td>
<td>11.20315</td>
</tr>
<tr>
<td>35%</td>
<td>0.12821</td>
<td>11.73952</td>
</tr>
<tr>
<td>41%</td>
<td>0.01644</td>
<td>12.85184</td>
</tr>
<tr>
<td>50%</td>
<td>0.00829</td>
<td>13.50000</td>
</tr>
<tr>
<td>62%</td>
<td>0.25426</td>
<td>18.75000</td>
</tr>
</tbody>
</table>

- Sanction Optimum: 0.84%
- Model Optimum: 8.94%

**Merge Section 3 (Close Out)**

<table>
<thead>
<tr>
<th>Percent Reduction</th>
<th>Std Dev</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1.00000</td>
<td>15.91</td>
</tr>
<tr>
<td>0.00</td>
<td>1.00000</td>
<td>15.91</td>
</tr>
<tr>
<td>0.28</td>
<td>0.03943</td>
<td>10.537</td>
</tr>
<tr>
<td>0.19</td>
<td>0.11095</td>
<td>12.372</td>
</tr>
</tbody>
</table>

- \( P_{\text{rej}} = 2.1735 \)
- \( \alpha_{\text{p}} = 11.35 \)

Overall Optimum: \( \frac{\sum_{i=1}^{n} p_i \cdot \alpha_i}{\sum_{i=1}^{n} \alpha_i} \)
Figure 5-7 Typical CD Schedules

Model 1

Path 1: Critical Path 6 tasks (12/16/25/3) = 56 days
Path 2: 8 tasks (16.15, 25) = 56 days

Model 3

Path 1: Critical Path 2 tasks (12/16/25/3/6/9) = 76 days
Path 2: 6 tasks (9/15/11/10/8/16) = 69 days
Path 3: 6 tasks (9/15/11/3/5/16) = 61 days
Path 4: 1 task (5, 5) = 55 days

Path 1: Critical Path 2 tasks (15/40/40) = 65 days
Path 2: 3 tasks (25/40) = 65 days
Path 3: 1 tasks (70) = 70 days
Path 4: 3 tasks (35/40) = 28 days
Path 2: 3 tasks (3/5/15) = 23 days
Path 3: 3 tasks (5/5/12) = 22 days
Path 4: 3 tasks (3/5/10) = 16 days
Table 5-20 Comparing Models with Simulations and CDSM

<table>
<thead>
<tr>
<th>Model</th>
<th>Lower</th>
<th>Upper</th>
<th>Sim</th>
<th>Model</th>
<th>Delta</th>
<th>M</th>
<th>P</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25%</td>
<td>25%</td>
<td>2.71%</td>
<td>2.67%</td>
<td>-0.04%</td>
<td>2</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>1</td>
<td>50%</td>
<td>100%</td>
<td>14.43%</td>
<td>14.16%</td>
<td>-0.27%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>88%</td>
<td>272%</td>
<td>44.66%</td>
<td>42.38%</td>
<td>-2.28%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>25%</td>
<td>25%</td>
<td>0.94%</td>
<td>0.66%</td>
<td>-0.28%</td>
<td>3</td>
<td>12</td>
<td>36</td>
</tr>
<tr>
<td>2</td>
<td>50%</td>
<td>100%</td>
<td>9.94%</td>
<td>8.98%</td>
<td>-0.96%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>88%</td>
<td>272%</td>
<td>39.42%</td>
<td>36.06%</td>
<td>-3.36%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>25%</td>
<td>25%</td>
<td>0.40%</td>
<td>0.44%</td>
<td>0.04%</td>
<td>5</td>
<td>16</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>50%</td>
<td>100%</td>
<td>9.52%</td>
<td>9.89%</td>
<td>0.37%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>88%</td>
<td>272%</td>
<td>38.03%</td>
<td>39.08%</td>
<td>1.05%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.4 Comparing the CDSM with Actual CD Schedules

In this section four completed CD projects were examined. The CDSM was applied to the initial baseline schedule of one of these project schedules and compared to the actual results. Monte Carlo simulations also verified the results. But first we needed to know what kind of task duration we were going to assume. Examination of the CDSM finds that the task duration distribution is a critical factor in determining the resulting optimism. This is determined not only by the least likely and most likely estimates but also more importantly by what the shapeness is. The four actual CD projects were analyzed for the actual task duration distribution. First, all the tasks in the project were considered as to what was estimated and what the actual duration was. The task duration distribution of all the tasks in the project proved interesting but was determined to be of little concern for assessing the optimism of the project. A typical constraint type on most tasks and used in this research is to “start as soon as possible.” This, by default, puts
slack in all tasks not on the critical path. All tasks on four projects were analyzed and proved the distributions were greatly skewed to the right (lengthened). No investigation was made into this effect other than to say that “Parkinson’s Law” (Parkinson 1957) where “Work expands so as to fill the time available for its completion” might have been the major factor. The first step was then to find the tasks along the critical path at the time the project was first base lined and then compare those tasks with the completed project.

This study then zeroed in on the tasks that were on the critical path at the initial baseline and to see how they faired at project completion. Even with these tasks care had to be exercised in interpreting the results. As a CD project progresses it is not unusual to add forgotten tasks, delete tasks not needed and change task relationships and constraints. Even with this reduced set of tasks the four typical CD projects showed they were highly skewed to the right. The task duration data was curve fitted to a beta distribution using MATLAB. Recall that a beta distribution is only defined in our case for values greater than zero and less than 1. This means that the data needed to be transposed to a scale of greater than zero and less than 1. The result of one of the projects is plotted in Table 5-21.
Table 5-21 Project 3 Analyzed

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Beta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log likelihood</td>
<td>4.59205</td>
</tr>
<tr>
<td>Domain</td>
<td>$0 &lt; y &lt; 1$</td>
</tr>
<tr>
<td>Mean</td>
<td>0.309833</td>
</tr>
<tr>
<td>Variance</td>
<td>0.0199319</td>
</tr>
<tr>
<td>Parameter Estimate</td>
<td>Std. Estimated covariance of parameter estimates</td>
</tr>
<tr>
<td>a</td>
<td>3.0142</td>
</tr>
<tr>
<td>b</td>
<td>6.7142</td>
</tr>
<tr>
<td>a</td>
<td>3.7098</td>
</tr>
<tr>
<td>b</td>
<td>7.5997</td>
</tr>
<tr>
<td>Number of critical tasks</td>
<td>10</td>
</tr>
<tr>
<td>Number tasks on time</td>
<td>1</td>
</tr>
<tr>
<td>Baseline duration</td>
<td>167 days</td>
</tr>
<tr>
<td>Actual duration</td>
<td>175 days</td>
</tr>
</tbody>
</table>

![Graph: Project 3 vs Beta Distribution]
Table 5-22 Completed CD Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>Mean</th>
<th>Mode</th>
<th>Total Number of Tasks</th>
<th>Number of Critical Tasks on Time</th>
<th>Number of Tasks</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.08</td>
<td>11.87</td>
<td>0.21</td>
<td>0.16</td>
<td>691</td>
<td>12</td>
<td>4</td>
<td>0.17</td>
<td>5.00</td>
</tr>
<tr>
<td>2</td>
<td>2.05</td>
<td>4.69</td>
<td>0.30</td>
<td>0.22</td>
<td>505</td>
<td>63</td>
<td>37</td>
<td>0.20</td>
<td>2.61</td>
</tr>
<tr>
<td>3</td>
<td>3.01</td>
<td>6.71</td>
<td>0.31</td>
<td>0.26</td>
<td>214</td>
<td>10</td>
<td>1</td>
<td>0.10</td>
<td>4.89</td>
</tr>
<tr>
<td>4</td>
<td>1.27</td>
<td>10.16</td>
<td>0.11</td>
<td>0.03</td>
<td>1392</td>
<td>91</td>
<td>37</td>
<td>0.03</td>
<td>2.36</td>
</tr>
</tbody>
</table>

Average: 2.35 8.36 0.23 0.17 701 44 20 0.12 3.72

Where: The letters $\alpha$ and $\beta$ are parameters of the Beta Distribution.
Lower and Upper Bounds are normalized to the mode i.e. an Upper Bound of 5.00 means the Upper bound is 5 times the mode.

The results of all four projects are shown in Table 5-22.

The numbers for “a” and “ß” are the key beta distribution parameters. See the discussion in Section 4.2.4.2. There it was stated that Risk + sets either a or ß equal to six. This is intended to make sure the resulting distribution represents a real life task duration distribution. The data in Table 5-22 generally supports this rule. In our case the letter “ß” averaged to be 8.36 meaning the standard deviation would be slightly less than what Risk + would have assigned.

Table 5-22 also includes the normalized lower bound and upper bound of each of the four projects. They were averaged for a lower bound of 0.12 and an upper bound of 3.72. This is interpreted as the task duration distribution having a mode of 1.0 and a lower bound of 88% under the mode and an upper bound of 272% over the mode. This was used in the model to analyze project 3.

The CDSM in Figure 4-2 and Figure 4-3 was applied to project 3 as being representative of a CD project. The analysis went through the following steps:
1. Analyze the initial project schedule for its critical path. Make sure to take into account any non-finish to start constraints. Identify all merge points along the critical path and all parallel paths leading into these merge points.

2. Construct a reduced project schedule with all tasks along the critical path and all parallel paths leading into merge points on the critical path.

3. Follow the steps in the proposed CDSM shown in Figure 4-2 and Figure 4-3. The results are shown in Table 5.23 and compared to the actual completion date. A Monte Carlo simulation was also run on both the reduced schedule and those results are also shown. All the results are close to the actual finish adding credibility to the model. It needs to be noted that the model in real life is highly dependent on the task duration distribution. The four projects were accomplished at the same engineering site and engaged in roughly the same type of development. Changing the site or type of development may very well change the task duration distribution.
Table 5-23 Predicting the Results with the CDSM on an Actual Project

| Merge 1 Path 1 | 45 | 1 | 1 | 1 | 18.2 | 45 |
| Merge 2 Path 1 | 20 | 1 | D21 | 12.00 | 1 |
| Path 2 | 15 | R2 | 0.25 | 0.30 | 1 | D22 | 9.00 | 1 | 1.2975 | 29.9 | 20 |
| Merge 3 Path 1 | 10 | 1 | 1.00 | 1 | 1 | 18.2 | 10 |
| Merge 4 Path 1 | 25 | 1 | D41 | 15.00 | 1 |
| Path 2 | 24 | R4 | 0.04 | 0.82 | 1 | D42 | 14.40 | 1 | 1.8237 | 44.4 | 25 |
| Merge 5 Path 1 | 10 | 1 | 1.00 | 1 | 1 | 18.2 | 10 |
| Merge 6 Path 1 | 5 | 1 | 1.00 | 1 | 1 | 18.2 | 5 |
| Merge 7 Path 1 | 15 | 1 | D71 | 9.00 | 1 |
| Path 2 | 10 | R4 | 0.33 | 0.20 | 1 | D72 | 6.00 | 1 | 1.1986 | 26.7 | 15 |
| Merge 8 Path 1 | 15 | 1 | 1.00 | 1 | 1 | 18.2 | 15 |
| Merge 9 Path 1 | 10 | 1 | 1.00 | 1 | 1 | 18.2 | 10 |
| Merge 1 Path 1 | 20 | 1 | 1.00 | 1 | 1 | 18.2 | 20 |

Upper bound = 272%  \[ S = \frac{\text{Upper bound}}{\text{Lower bound}} \]
Lower bound = 88%  \[ S = \frac{\text{Lower bound}}{100} \]
Upper bound + Lower bound = 360%  \[ S = \frac{\text{Upper bound} + \text{Lower bound}}{100} \]
B = 3.60

<table>
<thead>
<tr>
<th>Baseline duration: 175</th>
<th>Days</th>
<th>Optimism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual duration: 250</td>
<td>31.4%</td>
<td></td>
</tr>
<tr>
<td>Monte Carlo Simulation on CDSM: 221</td>
<td>26.4%</td>
<td></td>
</tr>
<tr>
<td>CDSM: 217</td>
<td>24.0%</td>
<td></td>
</tr>
</tbody>
</table>
5.5 Model Development and Simulation Experiments Summary

A 3-phase approach was used for the study. In the Phase 1, the initial model was developed starting with a full factorial 4-factor 3-treatment design. Three Monte Carlo simulations were run on each of the 81 cells in the design. The resulting ANOVA table identified the significant factors and their interactions. With this a linear model was produced. From this start many of the simplifying assumptions were eliminated and incorporated into the final model. Monte Carlo simulations and non-linear regression techniques were used to find good fit relationships that could be entered into the model. In Phase 2 the model was tried against typical CD schedules and verified against Monte Carlo simulations.
Chapter 6

Findings and Recommendations

6.1 Findings

The CDSM proposed in Figure 4-2 and Figure 4-3 produces good results when compared to Monte Carlo simulations of typical CD schedules. The model also compares well with completed actual CD project schedules. The research did reinforce the belief that CD schedules are dynamic as many project schedules are and they become adjusted as the project progresses to resolve mistakes in the initial schedule and more importantly to find work arounds as the project gets into trouble. This research focused in on the initial schedule.

The key benefit of this research is that the proposed model gives insight into what is driving optimism in a CD schedule. The model gives specific relationships on how the various factors interact. From this a series of findings are listed below that a decision maker can use to assess the optimism of the baseline schedule. The findings will tell what structural aspects of the project schedule are driving optimism and by how much. In addition, suggestions are given on what might be done to minimize the impact and give a magnitude of the improvement if made. The goal is the resulting baseline schedule will have less risk and have improved chances of completing on time.

Here are the findings:
6.1.1 Reduce Task Duration Estimation Uncertainty

Increasing the uncertainty in task duration estimation linearly increases optimism. This is with the factor B in the CDSM. Reduce task estimation uncertainty and you reduce schedule optimism. The B factor is also used in determining the impact of parallel path lengths less than the critical path. This follows a reciprocal e function.

6.1.2 Reduce Task Duration Shapeness

Shapeness of the task duration estimate has a dramatic impact on the schedule optimism. The CD Scheduling Model has shapeness factor S multiplied by 20.3. Reduce the task duration shapeness and you reduce schedule optimism. The shapeness number is probably being estimated from previous similar tasks and will probably be repeated if program action is not taken to improve this parameter.

6.1.3 Analyze Longest Merge Point Section First

The impact of multiple merge points is accounted for by a weighted average as shown in Equation 4-17. That is, sections with longer critical paths have a greater impact on the overall optimism. These are the sections to be analyzed first for improvement.

6.1.4 Break Large Tasks into Small Tasks

More tasks in a section reduce optimism by the reciprocal of the square root of the number of tasks in that section. Increasing the number of tasks in a section by breaking tasks into smaller tasks is one way to reduce the optimism.

6.1.5 Reduce Number of Parallel Paths

Each parallel path into a merge point increases optimism by a factor of 12.00 assuming the path is the same length as the critical path. Even when the parallel path
length is within 10 percent of the critical path and offset by determining an effective P, optimism is still added. Reducing the number of parallel paths within 10 percent of the critical path length will of course help. If they can’t be eliminated, at least they should be examined closely to see what could be done to shorten them to more than 10 percent of the critical path length.

6.1.6 Size Task Durations in a Path to be About the Same Duration

The more uneven the task durations are in a particular parallel path, the more the optimism. The research showed that when all the tasks in a particular path are of equal length, the optimism was the least. For example at the other extreme, the path with two tasks with one task at a duration of one unit and the other task at the length of the path minus that one unit, gives the maximum amount of optimism. Converting the task duration uncertainty to its standard deviation and then normalizing the results finds that this number is directly equal to the normalized optimism increase. One simple way to reduce optimism is to break long duration tasks in a path into smaller duration tasks closer to the overall average task duration in that path.

6.1.7 Ignore Parallel Paths Less Than 10 Percent of Critical Path

Path duration lengths into merge points that are more than 10 percent less the critical path have little or no impact on optimism.

The following is a list of other features investigated for their contribution to the optimism but was not incorporated into the final CD Scheduling Model.
6.1.8 Standard Deviation Determines Optimism

The magnitude of the standard deviation determines the optimism and not the selected distribution type. The common types often used in software applications are beta, triangle, normal and uniform. The beta and triangle distributions have the advantage that they can show shapeness. As normally identified with the limits of the distribution, the beta distribution has a smaller standard distribution than the triangle and also the normal distribution. However, when the same standard deviation is given for distribution the impact on optimism is the same.

6.1.9 Optimism is Relative to Other Task Durations

The duration of path is only important in comparison with other tasks. That is, if all task durations in one schedule were 5 days and on another schedule all task durations were 10 days, the impact on optimism would be the same in both cases.

6.2 Recommendations for Future Research

This research focused on a suspicion that virtually all CD projects have overly optimistic schedules simply because of the deterministic CPM, PDM and similar tools that were being used to develop the initial baseline schedules. Since the CPM has been around since 1958 and has been used extensively in many industries, many believe that these tools can be trusted. This research has shown that these tools have their shortcomings and these shortcomings need to be considered in producing schedules that are to be believed. The research concentrated on the CD project where software and hardware are being developed concurrently and where errors can be particularly severe. The findings should be of benefit to other projects engaged in parallel efforts. This
research was focused on proposing a model and developing a set of findings to help the
decision maker in analyzing the initial baseline schedule. The goal was to give a more
realistic outlook of the baseline schedule and give suggestions on how to reduce schedule
risk and improve the chances of completing on time.

Several areas are ripe for further study. First, one limitation of the model is all
tasks in the schedule have the same task duration distribution. In CD projects this is not
seen as a severe limitation since historical data on similar task duration is often sketchy
and at best a general sense of how good the estimate is. However, a good refinement to
model would be to consider different task distribution for different parts of the schedule.

Another area recommended for further study is to conduct analysis on a wider
selection of projects. In the current study a beta distribution was selected as fairly
accurately reflecting the task duration distribution however, a better representative
distribution may be found.

A third area recommended for study is what does the task duration estimators
believe the duration distribution is? Also, what instructions are estimators given in doing
the estimate and how does that impact the actual estimate they produce? A questionnaire
may be a good idea to gain this insight. Full understanding of what the estimator believes
they are providing as inputs to the schedule generation is critical to the veracity of the
final schedule. There are two specific aspects that are believed to be of paramount
consideration in developing a schedule. First, determine what are the assumptions a
typical estimator uses in estimating the length of time to complete a task. This is a key
factor in any schedule development and an area to be explored in this research.
Possibilities include the most likely time, the most optimistic time, and the most
pessimistic time to complete and at what point are the estimates made i.e. 0%/100%, 5%/95% or something else. Next, does any assumption change over time? What kind of distribution function does the estimator believe the task will follow? For example, is the distribution a normal distribution or triangular or possibly a beta function or something else? What an estimator believes the task duration distribution might be and what it actually is may well be different. The data collected from the questionnaires along with actual results of completed projects could be used in refinement of the model presented in this research.
References


Gutierrez, G. J., M. D. C. McCombs School of Business, The University of Texas at Austin, Austin, TX 78712, USA; e-mail: genaro.gutierrez@bus.utexas.edu , et al. (2001). “Robustness to Variability in Project Networks.” IIE Transactions 33(8): 649-660.


Appendices
Appendix A

Notations

a Most optimistic activity duration
b Most pessimistic activity duration
d Scaling factor in beta distribution
i Number of merge points
j Number of paths per merge point
e,f,g,h Parameters for Ramberg-Schmeiser distribution
K Reduction factor in SMCS method
M Number of merge points
m Expected activity/path duration
P Number of Paths
P1,P2 Intermediate probabilities used in NRB method
PL Lower probability bound in NRB method
PU Upper probability bound in NRB method
P(F_i) Probability of ith path failure in the NRB method
P(T) Probability that a project completes in time T
R_{ij} Correlation factor used in PNET and NRB methods
R(p) Ramberg-Schmeiser distribution
s Activity/path standard deviation
S Shapeness of the task duration
T Total number of tasks
U,V End points of beta distribution
Appendix A (Continued)

\( x \) Normalized value for a standard normal distribution

\( x_{ij} \) Equals 1 if the ith merge point has j paths, otherwise equals 0

\( \Phi \) Standard normal distribution function
About the Author

The author has a Bachelor of Science degree in Electrical Engineering from the University of Wisconsin, a Master of Science degree in Electrical Engineering from the University of New Mexico and a Master of Science degree in Industrial Engineering from the Kansas State University. He was registered as a Professional Engineer in the State of New Mexico. The Project Management Institute (PMI) has certified him as a Project Management Professional (PMP). He has worked 15 years in US government and 13 years in industry project management offices. He has served as the Project Manager on 5 projects.