

**A COMPREHENSIVE STUDY OF
PROJECT SCHEDULING TECHNIQUES
AND THE INTRODUCTION OF NEW TECHNIQUES**

thesis

**SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF**

DOCTOR OF PHILOSOPHY

IN CIVIL ENGINEERING

OF THE



UNIVERSITY OF CALICUT

BY

P.R. SREEMAHDEVAN PILLAI

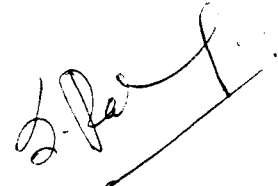


**DEPARTMENT OF CIVIL ENGINEERING
REGIONAL ENGINEERING COLLEGE
CALICUT - 673 601 KERALA**

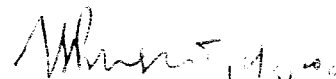
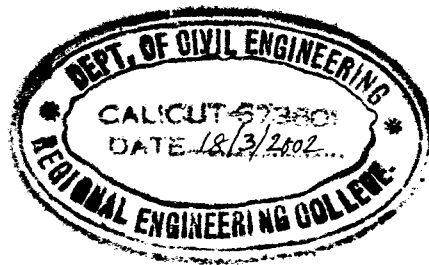
JANUARY 2002

CERTIFICATE

This is to certify that this thesis entitled **A COMPREHENSIVE STUDY OF PROJECT SCHEDULING TECHNIQUES AND THE INTRODUCTION OF NEW TECHNIQUES** is a bona fide record of original research work done by **Mr. P.R. Sreemahadevan Pillai** under my supervision and guidance. The thesis is submitted to the University of Calicut for the award of the degree of **Doctor of Philosophy in civil engineering**. The matter contained in this thesis has not been submitted elsewhere for any degree.



Dr. S. Palaniraj
Supervisor and Guide



Professor and Head
Department of Civil Engineering,
R.E.C., Calicut

R.E.C., CALICUT
JANUARY 2002

DECLARATION

I hereby solemnly declare that the work presented in this thesis entitled **A COMPREHENSIVE STUDY OF PROJECT SCHEDULING TECHNIQUES AND THE INTRODUCTION OF NEW TECHNIQUES** submitted to the University of Calicut for the award of the degree of **Doctor of Philosophy** is a bona fide record of fully independent and original research work carried out by me at the Department of Civil Engineering, Regional Engineering College, Calicut – 673 601 under the guidance of **Dr. S. Palaniraj**, during the period 1996 - 2001.

The matter contained in this thesis has not been submitted anywhere else for the award of any degree or diploma.



P.R. Sreemahadevan Pillai
Assistant Professor of Civil Engg.,
N. S. S. College of Engineering,
Palakkad – 678 008, Kerala

R.E.C., CALICUT
JANUARY 2002

ACKNOWLEDGEMENT

The author wishes to express his heartfelt sense of gratitude and sincere thanks to **Dr. S. Palaniraj**, Professor of Civil Engineering, Regional Engineering College, Calicut for his valuable guidance and encouragement throughout the course of this research work. It was a privilege to work under his able supervision and inspiration. The keen interest shown by him to give timely advice and support at all stages of the work is beyond words of gratitude.

The author wishes to express his thanks to **Dr. M. P. Chandrasekharan**, Principal and **Dr. S. Unnikrishna Pillai**, former Principal, Regional Engineering College, Calicut, for the help rendered in times of need.

The author is thankful to **Dr. V. S. Prasad**, Head and **Dr. Balagopal. T. S. Prabhu** and **Dr. K. Vincent Paul**, former Heads, of the Department of Civil Engineering, R.E.C., Calicut for the facilities provided by them to carry out this research work

The author is indebted to **Prof. T. Divakaran**, Head of the Department of Civil Engineering, N.S.S. College of Engineering, Palakkad for his help and advice in completing the research work.

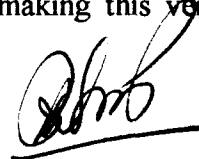
The author is thankful to **Dr. V. P. Mohandas**, Principal and **Dr. T. V. Baburajendran**, former Principal, N.S.S. College of Engineering, Palakkad, for their cooperation and help.

The author wishes to put on record, the gratitude to the staff of Regional Engineering College, Calicut for all the help rendered in the course of this research work.

Thanks a lot to the colleagues at the N.S.S. College of Engineering, Palakkad for their constant inspiration and help.

Gratefully acknowledges the constant support of his wife and colleague **Shmt. A.V. Hema Nalini** and children **Ms. Siddhi. S** and **Mr. Siddharth. S**.

Finally, the author thanks all others who helped him in making this venture a success.



P.R. Sreemahadevan Pillai

R.E.C., CALICUT
JANUARY 2002

ABSTRACT

Project is a one-time activity and hence the rectification of the flaws in a project such as time and cost overruns is not possible since the experience of the same project is not available at hand. The best approach for controlling the time and cost overruns is to make perfect planning of the entire project. Accurate planning and effective scheduling are critical to the performance of any organization. Even the smallest project with limited resources has a huge number of combinations of resources and courses of action. The importance of a carefully considered plan and robust schedule cannot be overstated. In the last four decades, many scheduling techniques have come up, with varying fields of application. The evolution of the network planning methods and the simulation techniques has revolutionized the project planning methods. All the existing methods of project planning and scheduling stress up on individual activities in the project and have come out with methods for monitoring these individual activities. The application of these techniques include areas like space research, logistics, electronic circuit design, engineering construction, production shop, queuing, computer networking etc., since proper planning and scheduling is necessary in all these fields.

The existing methods of project scheduling include bar (Gantt) charts , critical path method (CPM), decision critical path method (DCPM), program evaluation and review technique (PERT), Monte Carlo simulation (MCS) technique, line-of-balance (LOB) method, Petri net (PNET), graphical evaluation and review technique (GERT) and venture evaluation and review technique (VERT). The theoretical background of all these

methods is analyzed in detail in this thesis. The advantages and disadvantages of these methods are compared with specific examples under different situations. In addition to the above common methods, the possibility of application of the methods used in other fields like fuzzy set theory, fuzzy expert system, artificial neural network (ANN) and genetic algorithm (GA) applications in project scheduling is explored in this thesis. Though the theoretical formulations and applications suggested in these methods are not that comprehensive, it will help in further research in the field.

Computer software applications in project scheduling are as old as the computer itself. Two most popular packages available at present are the *Primavera* and *Microsoft project*. These two packages have been studied thoroughly and their applications with a critical analysis and comparison are presented in this thesis. The uses of these packages are also demonstrated with examples.

A new method by name *time step chart* is introduced here. This charting technique has the advantage that all possible information about activities are available from the chart itself and correlation of activities are shown without complicated networks. Probabilistic and deterministic activities are easily identified in the chart and highly compact nature of chart makes handling easier. This chart incorporates all the advantages of Gantt, CPM/PERT and GERT charts. The PERT mathematics can very well be applied to this chart also and critical activities are immediately identified from the chart itself.

CONTENTS

<u>Description</u>	<u>Page No.</u>
TITLE PAGE	i
CERTIFICATE	ii
DECLARATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
CONTENTS	vii
LIST OF FIGURES	xi
LIST OF TABLES	xiii
PRINCIPAL NOMENCLATURE ABBREVIATIONS	xiv
CHAPTER 1. INTRODUCTION	1
1.1 Purpose of the thesis	2
1.2 State of the art	3
CHAPTER 2. BAR CHARTS	8
2.1 Introduction to bar charts	9
2.2 Critical review of the Gantt chart	9
2.2.1 Advantages of the Gantt chart	9
2.2.2 Disadvantages of the Gantt chart	9
CHAPTER 3. CRITICAL PATH METHOD (CPM)	10
3.1 Introduction to critical path method	11
3.2 Assumptions of the CPM model	14
3.3 Perceived disadvantages of CPM scheduling	14
3.4 Factors affecting successful CPM scheduling	15
3.5 Can linear projects be scheduled using CPM?	17
3.6 Is CPM scheduling capable of fulfilling all required functions?	19
3.6.1 Offsetting uncertainty and change	19
3.6.2 Focusing attention on objectives	19
3.6.3 Gaining economical operation:	20
3.6.4 Facilitating control	21
3.6.5 Allocating contractual responsibilities	22
3.6.6 Coordinating contributions from various groups during engineering phase	22
3.6.7 Resolving delay and change order disputes on a predefined, quantifiable, and equitable basis	22
CHAPTER 4. DECISION CPM	25
4.1 Introduction	26
4.2 The mathematical basis of decision CPM	27
4.3 Decision project graphs	31
4.4 Decision graph solution by integer programming	33
4.5 Decision graph solution by heuristic techniques	39
4.6 Application of decision CPM to dynamic monitoring and control of projects	45

CHAPTER 5. PROGRAM EVALUATION AND REVIEW TECHNIQUE (PERT)	48
5.1 Basic PERT theory	49
5.2 Probabilistic PERT	51
5.3 Criticality indices for the activities in PERT networks	52
CHAPTER 6. MONTE CARLO SIMULATION TECHNIQUE	55
6.1 Introduction	56
6.2 Monte Carlo approach	59
6.3 Problems with Monte Carlo approach	60
6.4 Statistical distribution of cost items	60
6.5 Correlation between project cost components	62
6.6 Approximate method for incorporating correlation	62
6.7 Exact methods for incorporating correlations in simulation	63
6.8 Multivariate normal distribution	64
6.9 Multivariable lognormal distribution	65
6.10 Simplified Monte Carlo Simulation (SMCS) method	67
CHAPTER 7. LINE OF BALANCE (LOB) SCHEDULING TECHNIQUE	69
CHAPTER 8. PETRI NETS IN SCHEDULING	74
8.1 Introduction	75
8.2 Petri net approach	75
8.3 Mathematical definition	77
8.4 Properties of Petri nets	78
8.5 Petri net constructs useful in project modeling	79
8.6 Time Petri nets (TPN)	82
8.7 Extended transitions.	83
8.8 Coloured Petri Nets (CPNs)	84
8.9 Queuing disciplines for places	86
CHAPTER 9. GRAPHICAL EVALUATION AND REVIEW TECHNIQUE (GERT)	87
9.1 Introduction	88
9.2 Analysis of research and development expenditures	93
9.3 Queuing analysis using GERT	98
CHAPTER 10. VENTURE EVALUATION AND REVIEW TECHNIQUE (VERT)	101
10.1 Introduction	102
10.2 An example network	103
10.3 General comment on VERT	108
CHAPTER 11. FUZZY SETS THEORY	112
11.1 Introduction	113
11.2 Fuzzy sets theory	114
11.3 Fuzzy number	115
11.3.1 Operation of TFNs	116
11.4 Time representation	117
11.4.1 Crisp time and Interval	118
11.4.2 Fuzzy Time	119
11.5 Compatibility between fuzzy events	125
11.6 Criticality measurement	128

CHAPTER 12. FUZZY EXPERT SYSTEM FOR NETWORK RESOURCE ALLOCATION	131
12.1 Introduction	132
12.2 Fuzzy set application in measuring external impact	133
12.2.1 Basic concept of resource allocation	133
12.3 Relative weights of criteria	136
12.4 Susceptibility of activity	140
CHAPTER 13. ARTIFICIAL NEURAL NETWORK	143
13.1 History of artificial neural networks	144
13.2 Basic concepts	145
13.3 Back-propagation algorithm	150
13.4 Cost optimization model for a construction project using ANNs	151
13.5 Application of ANN for prediction of construction labour productivity	155
13.5.1 Network structure with focus on output layer	157
CHAPTER 14. GENETIC ALGORITHM (GA) APPLICATIONS IN SCHEDULING	160
14.1 Basic concepts	161
14.2 Scheduling with Gas	164
14.3 The GA-scheduler	166
14.3.1 Structure of GA-scheduler	166
14.3.2 Schedule builder	166
14.3.3 Chromosome representation	167
14.3.4 Interpretation of the chromosome	170
14.4 Resource scheduling with genetic algorithms	171
CHAPTER 15. COMPUTER SOFTWARE IN PROJECT SCHEDULING	177
15.1 Introduction	178
15.2 Primavera project planner	178
15.2.1 Features available in the P3 2.0 b release	178
15.2.2 Using the Primavera project planner	183
15.2.3 A planning example with Primavera	193
15.3 Microsoft Project	203
15.3.1 Facilities and features available with Microsoft project	204
15.3.2 The Gantt chart view in Microsoft Project	207
15.3.3 The PERT chart view	208
15.3.4 The Resource Usage view	210
15.3.5 Other views in Microsoft Project	211
15.3.6 Examples of using Microsoft Project	212
15.4 A comparison of Primavera and Microsoft project packages	220
CHAPTER 16. TIME-STEP CHART	221
16.1 Introduction	222
16.2 Advantages of the <i>Time-step chart</i>	230
16.3 How to draw the <i>Time-step chart</i> ?	231
16.4 Determination of critical path in the TSC	233
16.5 Comparison of Gantt chart with <i>TSC</i>	235
16.6 Comparison of the CPM/PERT network with <i>TSC</i>	236
16.7 An example for comparing the CPM/PERT network with <i>TSC</i>	237

CHAPTER 17. CONCLUSION AND SUGGESTIONS FOR FURTHER WORK	242
17.1 Summary of the thesis	243
17.2 Suggestions for further work	245
REFERENCE	247
PUBLICATIONS BY THE AUTHOR	266

LIST OF FIGURES

<u>Figure No.</u>	<u>Description</u>	<u>Page No.</u>
Fig 3.1	AON diagram for house construction project in Table 3.1	12
Fig 3.2	AOA diagram for house construction project in Table 3.1	13
Fig.4.1	Decision project graph No. 1	34
Fig.4.2	Project graph 1 – decisions S_{21} , S_{51}	36
Fig 4.3	Project graph 1- decisions S_{21} , S_{52}	37
Fig 4.4	Project graph 1 – decision S_{22} , S_{51}	38
Fig 4.5	Decision project graph No.2	41
Fig 4.6	Decision project graph No. 2 - optimum solution	42
Fig.4.7	State of project graph at the beginning of day 31 in Fig 4.5	43
Fig.4.8	Effect of incremental job cost on completion cost	44
Fig. 6.1 (a)	Symmetric distribution	57
Fig. 6.1 (b)	Asymmetric distribution	58
Fig.7.1.	Line-of-Balance diagram for activity	73
Fig 8.1	PN model of crane hoisting materials	76
Fig 8.2	Typical PN constructs useful in construction project scheduling	80
Fig 8.3	Example of use of extended transitions	83
Fig 9.1	A simple network with an AND node	90
Fig 9.2	A simple network with an inclusive-or node	90
Fig. 9.3	Calculation of the equivalent $w -$ function	92
Fig 9.4.	GERT network of a research and development process	93
Fig 9.5	Decision box network	95
Fig 9.6	Detailed segment of the network	95
Fig 9.7	A two-port description of an activity	96
Fig 9.8	GERT representation of machine – spare – repairman problem	98
Fig 10.1	The VERT network	105
Fig 11.1	Fuzzy numbers	115
Fig 11.2	Crisp times	118
Fig 11.3	Fuzzy times	118
Fig 11.4	Demonstration of possibility measure and agreement index	126
Fig 12.1	Basic concept of resource allocation	134
Fig 13.1	Typical model of a neuron	146
Fig 13.2	Certain transfer functions	147
Fig.13.3	Typical feed forward network	148
Fig 13.4	Recursive neural network	149
Fig 13.5	Neural network topology for neural dynamics cost optimization model	152
Fig 13.6	Goals of output structure	159
Fig 14.1	Chromosomal representation	162
Fig 14.2	Crossover mechanism	165
Fig 14.3	GA – Scheduler builder – Evaluation function interface	168
Fig 15.1(a)	Bar chart of the highway construction (widening) project (Page 1)	195
Fig 15.1(b)	Bar chart of the highway construction (widening) project (Page 2)	196
Fig 15.1(c)	Bar chart of the highway construction (widening) project (Page 3)	197
Fig 15.2	Costs graph of the highway construction (widening) project	198
Fig 15.3	Projected cash flow of the highway project (Primavera)	199
Fig 15.4	Pure logic diagram sample from Primavera	200
Fig 15.5	Time scaled logic diagram (Primavera)	201
Fig 15.6	Sample PERT chart for the highway project (Primavera)	202
Fig 15.7	Resource graph output from Microsoft Project package	212
Fig 15.8	Gantt chart for the highway construction (widening project)	214
Fig 15.9	Gantt chart from Microsoft Project	217
Fig 15.10	PERT chart output from Microsoft Project (rectangular connectivity)	218
Fig 16.1	Legends of the Time Step Chart	228

Fig 16.2	Time Step Chart for the small house construction project	229
Fig.16.3	Activity-on-arrow network for the construction of a brick compound wall with top fencing for a security area.	239
Fig.16.4	TSC for the construction of a brick compound wall with top fencing for a security area.	240

LIST OF TABLES

<u>Table No.</u>	<u>Description</u>	<u>Page No.</u>
Table 3.1	House Construction Project	11
Table 4.1	Decision set – total cost of project with given decision sets and due date	33
Table 9.1	Node characteristic and simple	89
Table 9.2	Possible Events for inclusive- or Nodes	91
Table 9.3	Definitions for research and development Network	94
Table 9.4	GERT Analysis of a Queuing problem	99
Table 10.1	Critical path (and PERT) data	104
Table 10.2	Cost data on each task	106
Table 10.3	Times and costs at terminal node under differing model assumptions	107
Table 10.4	Detail of the failure instances	107
Table 12.1	F_i and G_i recommendations in fuzzy sets	136
Table 12.2	Probability of rain in specific period	137
Table 12.3.	Susceptibility of an activity	139
Table 12.4	Degree of impact of an activity	141
Table 13.1	Productivity values for different zones	158
Table 15.1	Activities list of a highway construction (widening) project	192
Table 15.2	Activity particulars of the high way construction project	193
Table 15.3	Data for Fig 15.7	213
Table 15.4	List of activities of a small house construction project	216
Table 16.1	List of activities of a brick compound wall construction project	238

PRINCIPAL NOMENCLATURE ABBREVIATIONS

AI	Agreement index
AN	Activity network
ANN	Artificial neural network
AOA	Activity-on-arrow
AON	Activity-on-node
BP	Back propagation
c, C	Cost
CA	Criticality index of activity
CDF	Cumulative density function
CF	Current float
CFT	Current float time
C_i	Cost of activity I
CL	Critical list
CLT	Central limit theorem
COV	Coefficient of variation
CP	Criticality index of path
CPM	Critical path method
CPN	Coloured Petri net
CT	Current time
C_{tot}	Total project cost
DAN	Deterministic activity network
DCPM	Decision critical path method
df	Distribution function
d_i	Duration of activity I
E(t)	Expected duration
EF	Early finish
ES	Early start
F(t)	Function of time
FD	Fuzzy duration
FEF	Fuzzy early finish
FES	Fuzzy early start
FLF	Fuzzy late finish
FLS	Fuzzy late start
FT	Finish time
G	Graph
GA	Genetic algorithm
GERT	Graphical evaluation and review technique
J	Job set
LF	Late finish
LFT	Late finish time
LOB	Line of balance
LS	Late start

LVQ	Learning vector quantization
MCS	Monte Carlo Simulation
N	Node number
P	Probability
P3	Primavera 3 release 2 of 1997
PAN	Probabilistic activity network
PDF	Probability density function
PERT	Program evaluation and review technique
PLF	Preliminary late finish
PM	Possibility measure
PN	Petri net
Ra_i	Resource available on day I
S	Standard deviation of the project
s_{ij}	Susceptibility
SMCS	Simplified Monte Carlo simulation
ST	Start time
t	Time
t_e	Expected duration
TF	Total float
TFN	Trapezoidal fuzzy number
t_m	Most likely duration
t_o	Optimistic duration
t_p	Pessimistic duration
T_{proj}	Fuzzy project duration
VERT	Venture evaluation and review technique
VPM	Vertical production method
w_k	Weightage factor for k^{th} resource
X^T	Transpose of X
ρ	Mean rate
N	Standard deviation of activity
μ	Mean

INTRODUCTION

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 1

INTRODUCTION

1.1 PURPOSE OF THE THESIS

This thesis presents a comprehensive study of the project scheduling techniques, brought out all over the world, in the past half a century, since the emergence of such techniques and also introduces a new technique. The purposes of the thesis are listed below.

- (1). To bring out all available project scheduling techniques with its theoretical background, application etc.
- (2). To critically review all available project scheduling techniques, pointing out its advantages, disadvantages and suggesting modifications, if any.
- (3). To critically analyze two computer software available for project scheduling techniques with an insight into their scope and limitation.
- (4). To demonstrate the project scheduling techniques, by applying them in typical situations with examples.
- (5). To suggest new areas of application of existing techniques in project scheduling.
- (6). To present a new project scheduling technique, and bring out its application.
- (7). To present a condensed information on project scheduling techniques.
- (8). To utilize these ideas for application in construction management.

1.2 STATE OF THE ART

Practically each project is complex to some extent. In a large, complex project, there are hundreds or even thousands of operations and activities. Engineers use several techniques, with varying degrees of complexity, to handle project scheduling.

The object of scheduling process ensures that

1. A programmed rate of completion is met.
2. A constant rate of work is maintained.
3. Labour and plant move through the project in a continuous manner such that a balance labour force is maintained and kept fully employed.
4. The cost benefits of working are achieved.
5. There is total control over the execution of the project.

Bar charts or Gantt charts and mile stone charts were some of the early tools for project scheduling. While bar charts were improved into sophisticated networks, operation research techniques such as linear programming, simulation, time and motion studies, work-study methods, value engineering, statistical quality control, and inventory control were increasingly used. Essentially, the initial function of operation research was the analysis of existing operations to find more efficient performance methods. In 1956, the **critical path method (CPM)** was first formulated to schedule projects. In 1957, a technique called the **program evaluation and review technique (PERT)** was developed to integrate and coordinate contractors working on a single project. This method uses probability theory, and it is considered a stochastic critical path method. PERT enables

managements to plan and control projects by knowing the probabilities of occurrence of events. Then, a method called graphical evaluation and review technique was developed. GERT is the simplest way of showing the dovetailed operations in a construction project, and is useful when performance of all operations is not necessary for the completion of a project. Now-a-days, the use of these methods is increasing due to the ease of implementing them on computers

A historical evaluation of construction project scheduling and control in the past 25 years suggests that after the enthusiastic boom of the early 1960's, the use of network technique by construction companies has reached at steady level, after a slight decline in the early 1970's. There is evidence that contractors do not use networks in highly repetitive jobs because of their belief that high repetition would reduce the chances of successful scheduling and control by networks. The same concern is noticed in the literature and is an indication that linear scheduling methods were not, after all, replaced by network techniques but only continued to be seldom used, as was the case before the advent of networks.

All these methods can be broadly divided into two groups: deterministic and probabilistic. When the information needed for a particular method is assumed to be known during the analysis, it can be considered deterministic. Bar chart and CPM may be classified as deterministic methods. In reality, however, most of the information used in these methods is non-deterministic in nature. In other words, a particular value of a parameter, such as the duration of an activity, is not known with certainty. The

incorporation of uncertainty in the parameter in the project scheduling techniques leads to probabilistic methods. In these methods, each parameter is generally expressed in terms of mean, standard deviation, coefficient of variation (COV), and appropriate probability distribution. The mean value indicates the expected or average value of a parameter, e.g., duration. The standard deviation indicates the dispersion or scatter of the data from the mean value. The COV is a non-dimensional quantity, which is ratio of mean and standard deviation, and is a measure of uncertainty in the parameter. PERT and GERT can be classified as probabilistic methods.

Basically, whether the method is deterministic or probabilistic, all the parameters need to be estimated. However, some parameters may not be estimated properly, since some of the factors that affect these parameters cannot be quantified. Instead, they are qualified. Good or bad weather can be considered as a factor that influences the duration of an activity. However, future weather conditions can be at best described as good or bad, and there is no standard acceptable numerical value attached to this qualitative statement. Consequently, these factors were not properly incorporated in the past in the estimation of the parameter. For example, PERT require a subjective data interpretation and estimation of the duration of an activity in the form of most probable, pessimistic, and optimistic values. This subjective estimation procedure does not properly consider the different factors which affect the duration and may result in an inaccurate estimate, and, consequently, in project delays and losses.

Projects are divided into activities. The relationship and sequence of these activities are presented in the form of a network. Each activity requires a certain amount of resource, which may include time, labour, material, or money. The objective of a project manager is to find the combination of resources which will minimize the total cost of not only one activity but of all the activities involved in the project, and to finish the project on the time. In order to estimate the completion time of a project, the time required to finish each activity (duration of activity) need to be estimated. The nominal duration, or the mean value and the standard deviation of the duration or the probability distribution and its parameters of the duration of each activity need to be estimated, depending on which scheduling method is being used, i.e., CPM, PERT, GERT, VERT or simulation techniques. Obtaining reasonable activity duration estimates is important because all subsequent calculations and decisions are based on these estimates. There are many factors, which affect the duration of an activity, e.g., weather, labour skill (which change with time because of the learning effect), superintendent experience, types of equipments used, and level of operator's experience. The effect of these factors on the durations of an activity depends on the activity being considered. For example, the pouring of concrete in an open area is highly sensitive to weather conditions compared to other factors. The construction engineer or superintendent estimates the duration of the activities using experience and judgment. The level of experience and judgment will affect the final outcome and result in uncertainties in the durations; these uncertainties need to be modeled mathematically.

The major problem lies with the factors that are expressed in linguistic, rather than mathematical terms. Good or bad weather, long or short experience, etc., fall into this category. Even sensitivity of the activity's duration to any of these factors is measured in linguistic terms, e.g., highly sensitive, strong influence, etc. Not only are future weather conditions uncertain at the present time, the definition of good or bad weather complicates the problem. Uncertainties in future weather conditions can be modeled mathematically; however, additional sources of uncertainty due to the qualitative assessment of good or bad weather need to be considered. The linguistic variables can be translated into mathematical measures by fuzzy sets and system theory. Conventional procedure like PERT can be used, if updated probabilistic input is used to obtain the required information.

BAR CHARTS

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 2

BAR CHARTS

2.1 Introduction to Gantt (bar) charts

Bar chart or Gantt chart developed by Henry Gantt is the most commonly used technique of scheduling for small and medium sized projects. Figs 15.1, 15.8 and 15.9 show typical Gantt charts.

2.2 Critical review of the Gantt chart

2.2.1 Advantages of the Gantt chart

- (i) It gives the pictorial representation of the status of a project at any point of time.
- (ii) The overall performance of the activities can be judged from the bar charts quickly without going through the status reports.

2.2.2 Disadvantages of the Gantt chart

- (i) The interaction and nature of dependence of one activity on another is not reflected in the Gantt chart.
- (ii) It is not possible to identify the problem areas, when a project runs behind schedule.
- (iii) Gantt charts are very lengthy for a big project and modification becomes tedious.

CRITICAL PATH METHOD (CPM)

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 3

CRITICAL PATH METHOD (CPM)

3.1 Introduction to critical path method

Developed in 1956 to aid the planning and development of large projects, CPM is now widely used in industry and service. CPM uses two types of network diagrams, viz., activity-on-node (AON) and activity-on-arrow (AOA). The meanings of these two networks are explained by a simple house construction project (Jerome D. Weist and Ferdinand K. Levy, 1985) as given in *Table 3.1*.

Table 3.1 House Construction Project

Job name	Description	Immediate Predecessors	Time (days)
a	Excavate, pour footers	-	4
b	Pour concrete foundations	a	2
c	Erect frame and roof	b	4
d	Lay brickwork	c	6
e	Install drains	b	1
f	Pour basement floor	e	2
g	Install rough plumbing	e	3
h	Install rough wiring	c	2
i	Install air conditioning	c, f	4
j	Fasten plaster and plaster board	g, h, i	10
k	Lay finished flooring	j	3
l	Install kitchen equipment	k	1
m	Install finished plumbing	k	2
n	Finish carpentry	k	3
o	Finish roofing and flashing	d	2
p	Fasten gutters and downspouts	o	1
q	Install storm drains	b	1
r	Sand and varnish floors	n, s	2
s	Paint	l, m	3
t	Finish electrical work	s	1
u	Finish grading	p, q	2
v	Pour walks, and landscape	u	5

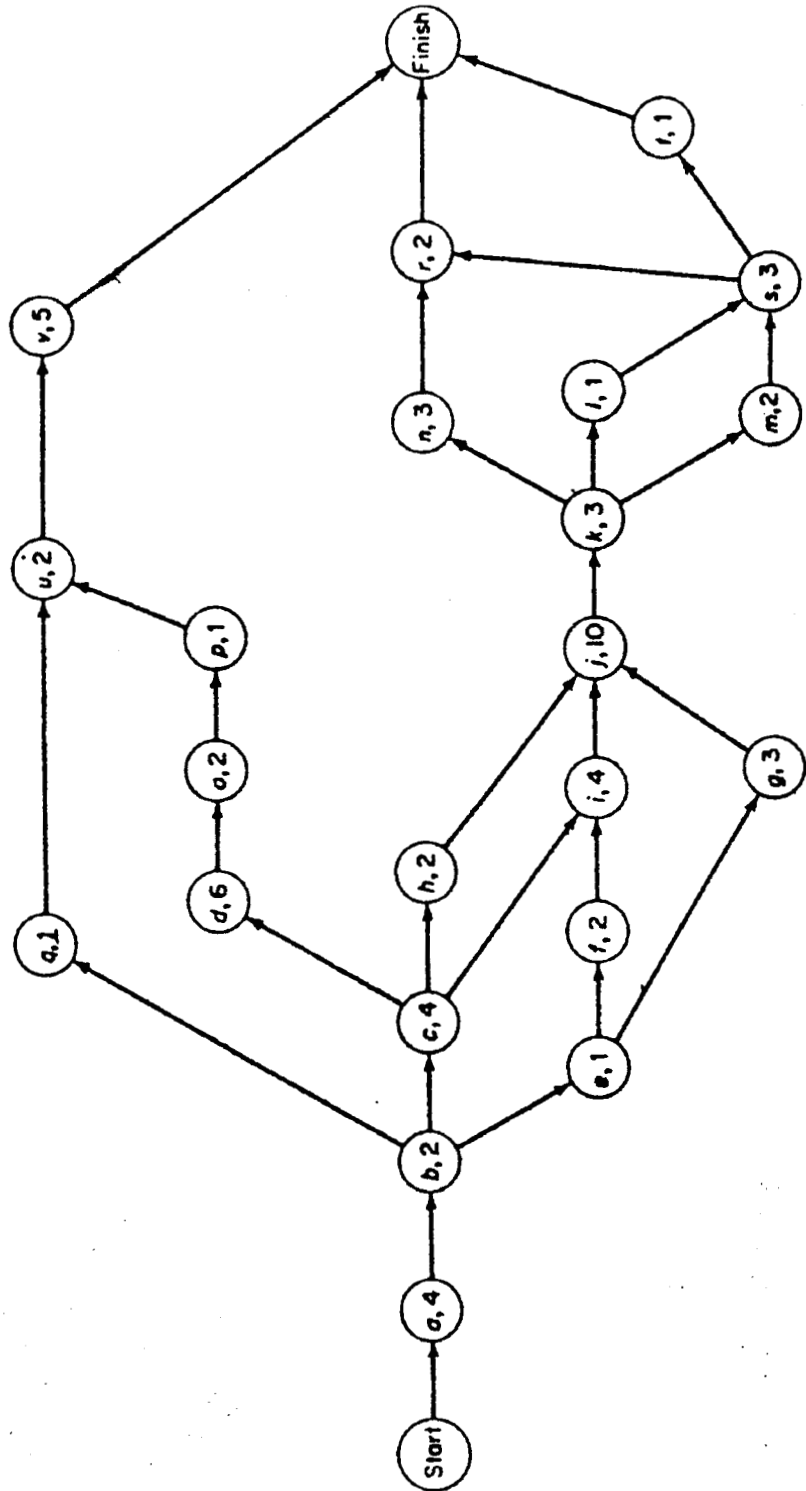


Fig 3.1

AON diagram for house construction project in Table 3.1

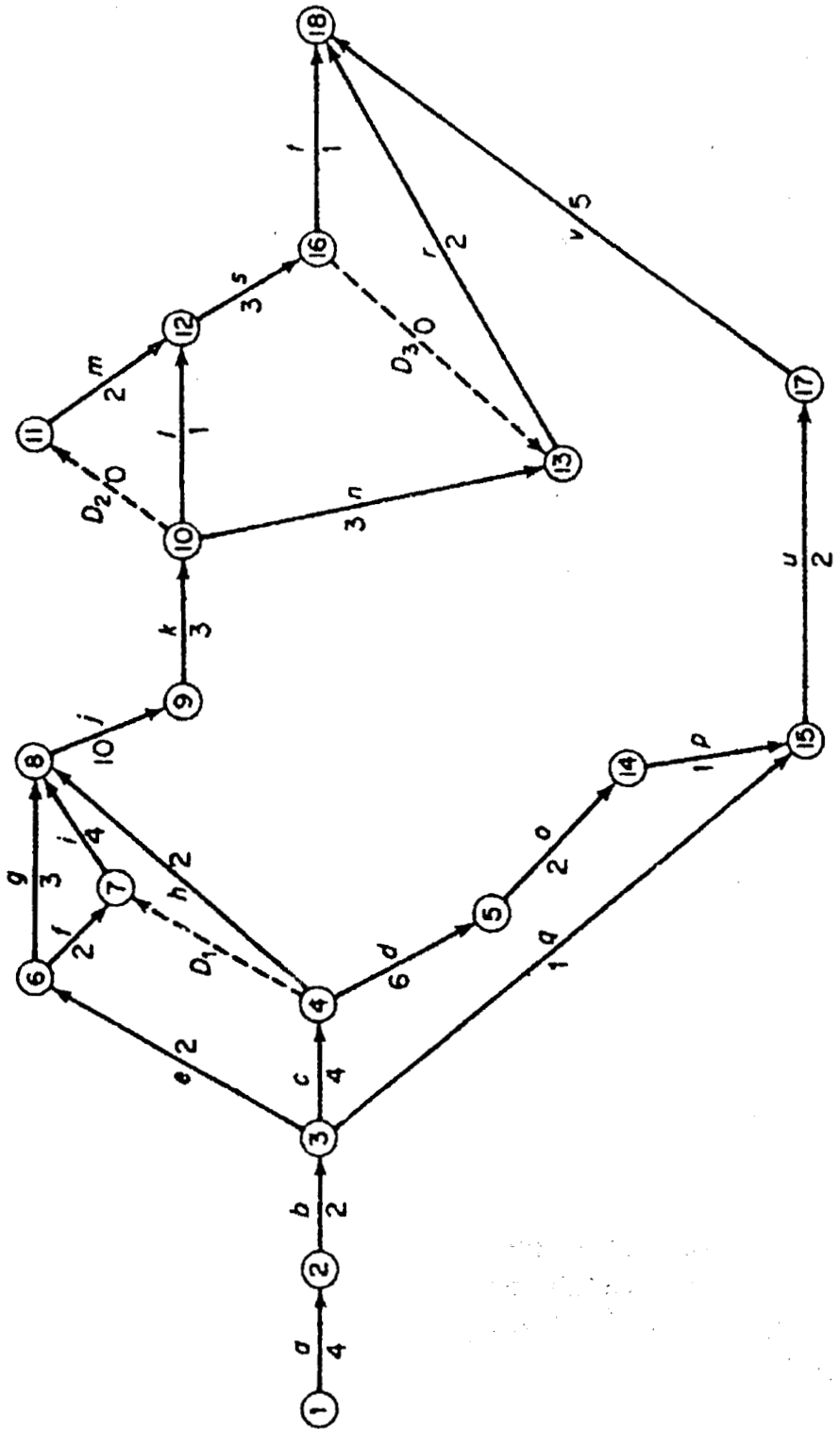


Fig 3.2 AOA diagram for *house construction project* in Table 3.1

(Note: D₁, D₂ & D₃ are dummy activities having no duration, used for connectivity)

3.2 Assumptions of the CPM model

1. A project can be divided into a set of predictable, independent activities.
2. The precedence relationships of project activities can be completely represented by a non-cyclical network graph in which each activity connects directly into its immediate successors.
3. Activity times are independent of each other.
4. The duration of an activity is linearly (and inversely) related to the cost of resources applied to the activity.

3.3 Perceived disadvantages of CPM scheduling

As seen from the preceding analysis, activities are constructed of tasks with continuity using known or devised construction processes. The final network is a plan, which can be as good as the amount of effort, which has gone into its formulation. There is nothing inherently wrong in either CPM concept or the subsequent schedules resulted from its analysis, the fault lies in the way it is applied in practice.

A number of publications in recent years (Ali Jaafari, 1984; Amir Tavakoli and Roger Riachi, 1990; Hoffmann. T.R and G.D. Scudder, 1983; Hughes. M.W.,1986; Kidd. J, 1989; Timothy Ch. U. Kalu, 1990) have advocated that different scheduling techniques

be developed to replace CPM. Argument put towards vary and following are the interpretations of some of these arguments:

1. CPM has not been developed to satisfy the planning needs in construction projects.
2. The basic assumptions of project activities having fixed time and discrete nature are unrealistic, especially when repetitive units or linear projects are to be constructed.
3. Construction planning essentially involves giving equal attention to all, and perhaps making most activities or processes critical (cost wise), and not just determining an incidental path related to the activity duration
4. Resource allocation, smoothing or leveling procedures are incapable of ensuring full continuity for a production crew or process which is the backbone of operational planning in construction processes, especially in repetitive cases.
5. CPM scheduling is expensive to run. Status reports take time to reach managers and decision makers, and, by the time they receive the information contained in them, tend to be out of date.
6. Practical integration of CPM based progress and cost control has been extremely difficult, expensive, and non-productive.

3.4 Factors affecting successful CPM scheduling

The two main factors are as follows:

Factor 1: Realistic estimation of the productivity of crews and adoption of a representative value for job-management efficiency factor (caution for value of $f > 0.5$).

Factor 2: Inclusion of sufficient time buffers between dissimilar trades; this, in fact, is to simulate actual field conditions, which indicate delay in start up of new operations.

Factor 1 indicates that the expected average productivity (or output per hour) of a given crew is equal to $f \times q$ in which q is the ideal productivity. Practical observations tend to show that, contrary to common assumptions and theoretical considerations (e.g., learning curve), the actual productivity in construction varies considerably at random; this is so even when the work is typically repetitive. Productivity shows a gradual build up at the start (often associated with unavailability of specific tools or materials at the time required, or lack of foreman instruction, or absence of key craftsmen). Steady progress in productivity continues unless interrupted externally, then followed by unexplained drag at the end, or often unfinished 10 – 15 % for a variety of reasons such as urgent start elsewhere, technical problems, or breakdown in tools.

Despite these variations, one conspicuous result is of interest, i.e., over a sufficiently long duration - and provided no major external disturbance is observed - the average actual productivity tends to be close to the expected value, $f \times q$.

Provision of time buffers between principal processes further ensures a realistic plan. These are practical hints for those who are committed to sound planning and scheduling; as every practitioner knows well, decision which are made by planners and

managers in specific projects affect the schedule. There is no law to predict such patterns of decision. Even in an obvious case as the excavation of the pit for the basement in the building project, which is normally carried out at the onset of the project, there are exceptions.

A point may be concluded from this review, i.e., the planners induce their own critical path through the set of decisions they make at the time of planning, thus the critical path is neither irrelevant nor incidental (e.g., they decide to build the structure before the pipe work). Some time technological constraints must be followed, e.g., columns must be built before floors, but more often preferences are reflected in interfacing of activities, some for pure convenience, other for economical reasons. Some times dissimilar trades may prefer to move in opposite directions. The movement of crews is not necessarily sequential through adjoining sections.

These facts show that there is sense in designating a stream of operations by the same crew within a section or block of construction as a discrete activity interfaced with relevant activities in the preferred sequence.

3.5 Can linear projects be scheduled using CPM

Some authors (Adel M. Salem and Salah E. Elmaghraby, 1984; Edmund N. Chrzanowski Jr. and David W. Johnston, 1986; Hand V.K. and Barcia R.M., 1986; Poiaga. L, 1989; Punmia. B.C. and K.K. Kandelwal, 1988; Ragsdale. C, 1989) argue that,

in the case of linear project (highway and pipelines), it is advantageous to use linear scheduling method. Even in these situations, CPM scheduling is not only relevant, but also most constructive. This is because these constructions are seldom as uniform in cross-section or material composition as they first appear to be. Moreover, there are bridges, etc., which, in addition to the usual ground undulations, interrupt the continuity of work. These facts mean different work content per unit length, so the linear synchronized scheduling is of little practical interest. Is it not more sensible, for the purpose of planning their constructions, to divide linear structures into discrete sections at these convenient points? Any continuity of work of a crew can be maintained by simply linking the end of the work in one section to the start of the crew's work in the next (which, depending on the decision made at the time of planning, may or may not be the adjoining section).

As the cross section and material composition in each discrete section tend to vary, the speed and duration for each crew can be estimated more realistically in considering all the relevant facts in that section and using *Factor 1*. Implied in this procedure is the fact that, even though projects are considered as discrete sections, activities are not solely related to individual sections but are designed for continuity of the work of a crew passing through sections in what ever sequence one may prefer. What is important is that an unrealistic assumption of 'synchronized' processes being planned in successive sequences is precluded.

3.6 Is CPM scheduling capable of fulfilling all required functions?

The answer is, of course, a qualified yes as seen from the following:

3.6.1 Offsetting uncertainty and change

Productivity of each crew fluctuates considerably even in a repetitive operation. However, if activities are planned in broad terms, the CPM schedule will be less sensitive to variations in daily performance. Major disturbance can be measured easily by this tool and the effects expressed in units of time and cost.

Flexibility in the CPM schedule can be further secured by not rescheduling non-critical activities to level resource histograms. The total number of activities must, however, remain manageable (50-500) and the inter linkage kept to an absolute minimum.

3.6.2 Focusing attention on objectives

In the present inflationary economic environment, time is often the essence; the economic viability of a project is crucially dependent on shortening the engineering phase even at the expense of increasing construction cost by a modest percentage.

As a major emphasis in CPM scheduling is on criticality of the project duration, it will naturally concentrate attention on to criticality of principal operations; thus, it

meets the promoter's objectives. The emphasis on time will also benefit the contractor in the following ways:

1. **Work for motivation:** If the urgency of the project and the importance of its success are communicated to the workforce and site personnel, they tend to improve their motivation (sense of mission).
2. **More thorough study of the work by the site management team at the time of planning prior to scheduling and, thus, more awareness of the critical items and external dependencies in the project.** This helps the site managers to 'know' the project in detail before the work is actually done.
3. **The emphasis on meeting milestones in the course of a project increases the pressure for orderly functioning and sound logistics.** These will have economic benefit to the contractor.
4. **Financial incentive:** this applies to the case of contracts based on target completion dates with bonus for early completion and liquidated damages for late completion date. This practice is becoming popular

3.6.3 Gaining economical operation:

As far as the actual planning is concerned, enough emphasis must be placed on continuity of work of the crews when considering the processes and the sequence of activities. Site productivity must be monitored subsequently using work-sampling

techniques to ensure that planned processes are effective, and that no bottlenecks interfere with the performance of crews and their associated equipment.

If a particular process planned beforehand was found to be expensive or incapable of meeting the scheduled date, improvement may be sought. For short activities of an odd nature, this is unnecessary and impracticable. For activities greater than 4-6 weeks in duration a simple activity sampling exercise, which is an inexpensive way of getting an estimate of the actual value of job-management efficiency factor f , in the field, is a good way of monitoring productivity (It does not reflect effectiveness of the work by personnel and equipment).

3.6.4 Facilitating control.

As every project manager in the construction industry well knows, control of the total construction process not only involves the control of the actual performance of the crews on site, but also items such as provision for material, owner-furnished equipment or materials, shop drawings, sample approval and a host of other interrelated activities. As has also been confirmed by many authors, there is no alternative method for control of the total process but a logical network.

Perhaps development of more user friendly or interactive system that can be operated on a single work - station computer on the site can assist in speedier control of the project.

3.6.5 Allocating contractual responsibilities

As the network is an interrelated time – event model of the project, it is the best means of allocating, communicating, and controlling contractual responsibilities within the engineering phase of the project.

Obviously, the eventual network must incorporate contractors' preferred sub networks because, irrespective of the type of work and locality, different contractors plan their operations in different ways.

3.6.6 Coordinating contributions from various groups during engineering phase

Many of the projects undertaken today are complex and require input from a variety of disciplines. The only formal method of coordinating such an extensive array of works and services is through an interrelated network of activities.

3.6.7 Resolving delay and change order disputes on a predefined, quantifiable, and equitable basis

The very concept of critical path can be used to resolve claims related to delays. Non - critical activities may be exempt from time extension claims up to the time they become critical (subject to prior agreement). Any additional cost the contractors incurs, as a result of a change order or owner's interference must, however, be compensated.

Events beyond the control of the owner and contractor qualify for contract time extension but without monetary compensation to either party.

In all these cases the CPM schedule can be used to resolve both delay and change order dispute provided the schedule represents a reasonable and feasible plan, and it has been instituted as the control tool throughout the project. No other viable alternative is available to owners and contractors for estimation of the delay and change order or any other disturbance. (Note that the network allows quantification of time-related costs and estimation of claims on a present value basis.)

It is seen that, despite much criticism, there is nothing inherently wrong with the concept of CPM scheduling. Failure to see the project in its entirety, and to plan the activities in terms of continued processes, may result in total failure, which may be blamed on the scheduling technique. Conditional to the success of CPM scheduling are two factors: realistic estimation of productivity of crews in the context of expected job management efficiency conditions, and inclusion of sufficient time buffers between dissimilar trades or crews. Linear and the other projects are best scheduled using CPM: the network may be assumed to comprise a set of activities each of which may comprise a chain of repetitive operations performed by a composite crew of known productivity. The continuity of work of crews may be secured by interfacing the relevant activities in the preferred sequence. Although all crew must be planned to complete their work within known dates, it is unrealistic to schedule this in uniform speed since the size and composition of a crew relate, primarily, to the process for which it is designed. Site

control of productivity and conformity with plans is best carried out using work - study techniques; for activities of short duration and of an odd nature it is impracticable to do so. For major operations over periods greater than 4 weeks, work - study and sampling technique may be employed to ensure full productivity of the process selected beforehand and removal of all possible bottlenecks. In extremely inefficient processes, a change of process may be necessary to meet the target date or lower the costs. For CPM planning and scheduling to fulfill the functions required in planning and controlling of the engineering phase of construction projects, one must pay attention to the organization, people, and their attitude. Involvement of people and inadequate knowledge of the future means that planning must be a dynamic process - ready for evaluation of the effects of disturbance (and there is a high probability that after initial briefing and definition, the redefinition of the project's functions and its component parts will continue down to the last possible moment). CPM scheduling, if kept in manageable size and operation on interactive software using a single work - station computer (and not data bureau), can respond well to this need. Resource leveling at the initial stage of planning is useful, as in any other simulation or experimentation. However, if relevant operations are interlinked properly, there is very little to gain from such experiments except, perhaps, rescheduling of non - critical activities. These activities must be left with float to allow managers to utilize them for keeping the usual 'floating' or temporarily unoccupied resources engaged. This is an insurance against inelastic state of supply of resources in construction projects.

DECISION CPM

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 4

DECISION CPM

4.1 Introduction

Critical path analysis is commonly considered to be a technique for planning and scheduling of projects. The planning phase is usually identified with the construction of the method of performing jobs as well as their technical ordering. At the same time standard times are assigned to these jobs. At the completion of the planning stage it is possible, using the conventional CPM calculations, to schedule the starting time of each job in the project. Unless several different planes are evaluated in this way, or unless the technique of job crashing is used, there is no interaction between the planning and the scheduling phase of the usual CPM analysis.

Thus if there are a number of competing methods of performing some of the jobs, each method having a different cost, a different time duration, and different technological dependencies, we shall include all these in the project graph rather than making the decisions in advance. Then in the scheduling phase we shall consider the effect of all alternate methods of performing a task on the total cost of completing the project, and choose those alternatives that minimize the cost. We may apply the same method to the control of projects being carried out. Thus decisions previously considered to be optimal, may be changed after partial completion of the project because of the delay in completion of certain jobs. The complete problem is called Decision CPM (DCPM), and it can be set up as a formal mathematical problem to solve using various techniques.

A project graph is defined as that containing information of all jobs to be completed in the projects, alternative methods of performing some of the jobs, and

precedence relations between the jobs. Integer programming and heuristic techniques can be used for solving such graph for the set of jobs that are to be performed and the criticality of these jobs. The technique can be applied to dynamic monitoring and control of projects during their execution.

4.2 The mathematical basis of decision CPM

Let $J = \{S_1, S_2, S_3, \dots\}$ be a set of *job sets* that must be done to complete a project. Some job sets are unit sets $S_i = \{S_{i1}\}$ and other job sets have several members, $S_i = \{S_{i1}, S_{i2}, S_{i3}, \dots\}$. In order to complete the project, some of the jobs from each job set must be completed. Associate with each job set

$$S_i = \{S_{i1}, \dots, S_{ik(i)}\} \quad (4.1)$$

$k(i)$ variables

$$d_{i1}, \dots, d_{ik(i)} \quad (4.2)$$

having property that

$$d_{ij} = \begin{cases} 1 & \text{if job } S_{ij} \text{ is to be performed,} \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

If exactly one of the jobs in S_i must be performed (this may or may not be the case) then S_i is said to require the mutually exclusive interdependence condition. Clearly the mutually exclusive interdependence condition is expressed by

$$\sum_{j=1}^{k(i)} d_{ij} = 1 \quad (4.4)$$

Many possibilities also exist.

If all job sets are unit sets, then the project reduces to the ordinary project of the usual CPM variety. If one or more of the job sets have more than one member, then for each such set a decision must be made as to which job of the set is to be done. Once such a decision is made for each job set, the result is an ordinary CPM project.

It should be noted that the decisions might be complicated by many other kinds of conditions than (4), which may be of the mutually exclusive or contingent kind. For instance, the following equations give examples of such interdependencies among decisions.

$$\begin{aligned} (a) \quad & d_{ij} + d_{mn} \leq 1, \\ (b) \quad & d_{ij} \leq d_{mn}, \\ (c) \quad & d_{ij} \leq d_{mn}, \\ (d) \quad & d_{ij} \leq d_{hk} + d_{mn} - d_{hk}d_{mn} \end{aligned}$$

Finally the design problem may not always be the simple choice of one job from each set.

For instance,

$$(e) \quad \sum_{j=1}^{k(i)} d_{ij} = 2,$$

$$(f) \quad \sum_{j=1}^{k(i)} d_{ij} \leq 5$$

$$(g) \quad \sum_{j=1}^{k(i)} d_{ij} + \sum_{j=1}^{k(l)} d_{lk} = 3$$

Note that (e) says that we must choose exactly two alternatives; (f) says we may choose at most 5 alternatives; and (g) says that at the two decision nodes i and l we must choose exactly 3 alternatives.

The above discussion illustrates some of the possible richness of problem formulation that is possible within the Decision CPM framework.

In addition to the relations described above there will be precedence relations between the jobs of a decision project. Let ' \ll ' denote a relation between pairs of jobs in J such that $S_{ij} \ll S_{mn}$ is defined for some pair of jobs S_{ij} , S_{mn} and is read S_{ij} is an immediate predecessor of S_{mn} . The interpretation of this statement is that all immediate predecessors of a job must be completed before that job can be started. A decision project is the set J together with the specified interdependencies and relation \ll defined on J .

The decision project graph of a project, G , is a graph with nodes representing jobs and directed line segment, connecting two nodes S_{ij} , S_{mn} if and only if $S_{ij} \ll S_{mn}$ holds. A path in G is a set of nodes connected by immediate predecessor relations. A cycle in G is a closed path of the form

$$S_{ij} = a_1 \ll a_2 \ll \dots \ll a_n = a_1 = S_{ij}.$$

A project graph is acyclic if and only if it has no cycles.

$S_{ij} \prec S_{mn}$ implies that S_{ij} precedes S_{mn} if and only if there is a set of jobs $\{a_1, a_2, \dots, a_n\}$,

$n > 2$ such that
$$S_{ij} = a_1 \ll a_2 \ll \dots \ll a_n = a_1 = S_{mn};$$

In other words, S_{ij} precedes S_{mn} if and only if there is a path from S_{ij} to S_{mn} in the decision project graph G .

The precede-relation is asymmetric, that is if $S_{ij} \prec S_{mn}$ then it is false that $S_{mn} \prec S_{ij}$ for all S_{ij} and S_{mn} in J .

A relation that is transitive and asymmetric is said to be a preference relation. A technologically ordered job list $J = \{a_1, a_2, \dots, a_n\}$ is obtained from a set of jobs $J = \{a, b, c, \dots\}$ by listing them so that no job appears on the list until all of its predecessors have already appeared.

In addition to these definitions and theorems several additional conventions are necessary because of the facts that some jobs may be eliminated from the decision project graph as the result of decisions that are made. If we decide to do one of the jobs in a job set, then all immediate predecessor relations that the job satisfies must hold in the final graph. In the decision project graph if we decide not to do a given job, then we must remove that job together with all edges that impinge on it from the decision project graph to obtain the final project graph. It follows from this that if any job, S_{ij} , has a sole

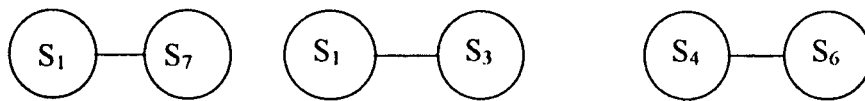
immediate predecessor, S_{mn} , and if that predecessor is a member of a job set, it will be necessary to create a dummy immediate predecessor relation between S_{ij} and a job that is a predecessor of S_{mn} . If this is not done, then it would be possible for the path containing S_{ij} to be broken and S_{ij} would lose its project time ordering. Similarly a dummy immediate successor relation must be established for jobs having only one immediate successor, if that successor is a member of a job set. In addition it may be necessary to create a dummy relation between two jobs even if both have several immediate predecessors and successors. If on any path, two jobs are separated by a job that could be eliminated, and if it is desired to maintain a technological order of the two jobs, a dummy immediate predecessor relation must be established between them.

For a given project when the jobs are technologically ordered and all planning decisions are made designating the jobs to be performed in each set, the normal critical path analysis may then be carried out. The usual concepts of early start, late start, critical path, etc. will apply to this reduced graph.

4.3 Decision project graphs

A graphical representation of the combined planning and scheduling problem is shown in the decision project graph of Fig 4.1 and Fig. 4.5. In these graphs the circular nodes represents jobs and the triangular nodes introduce the mutually exclusive job nodes of a job set. In Fig 4.1 there is the additional interdependence of a contingent relation between the jobs S_{51} and S_{22} . We may include job S_{22} if and only if we perform job S_{51} .

Therefore the possible sets of decision are $\{S_{21}S_{51}\}$, $\{S_{21}S_{52}\}$ or $\{S_{22}S_{51}\}$. The project graphs resulting from each of these sets or decisions are shown in Figs 4.2, 4.3, and 4.4 respectively. It should be noted in Fig.4.1 that the links are only necessary because jobs S_{21} and S_{51} are members of job sets.



The total project cost, given any set of decision will vary with the due date established because of overtime penalties and early finish premiums. For example if jobs S_{21} , S_{52} are selected the cost of performing all jobs would be Rs.1650 thousand. Since it would require 97 days to finish the project, a due date of 98 days would give the project a total cost of Rs.1650 thousands. However given a daily penalty and premium of Rs.150 thousands and Rs.25 thousands respectively a due date of 97 would cost Rs.1650 thousands + Rs.150 thousands = Rs.1800 thousands and a due date of 101 days would cost Rs.1650 - (3) (25) = Rs.1575 thousands. Table 4.1 illustrates how total cost changes with due date for each set of decisions. It also indicates that the optimal decisions (as indicated by the boxes) vary with the given due date. This example demonstrates the difficulty of making decisions in this project without the knowledge of scheduling information and due date.

Table 4.1 Decision set – total cost of project with given decision sets and due date

Due date	$S_{21}S_{51}$	$S_{21}S_{52}$	$S_{22}S_{51}$
101	1670	1575	1535
99	1720	1625	1710
97	1770	1800	2010

It is relatively simple matter to solve very small decision project graphs by making a complete listing of alternatives. It is clear however that the number of possible combinations rises rapidly. For example the decision projects graph of Fig 4.5 has a total of 864 possible decision patterns. It has therefore been necessary to develop other techniques to handle large problems.

4.4 Decision graph solution by integer programming

Consider a job set (4.1) and its associated decision variables with constraints given by (4.2), (4.3) and (4.4). Besides these, there may be any of the other constraints showing various types of complicated interdependencies between jobs in the project.

As in a graph of Fig. 4.1, we associate with each job, S_{ij} , a time, t_{ij} , and a cost, C_{ij} . Also we assume a reward payment of ' r ' thousand rupees per day for each day under D , the required due date of the project and a penalty payment ' p ' for each day beyond D . we

can now formulate the integer programming problem of selecting the best project graph and finding its critical path.

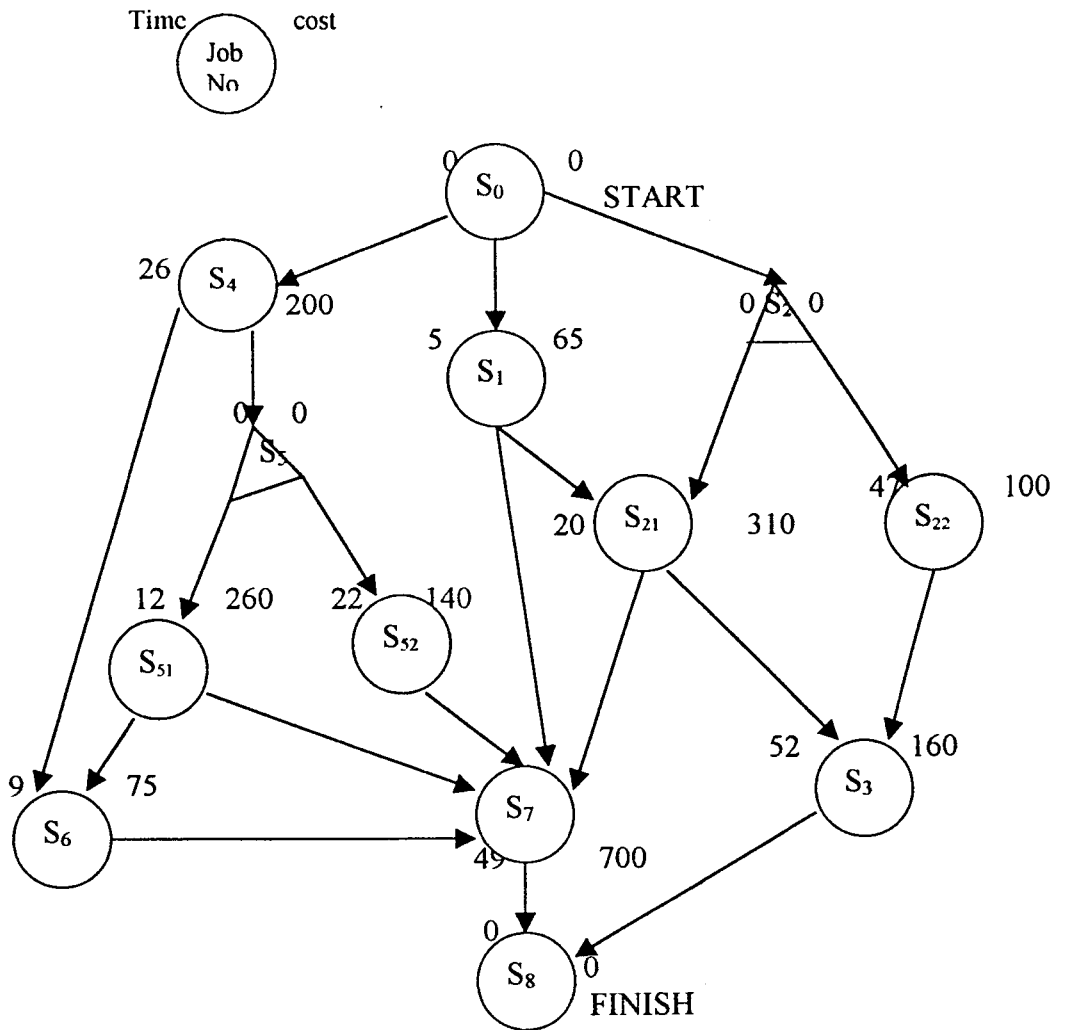


Fig.4.1. Decision project graph No. 1

$$\text{Min} \sum_{i=1}^{i=h} \sum_{j=1}^{j=k(i)} d_{ij} c_{ij} - rW_F^- + pW_F^+ \quad (4.5)$$

The first term calculates the costs of all the decision jobs that are to be performed. It is governed by the constraints

$$0 \leq d_{ij} \leq 1, \quad (4.6)$$

$$\sum_{j=1}^{j=k(i)} d_{ij} = 1 \quad (4.7)$$

Where d_{ij} is an integer, the second term is explained by the constraint

$$W_F - W_F^+ + W_F^- - D \leq 0, \quad (4.8)$$

Where W_F is the early start time of Finish, the last job in the project. If $W_F > D$ then the project is not to be completed until after the due date so that $W_F^+ = W_F - D$ and a penalty of pW_F^+ is incurred in the objective function.

We assume that $p > r$ so that not both W_F^+ and W_F^- will be in the basis.

Other constraints must hold because of precedence relations. For instance if S_i and S_m are unit set jobs and $S_i \ll S_m$ we have

$$W_i + t_i \leq W_m \quad (4.9)$$

Where W_i is the early start time of job S_i . If S_m is a unit job set and S_{ij} is from a multi job set and $S_{ij} \ll S_m$ then

$$-M(1 - d_{ij}) + W_{ij} + t_{ij} \leq W_m \quad (4.10)$$

Where M is a large enough number so that the inequality is restrictive only if $d_{ij} = 1$.

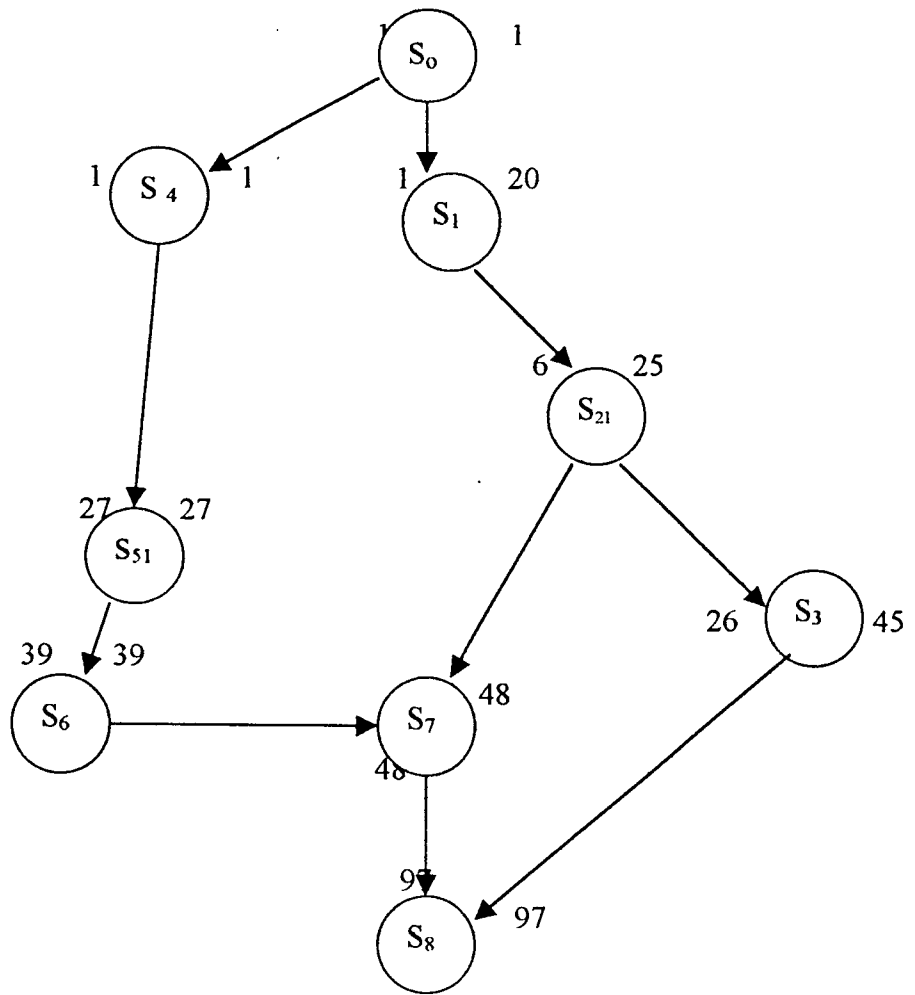


Fig. 4.2. Project graph 1 – decisions S_{21} , S_{51} .

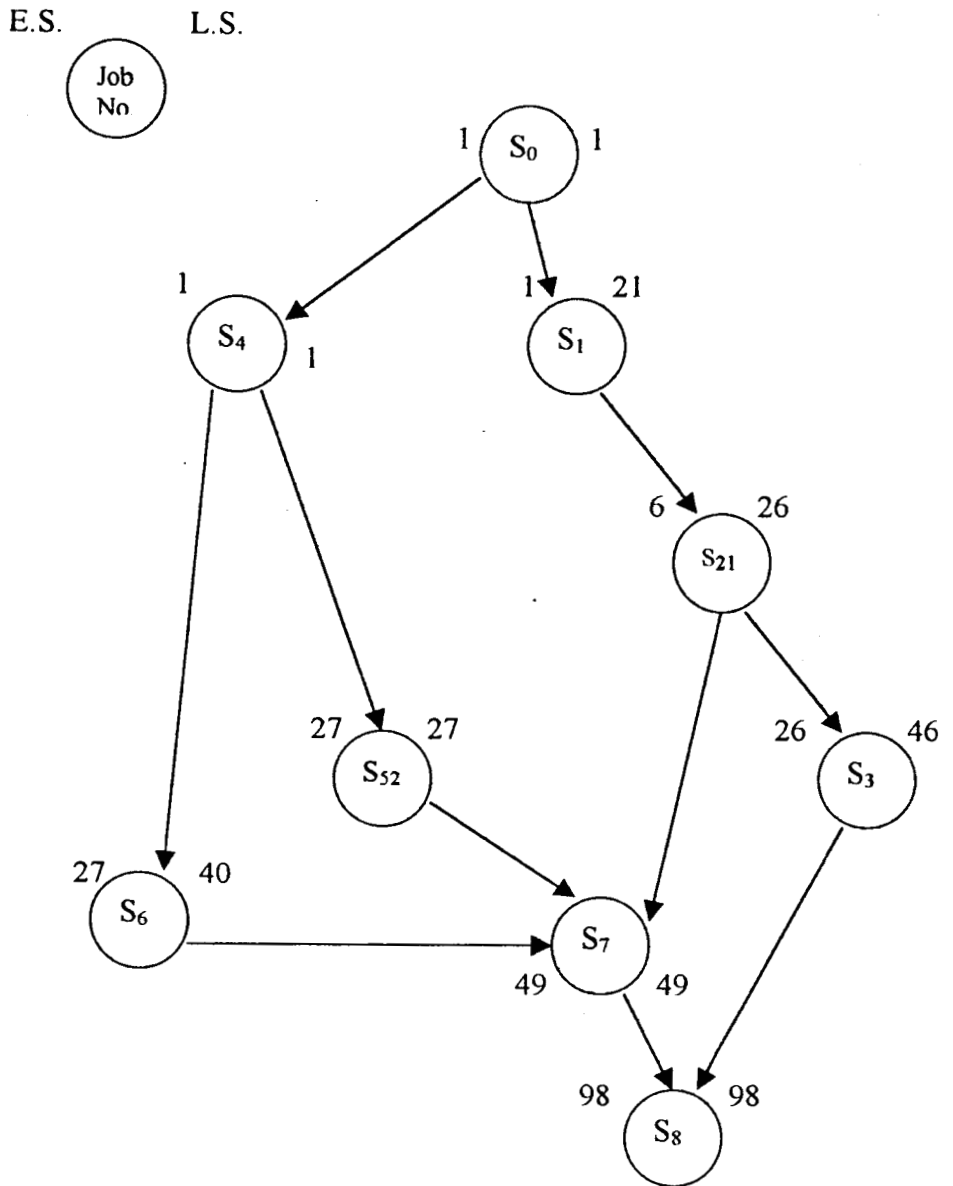


Fig 4.3 Project graph 1- decisions S₂₁, S₅₂

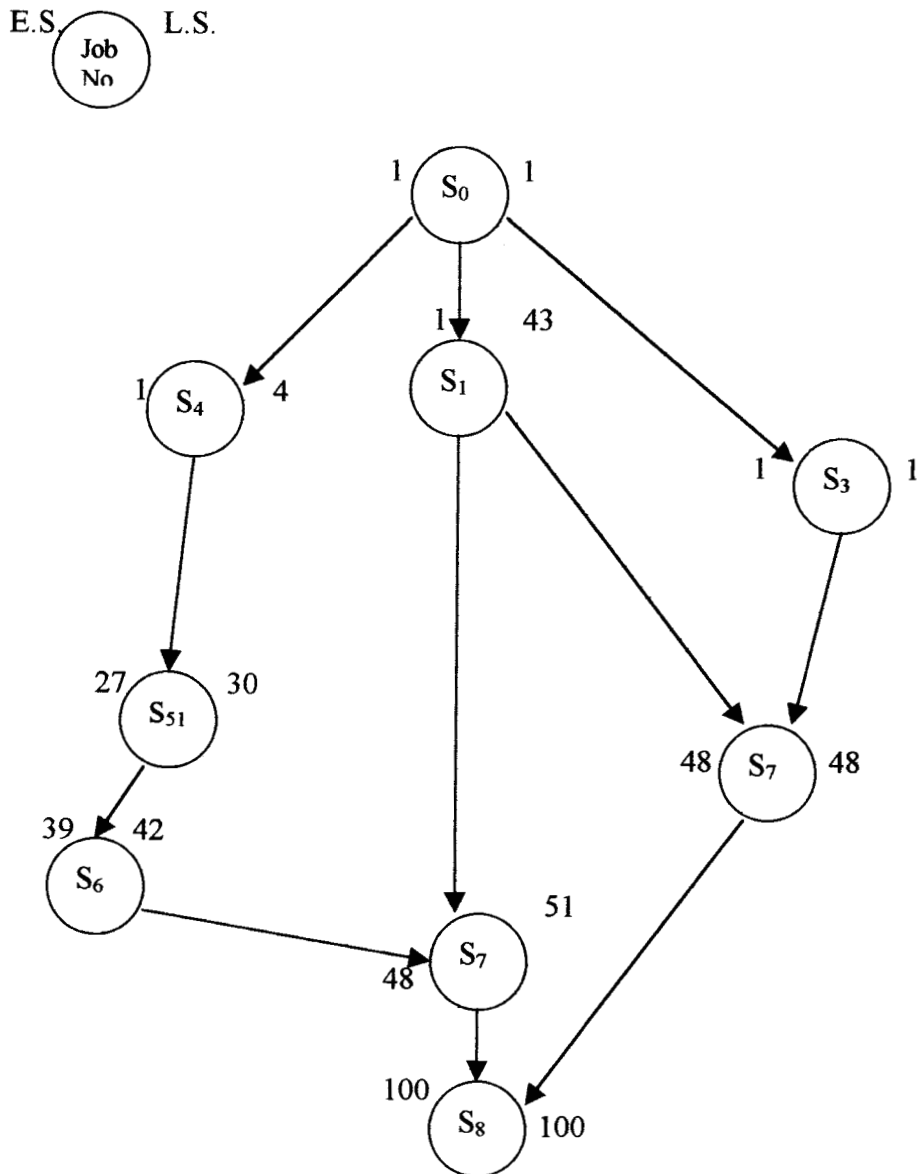


Fig 4.4. Project graph 1 – decision S_{22}, S_{51}

If S_{ij} is not performed (i.e., $d_{ij} = 0$) the inequality does not constrain the variables. Thus all paths through jobs that are not performed will be broken.

4.5 Decision graph solution by heuristic techniques

The first integer programming formulation of the preceding section requires a constraint for each link of the graph (except for the links emanating from the start node or a decision preceded only by the start node), one constraint for each multi job set plus a final due date constraint. In both cases there may be interdependence constraints as well. In very large problems the integer programming solution technique became impractical because of the resulting large number of constraints and variables. For that reason it is advisable to develop heuristic solution techniques for solving the problem. The heuristic methods are designed to handle graphs containing only the mutually exclusive type of interdependency. Further more, all jobs in a job set S_i are assumed to be bound by the same precedence relations, and jobs in a decision set have the following cost and time relations.

$$t_{i1} < t_{i2} < \dots < t_{ik(i)},$$

$$c_{i1} > c_{i2} > \dots > c_{ik(i)}.$$

The heuristic routine contains the following steps.

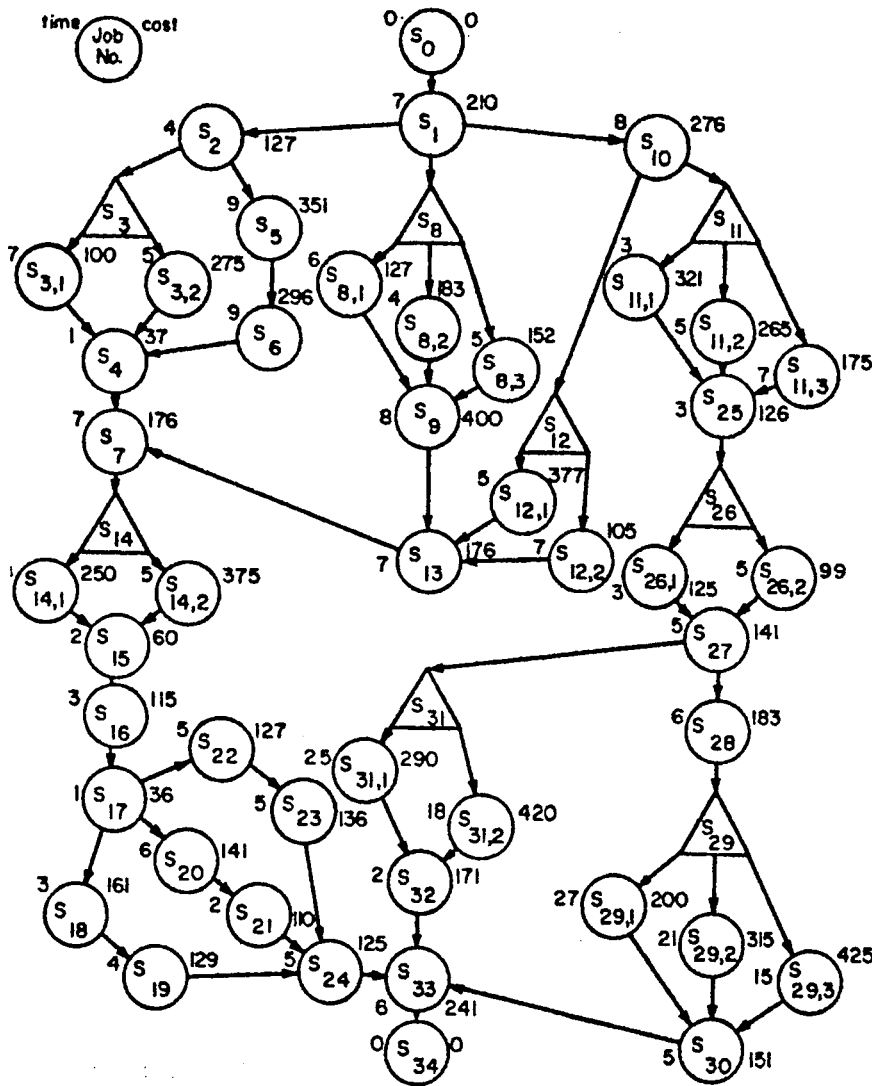
- (1) Technologically order the jobs.
- (2) Set each decision node to the alternative having the lowest cost.
- (3) Calculate the critical path.
- (4) Reorder by Early start (also a technological ordering).
- (5) Go to (7).
- (6) Recalculate the critical path starting at the position in the ordered job list held by the decision node of step (10).

- (7) Identity all decision nodes on the critical path.
- (8) For the all the nodes of step (7) calculate the net reduction in total project cost achieved by substituting the more costly alternatives.
- (9) If no alternative reduces overall cost, go to (12).
- (10) Find the alternative that gives the maximum cost reduction and switch the relevant decision node to that alternative.
- (11) Go to step (6).
- (12) Review all decision nodes that were previously changed to see if sufficient slack has been regenerated to allow the reintroduction of a longer but cheaper alternative, HALT.

Some explanation is required for several of the steps. First in (7) it is only necessary to examine decision on the critical path since for decision not on the path any reduction in the job time would have no effect on the penalty cost. Secondly, the method of calculating cost reduction in (8) is as follows. The additional cost of switching to a more expensive alternative is simply the difference in costs. The saving to be made from reduced over time penalty on increased premium is not straightforward. The total length of the critical path is not necessarily reduced by the same number of days as the job length is reduced since a parallel chain might become critical if the job is reduced in length by one or more days. It is therefore necessary to check the slack available in all parallel chains and calculate the reduction as the smallest value of slack found.

In the method, the maximum 'shrink' or minimum 'total slack' is calculated as follows. Given the decision node S_i find the slack of all jobs S_{mn} such that EARLY

$START_i \leq EARLY\ START_{mm} \leq LATE\ FINISH_i$. Given these conditions no job selected could be a successor or predecessor of S_i , and from all chains not containing S_i , at least one job will be selected.



Due Date: Day 70
 Early Premium: 250 thousand/day

Late Penalty: 475 thousand/day
 Possible Solutions: 864

Fig 4.5. Decision project graph No.2.

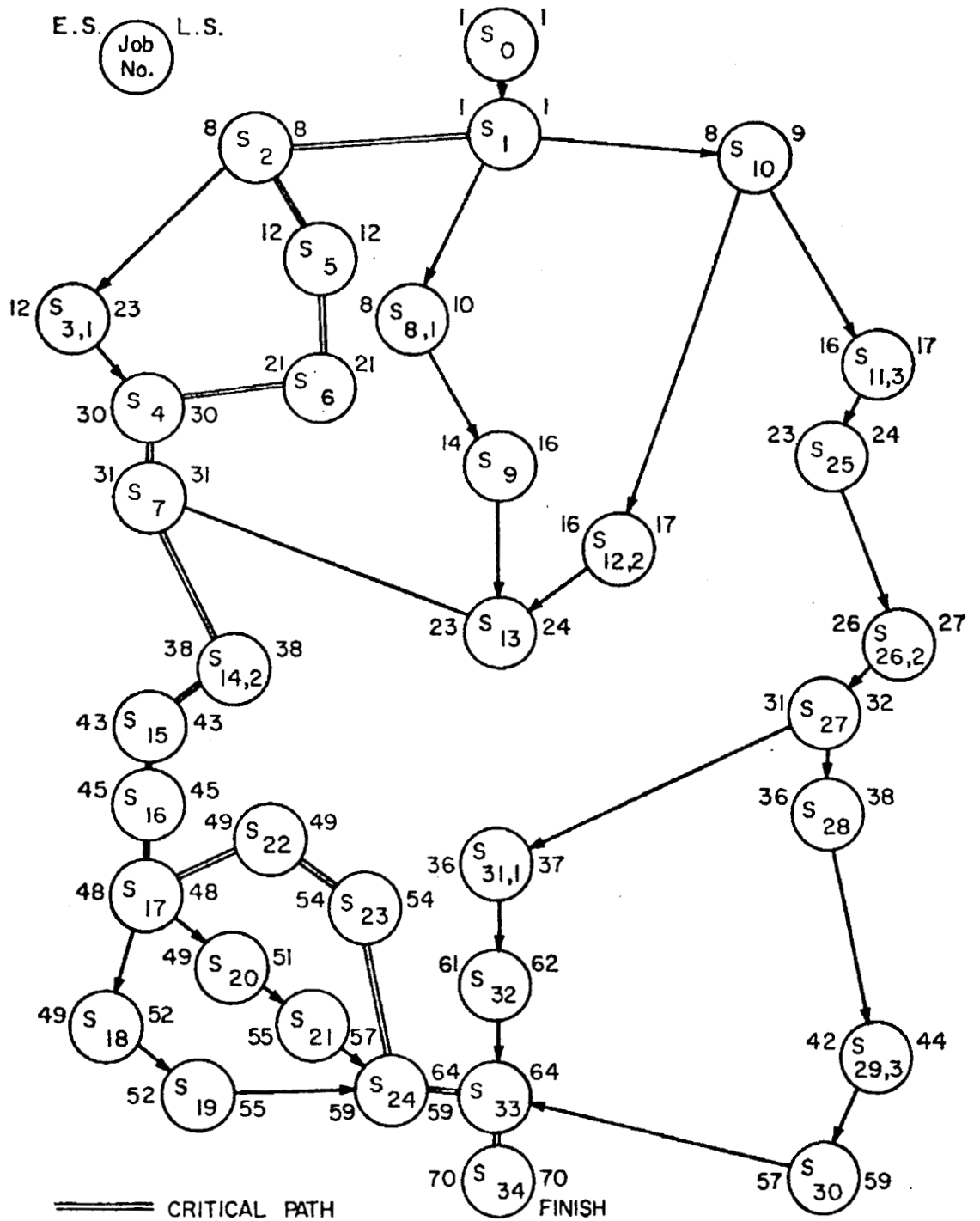


Fig 4.6

Decision project graph No. 2 - optimum solution

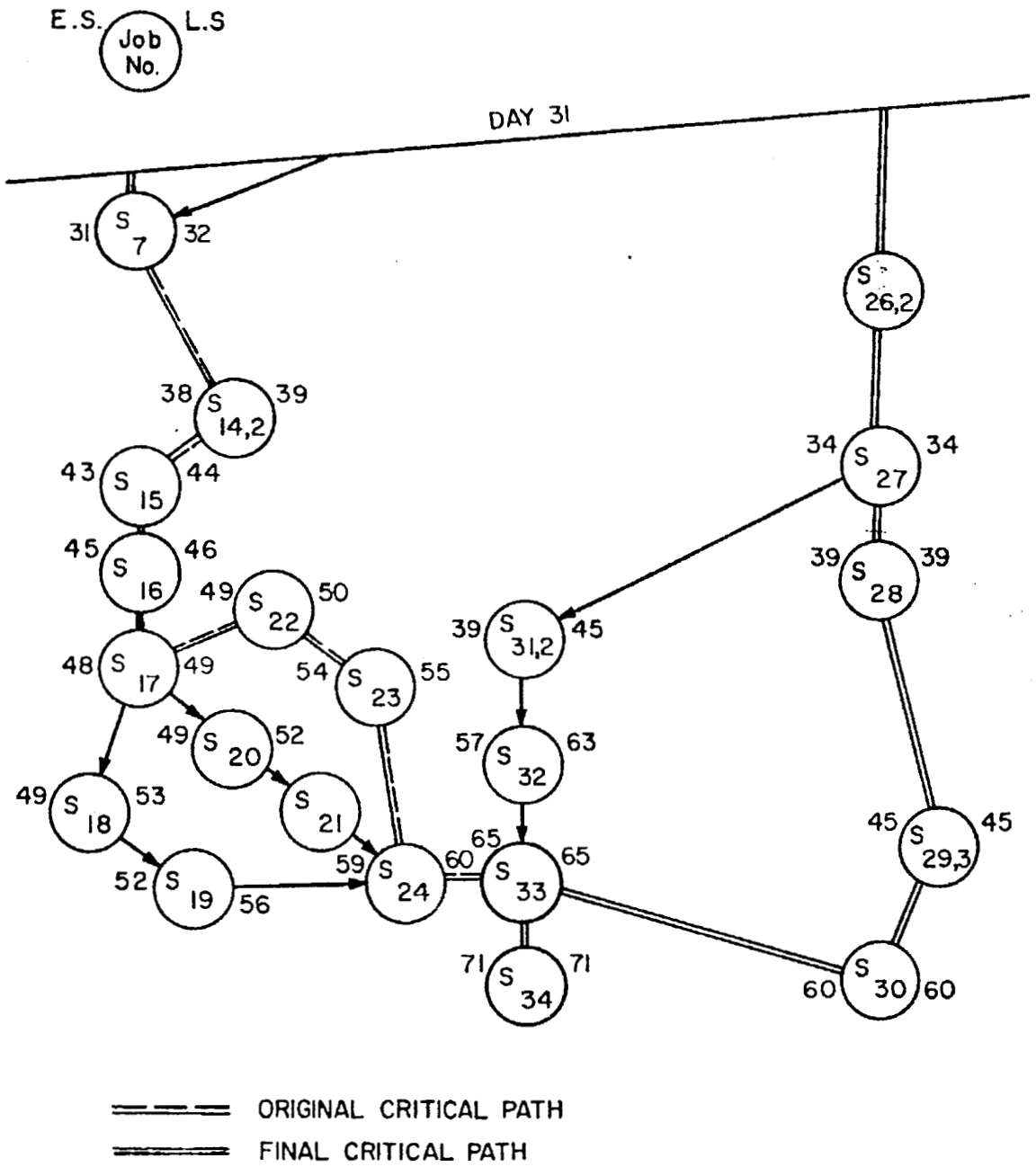


Fig. 4.7 State of project graph at the beginning of day 31 in Fig 4.5

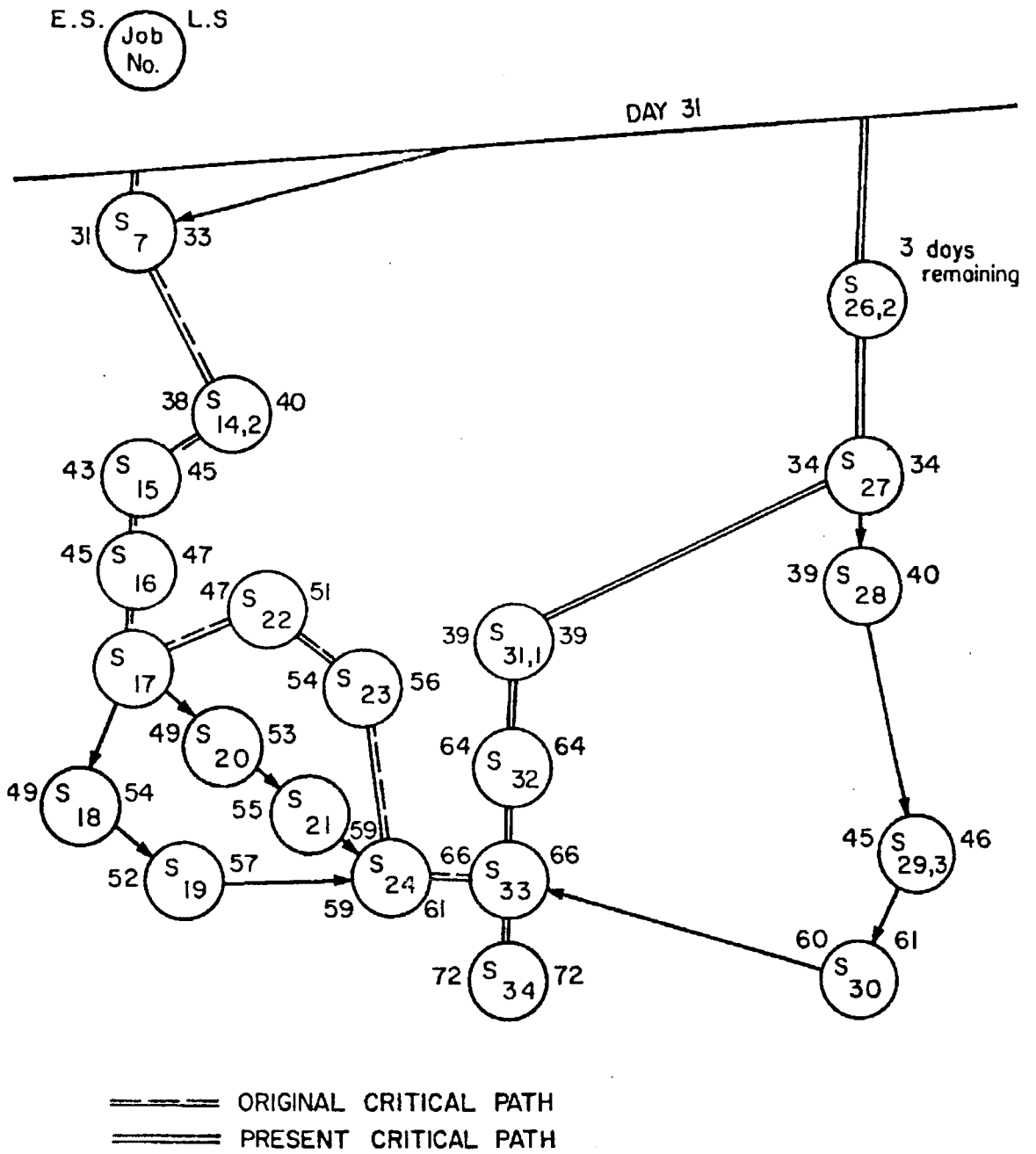


Fig 4.8 Effect of incremental job cost on completion cost

Also, given that the length of the project is long compared to the length of the longest job in a multi job set, and that the jobs are ordered by early start time it is only necessary to examine a small fraction of the jobs in the search for parallel jobs.

4.6 Application of Decision CPM to dynamic monitoring and control of projects

In the previous sections Decision CPM has been presented as a technique to solve combined planning and scheduling problems for construction type projects, as originally stated. However it is clear that such things as project crashing can be handled by the technique if the assumption is made that there are several discrete levels of performance possible, rather than a continuous linear relation between job cost and job time. The assumption of discreteness is particularly fitting in construction projects where there is a choice such as one or two-shift operation and the use of regular or quick drying cement, etc.

Further more the potential of the technique for project design and job crashing has important implications for the dynamic monitoring control of the project as we carry out the plan. Suppose that daily information is collected concerning the status of jobs in a project being worked on. At any given time some jobs would have been finished on time or early and others will have been delayed. Given this information it is possible to solve again for a new optimum in the decision project graph of the remaining problem. It is true that some earlier decision may be unchangeable; perhaps because of the purchase of material, but some decision sets and many possibilities for crashing of jobs will usually remain. Given the actual status of the project, new decisions may now be better than those previously accepted.

An example of such a situation is given in Fig. 4.7, which shows the state of the project graph of Fig 4.5 at beginning of day 31. According to the original schedule jobs

S_7 and S_{27} should have been started. However $S_{26,2}$, an immediate predecessor of S_{27} with a job time of five days, has been in progress only two days.

An examination of the original schedules showed that S_{27} had one day of slack. Therefore the three-day delay will lengthen the completion date by two days and incur a penalty of 900 thousand. If the graph is solved with the new information a switch from $S_{31,1}$ to $S_{31,2}$ is shown to be desirable. For an incremental job cost of 130 thousand one day can be removed from the completion cost (Fig 4.8). The total cost of the delay is therefore reduced to $900-450+130 = 580$ thousands. Thus specifying alternatives, as is done in the decision project graph, allows continual adjustment of decision in response to operating experience.

In summary, the dynamic design and control processes will operate as follows.

(1) Collect information on

- (a) tasks to be performed and alternative methods of performing them (jobs),
- (b) technological relations between jobs,
- (c) interdependences among decisions,
- (d) job times,
- (e) job costs,
- (f) job crashing methods and costs,
- (g) project due date, penalty, and premium.

(2) Solve problems using Decision CPM for original plan.

(3) Begin project.

(4) At regular intervals collect information on project progress.

- (5) Update job cost and job time data.
- (6) Replace decision nodes by single jobs as decisions become irrevocable over time.
- (7) At regular intervals resolve the remaining decision project graph to see if savings are possible by the implementation of a new plan. Go to 4
- (8) Halt when project is complete.

PROGRAM EVALUATION AND REVIEW TECHNIQUE (PERT)

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 5

PROGRAM EVALUATION AND REVIEW TECHNIQUE (PERT)

5.1 Basic PERT theory

The PERT model utilizes 3 durations for an activity viz., optimistic, most likely and pessimistic. The expected duration of an activity t_e and the standard deviation σ are defined as

$$t_e = \frac{t_o + 4 t_m + t_p}{6} \quad (5.1)$$

and

$$\sigma = \frac{(t_p - t_o)}{d} \quad (5.2)$$

where t_o = optimistic duration
 t_m = most likely duration
 t_p = pessimistic duration
 σ = standard deviation
and d = scaling factor.

This is conveniently taken as

$$\sigma^2 = (1/36) (t_p - t_o)^2 \quad (5.3)$$

Subject to the assumption that the density distribution of the activity time is

$$F(t) = \frac{\Gamma(\alpha + \beta) (t - t_o)^{\alpha-1} (t_p - t)^{\beta-1}}{\Gamma(\alpha) \Gamma(\beta) (t_p - t_o)^{\alpha+\beta-1}}, \quad t_o < t < t_p, \alpha, \beta > 0. \quad (5.4)$$

Though a scaling factor of 6 is commonly used, a scaling factor of 3.2 is most suitable for the statistical distribution as applicable to construction projects (Dimitri. G. G., 1988, 1989; Dodin. B., 1985; Keefer. D. L. and Verdini. W. A, 1993; Kulkarni. V. G, 1984, 1986; Monhor. D, 1983; Sculli. D, 1989). PERT uses the central limit theorem (CLT) to find the expected project duration. The central limit theorem indicates that for independent random variables.

$$E(t) = t_1 + t_2 + t_3 + \dots + t_n \quad (5.5)$$

$$\text{and } S^2 = \sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \dots + \sigma_n^2 \quad (5.6)$$

where

$E(t)$	=	expected duration
t_i	=	expected duration of i^{th} activity
S	=	standard deviation of the project and
σ_i	=	standard deviation of the i^{th} activity

PERT assumes that the network $F(t)$ follows a normal distribution with expected duration $E(t)$ and standard deviation S . Values for $F(t)$ can be taken from the standardized normal distribution table. $F(t)$ of the network is determined exclusively by the expected duration and the standard deviation of the critical path. If this is programmed, the scheduler has to find the probability $S(t)$ that the network duration will be longer than a duration t .

$$F(t) = \frac{1}{1 + \exp[-1.5957691(X)(1 + 0.444715(X^2))]} \quad (5.7)$$

$$\begin{aligned} \text{where} \quad S(t) &= 1 - F(t) \\ \text{and} \quad X &= [E(t) - t] / S \end{aligned}$$

$S(t)$ can be calculated for as much durations, t , as the scheduler feels necessary.

5.2 Probabilistic PERT

PERT networks are referred to by some researchers as “probabilistic activity networks” (PAN) because the duration of some or all of the arcs are independent random variables with known probability distribution functions, and have finite ranges.

PERT networks are frequently used as models for the management and control of real life projects in many areas such as research and development, production, construction, maintenance, and others. In the PERT network of a project, the arcs represent the precedence among the activities.

One of the main purposes of using network analysis for project planning and control stems from the need to identify the paths and activities that are critical to the achievement of the project objectives. In deterministic activity networks (DANs), in which the activity duration is a constant, it is relatively easy to answer questions such as: which is the critical path(s)? Which are the most the critical activities? Which is the critical list (CL) of activities?

In PANs, such as the PERT model, one must phrase these questions in probabilistic terms, such as: which path (or paths) is (are) the most probable to be critical? Which activities are the most probable to be critical? Which activities are the most critical activities? Can the activities be ranked in decreasing order of their relative criticalities?

5.3 Criticality indices for the activities in PERT networks

A stochastic PERT network is a directed, connected, acyclic graph, $G(N,A)$, composed of N nodes and A arcs (in which the symbol indicates the set and its cardinality). The network must have only one starting and one ending node; dummy activities can be used if necessary to insure that. The node can be numbered ascending so that an arrow leads from a smaller to a larger numbered node. Henceforth, take start node to be node 1 and the terminal node to be node N .

If we let P denote the set of all paths in the PERT network, and $Z(\pi_l)$ denote the duration of path $\pi_l \in P$, then:

$$Z(\pi_l) = \sum_{(ij) \in \pi_l} Y_{ij}$$

where Y_{ij} is the duration of arc $(ij) \in A$. We measure the criticality of a path $\pi_l \in P$ by the probability that its duration $Z(\pi_l)$ is greater than or equal to the duration of any other path.

This probability is called the criticality index of the path, and is denoted by CP . Therefore, for any path $\pi_l \in P$:

$$CP(\pi_l) = Pr [Z(\pi_l) \geq Z(\pi_q) \text{ for all } \pi_q \in P; \pi_l \neq \pi_q] \quad (5.8)$$

The criticality index of an activity, denoted by CA , is defined by the sum of the criticality indices of the paths containing it; therefore, for any activity $(ij) \in A$;

$$CA(ij) = \sum_{\pi_l \text{ ((ij)} \in \pi_l)} CP(\pi_l) \quad (5.9)$$

Evidently, the larger the value of $CA(ij)$ the more critical is the activity; and conversely. The CA appears to be an exceedingly useful measure of the degree of attention an activity should receive by management, since it carries more pointed information than the critical path concept now used. Specifically, the CA s indicate which activities are the bottleneck activities and should be expedited if the entire project is to be expedited. The CA is the basis for project scheduling, resource leveling, project compression, and project monitoring.

The evaluation of (5.8) and (5.9) above may be achieved by complete enumeration, which is practical only for small size ANs, or can be estimated by the using general Monte Carlo Simulation (MCS). In fact, the issue of determining the CP s and the CA s has received little attention in the literature. This is due to the difficulty in evaluating both expressions (5.8) and (5.9). The evaluation of the CA s requires the enumeration of all the paths in the AN, the approximation of the corresponding CP s, and the identification of the paths passing through each activity, all of which are burdensome, time consuming tasks. Theoretical results are developed which lead to the first analytical approximating procedure to estimate the CA s with out either using MCS or identifying the paths and their CP s. Such a procedure is a direct application of the approximation of the distribution function (df) of the project completion time.

The procedure starts at node N , approximate the CA s of the arcs ending in node N , then proceeds recursively to nodes $N - 1$, $N - 2$ and so on until it finally reaches node 2 where the CA of arc (1,2) is estimated.

MONTE CARLO SIMULATION TECHNIQUE

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 6

MONTE CARLO SIMULATION TECHNIQUE

6.1 Introduction

Monte Carlo Simulation method is a probabilistic method, which uses randomness in its calculations. This method is suitable for computer applications only, since it requires a large number of calculations. The model involves a computer simulation procedure and before running a simulation of the duration of a construction network, the cumulative density function (CDF) of each activity is determined. During each replication in the simulation, random values in the range 0 to 1 are assigned to the probability of completion of the activities. Once the CDF and the probability of completion of the activities are known, their durations can be determined by solving equation (5.5) for the given random probability. Thereafter, the duration of each path can be found by summing up the duration of all activities in the path. The whole process is repeated as many times as necessary. A large number of replications, say 10,000 is needed to obtain very accurate results. A simulation with 1000 replications gives only satisfactory results for construction networking purposes.

To perform an MCS, the activities in the network are assumed to be independent. This means that the duration of any activity will not affect the duration of another. Triangular approximations for the distributions as shown in Figs. 6.1 (a) and 6.1(b) can be suitably used for construction activities. The optimistic, most likely and pessimistic durations as used in PERT can be determined the same way in MCS also, since these values are independent of the method used.

After the desired number of replications has been run, the network durations are ranked in the order of the shortest to the largest duration. Since the MCS result is based on a number of replications, equation (6.1) gives a good approximation for the probability, P , that the project will be completed in time, t , or longer.

$$P = 1 - (n / N) \quad (6.1)$$

where n = number of replications with project duration equal or smaller than T
and N = total number of replications

A large number of values for P for different values of t will guarantee a smooth survival function curve.

In this method, every cost component with high potential for variability is modeled as a random variable. A computer program is used to generate random numbers based on the assumed statistical distributions for various cost components. The generated numbers and the constant cost

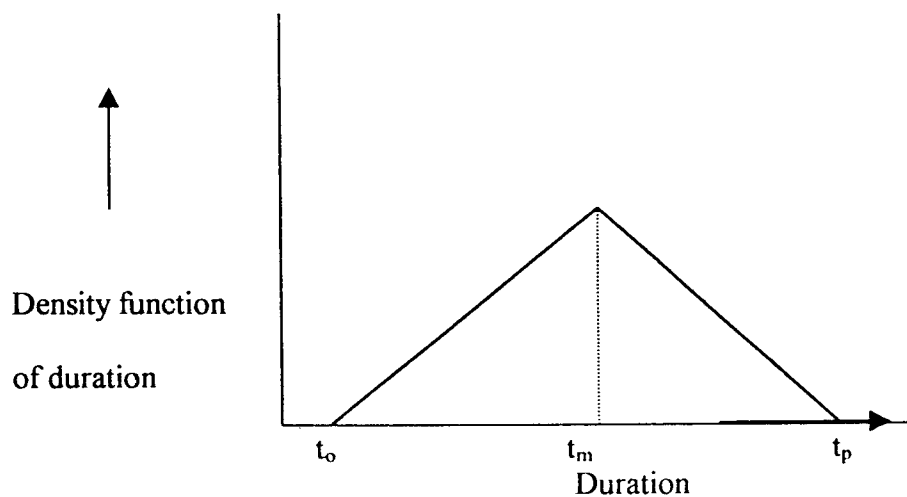


Fig. 6.1(a) Symmetric distribution

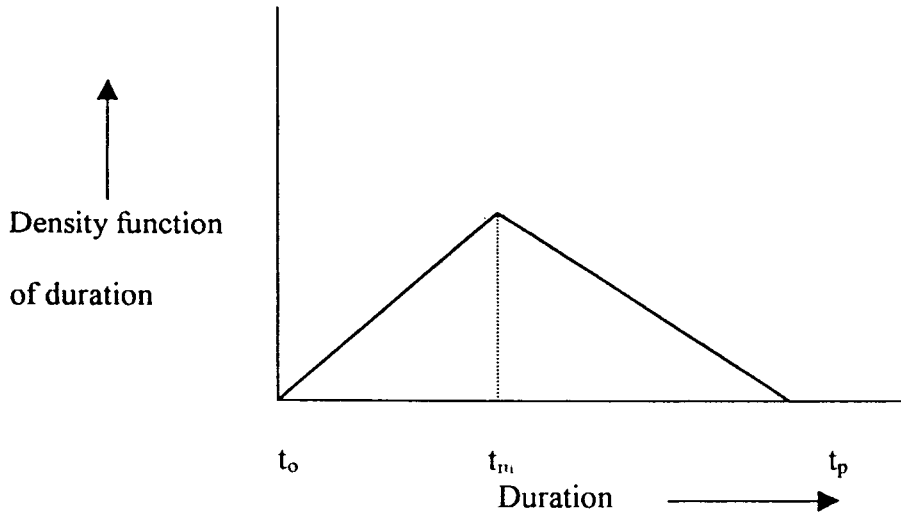


Fig. 6.1 (b) Asymmetric distribution

figures (those cost items believed to be estimated fairly accurately and not expected to show large variations) are added up, and value for the total cost is computed. This procedure is repeated hundreds of times, and cumulative distribution is obtained for the total project cost. This cumulative distribution is then used to compute the probability of finishing the project at or below various costs.

The Monte Carlo technique seems easy to use because it does not require the user to compute the total project cost analytically. As long as the user has reasonably accurate data for various cost components and has access to the necessary software, he or she can get a risk profile for the total project cost. This cumulative distribution is obtained for the total project cost. The problem is that first some reasonable statistical distribution should be specified for any cost component subject to variation. Second, cost correlations between individual components should be known and accounted for. This

second requirement is the one usually neglected partly because it is difficult to measure correlations.

6.2 Monte Carlo approach

In probabilistic estimating, it is understood that because of the inherent random nature of costs, the total project cost should be modeled as a random number rather than the fixed sum. Each project cost component can be represented by the suitable distribution or at least with a mean and variance. The total project cost will be a random variable that is the sum of several random numbers.

$$C_{\text{tot}} = \sum_{i=1}^n C_i \quad (6.2)$$

where C_{tot} = the total project cost and C_i = various project cost components.

Obviously, If one want to consider cost variation in every small cost component that goes into a detailed estimate, the approach would be impartial. Because of this, the considered C_i includes major items that generally appear on the estimate summary sheets and the recap sheets. Also, it is understood that most of the total cost variation is due to the variability of a limited number of components. So only those items with high potential for variations are considered to be random variables, and the rest of the items are assumed to be fixed. C_{tot} in (6.2) is then composed of one fixed and one random component. As various C_i can have various distributions, accurate computation of the

probability distribution of C_{tot} involves computing of a number of convolution integrals and becomes very lengthy.

Monte Carlo simulation can simplify the process if a computer and relevant software are available. It consists of generating random numbers according to C_i distributions, adding these items, and adding the fixed costs, thereby computing the total project cost. This procedure is repeated for several hundreds to several thousands times and every time a value for C_{tot} is computed. A histogram and later a cumulative distribution function (CDF) can be constructed with the value of C_{tot} . The CDF can then be used to estimate the probability of computing a project at or below a certain budget.

6.3 Problems with Monte Carlo approach

Although the Monte Carlo approach provides a straightforward means of probabilistic estimating, there are major limitations in its applications. First, one needs to establish statistical distribution for various cost components. Second, if the random numbers are not independent, their correlations should be accounted for in the correct application of the Monte Carlo technique.

6.4 Statistical distribution of cost items

Specifying parameters of the distribution such as mean, variance, and even the range is generally not sufficient for generating a random number. Specification of the

underlying distributions for various cost components is a necessary ingredient of the Monte Carlo technique. Normal, lognormal, triangular, beta and uniform distributions have been suggested by various authors (A. Touran and E.P. Wiser, 1992; F. M. Martin, 1968; F. J. Maryanski, 1980; J. Reitman, 1971).

One general approach when there are insufficient data regarding the mode of the distribution, or where the range is relatively small, has been to assume a uniform distribution. When some information regarding the most likely cost value is available, the next simplest distribution would be triangular distribution. It has also been argued that individual cost components (cost components that are not the sum of a large number of sub components) are unimodal and skewed. In addition the cost values have to be positive numbers. Because of these assumptions, the beta family of the distributions and the lognormal distribution has been suggested to represent cost variations. The beta distribution is particularly attractive because it is confined between two definite limits. Also it is used extensively in the PERT. Because of this, a good number of cost estimators are familiar with the method of three times estimates for modeling the duration distribution. One logical method for investigating the distribution type is to collect data from similar projects, assume a distribution, and perform a proper test of goodness of fit to evaluate the hypothesis.

6.5 Correlation between project cost components

One of the more common sources of error in Monte Carlo simulation is that cost components are assumed to be independent, so change in one cost component does not affect any other component. This is clearly inaccurate in typical construction project. However, if the correlation between variables is sufficiently small, the assumption of independence does not create large errors. Generally, disregarding the correlation between variables in a Monte Carlo simulation results in an underestimation of total cost variance. It is important to emphasize that an accurate estimate of cost variance and even cost distribution is all important in probabilistic estimating because the cost estimator is interested in deviations from the mean estimate for computing the probability of cost overrun or under-run.

6.6 Approximate method for incorporating correlation

Exact methods of incorporating correlations are time-consuming and require a great deal of data that are not always available. In some cases, depending on the underlying distributions, it is not possible to make an exact analysis. For less detailed studies, one suggested method involves combining highly correlated cost items into a single cost item such that all remaining items (some of which may be combination of several other correlated cost items) can be considered independent. For example, assume that a project cost consists of 10 cost items C_1 to C_{10} . So we have

$$C_{1\alpha} = \sum_{n=1}^{10} C_i \quad (6.3)$$

Further, assume that we have reason to believe that C_4 , C_5 , and C_6 are highly correlated and that C_9 is correlated with C_{10} . Define C' and C'' such that:

$$C' = C_4 + C_5 + C_6 \quad (6.4)$$

$$C'' = C_9 + C_{10} \quad (6.5)$$

If the estimator can specify underlying distribution and parameters of C' and C'' and if the rest of the cost components can be assumed to be independent, then by rewriting (6.2) as the following, a Monte Carlo simulation can be conducted;

$$C_{\alpha} = C_1 + C_2 + C_3 + C' + C'' \quad (6.6)$$

6.7 Exact methods for incorporating correlations in simulation

To conduct an exact simulation analysis for the total cost distribution when cost components are correlated, one needs the joint density function of the correlated cost components. Without the joint density function, properly correlated random numbers cannot be generated for Monte Carlo simulation. The probability density function (PDF) that the cost estimator or risk analyst specifies for a certain cost component is actually the marginal distribution of that component. The marginal distribution of a cost component gives the probability distribution of that component without reference to other correlated components.

When cost components are independent, the product of their marginal distributions gives their joint density function. If different cost components are not

independent, knowing the marginal of this random variable is not sufficient to obtain their joint density function. The case of the multivariate normal distribution is special in that the only information needed besides the marginal distribution of each random variable is the values of covariance between the variables. If the various cost components in the construction project are assumed to follow a normal distribution, one can use a multivariate normal distribution for generating correlated cost components, given that covariance between variables are known. The multivariate normal assumption implies that the individual cost components are normally distributed. Multivariate normal distribution can be transformed into multivariate lognormal. Also in special case one can use approximations to analyze the correlated random variates at the cost of reduced accuracy.

6.8 Multivariate normal distribution

The n-dimensional multivariate normal distribution with mean $\mu = (\mu_1, \mu_2, \dots, \mu_n)^T$ (T denotes transpose of a vector) and covariance matrix Z, where σ_{ij} (the covariance between variable i and variable j) is the (i, j)th entry in the matrix, has the joint density function as given by

$$f_c(c) = (2\pi)^{-n} \cdot 2 |Z|^{-0.5} \exp \left[- (c - \mu)^T Z^{-1} (c - \mu) / 2 \right] \quad (6.7)$$

where $c = (c_1, c_2, \dots)^T$ are random variables (cost components); and $|Z|$ is the determinant of Z. Note that $C = (C_1, C_2, \dots)^T$ is normal (μ, Z) , $\text{cov}(C_1, C_2) = \sigma_{ij}$, and Z is symmetric (i.e., $\sigma_{ij} = \sigma_{ji}$) and is positive definite. Lognormal distribution is a good choice of distribution for representing construction costs for commercial buildings.

As Z is the covariance matrix, it has to be symmetric and positive definite. And it can be written as $Z = XX^T$, where X is an $n \times n$ lower triangular matrix. C can be represented as

$$C = X \cdot N + \mu \quad (6.8)$$

Where $N = (N_1, N_2, \dots, N_n)^T$, where N_i is an independently identically normally distributed random variable with mean 0 and variance 1. C_i can be generated from (6.8).

For every cost component C_i , one can write;

$$C_i = \mu_i + \sum_{j=1}^n X_{ij} \cdot N_j \quad (6.9)$$

So the X matrix is computed first. Then a number of independently and identically distributed normal variates with means 0 and variances 1 are generated (N_i); then by using (6.9), C_i can be computed. The described procedure assumes that data are normally distributed.

6.9 Multivariable lognormal Distribution

If we assume that the cost data are log normally distributed; the data are first transformed to normal distributions using

$$\mu_i = \ln \left(\mu_i^2 / \sqrt{\sigma_i^2 + \mu_i^2} \right) \quad (6.10)$$

$$\sigma_i^2 = \ln \left(1 - \sigma_i^2 / \mu_i^2 \right) \quad (6.11)$$

$$r_{ij} = (1 / \sigma_i \sigma_j) \ln \left(1 + r_{ij} \cdot \left| (\sigma_i^2 \sigma_j^2 / \mu_i^2 \mu_j^2) \right| \right) \quad (6.12)$$

where μ_i , σ_i^2 and r_{ij} = means, variances and correlation coefficients, respectively, of the multivariate normal distribution; and μ'_i , $\sigma_i'^2$ and r'_{ij} = means, variances and correlation coefficients, respectively, of the multivariate lognormal distribution. After the cost data are transformed to normal, compute matrix X , and (6.9) is used to generate random variable C_i .

If there were three correlated random variables C_4 , C_5 and C_6 , instead of modeling their sum C' , an exact method for incorporating correlation in the analysis were to be used. One would consider the joint density function of these three variables and generate C_4 , C_5 and C_6 as a function of their means, variances and covariance. By considering covariance between random variables, the impact of correlations on the total can be evaluated by using the procedure already described. This can be done if C_4 , C_5 and C_6 can be assumed to be normally or log-normally distributed. The same procedure would be followed regarding C_9 and C_{10} . C_{tot} can be computed by adding up C_i in every simulation run. By repeating the process several times a CDF for C_{tot} can be obtained. If the data cannot be assumed normally or log-normally distributed, the process of generating correlated random variables becomes more complicated and availability of a methodology is not guaranteed. However, approximation usually can be used to arrive at acceptable results.

6.10 Simplified Monte Carlo Simulation (SMCS) method

From the MCS method, SMCS simplifies the scheduling network to those activities and paths that are more likely to cause delay of the project completion. Though the SMCS method is similar to the MCS method, the elimination of path(s) and activities in the network is performed prior to the first replication. Each replication involves the calculation of the duration of each activity and each path in the simplified network. The project duration for a particular replication is the duration of the longest path. Although with the SMCS method the number of calculations is considerably reduced, the use of a computer is recommended.

The first step in SMCS is the calculation of the expected duration of each activity. Equation (5.5) can be used to calculate the expected duration $E(t)$ of the network. Those paths with an expected duration of less than t_{min} are not considered in further calculations. Equation (6.13) gives the formula to determine t_{min}

$$t_{min} = K \times E(t) \quad (6.13)$$

where K = a coefficient that indicates how close a path must be to the critical path if it is to cause the delay of the project, and can range from zero to one. The selection of K for a particular network is left to the scheduler's judgment. An approximation of K as $2/3$ result in considerable reduction of the computing time and unnoticeable changes in accuracy compared to MCS.

After t_{min} is used to filter the network paths, the network activities can be reduced. Those activities that are absent from any of the remaining paths need not be considered in the rest of the calculations. The result of these two refinements is a simplified network, which only includes activities that are likely to cause delay of the project. The simulation process and the procedures used to develop the network survival function curve are similar to those of the MCS method. The only difference in the SMCS method is that the calculations are performed using the simplified network.

LINE OF BALANCE (LOB) SCHEDULING TECHNIQUE

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 7

LINE OF BALANCE (LOB) SCHEDULING TECHNIQUE

Methods for planning and controlling highly repetitive projects have been investigated in the last two decades. The techniques that were developed are grouped under the generic term of "Linear Scheduling Methods". Their origins are not clear; there may actually have been multiple origins, possible in different countries. They have been originally devised to solve industrial production problems and their considerations for use in the construction industries is rather a recent event. They include a multitude of variations that are based on the same resource oriented principles. They are named differently: Line of Balance Schedules, Vertical Production Method (VPM), Time-Space Scheduling, Cascade Networks, Velocity Diagrams, Fenced Bar-Charts, chain Bar-Charts, Construction Management Systems, and combined PERT/ LOB; but they have common features. Some others make use of process interaction simulation techniques, stochastic approaches, and dynamic or linear programming.

The literature indicates (Dumond. J and V. A. Mabert, 1988; Edmund. N. C. Jr. and D. W. Johnston, 1986; Handa. V. K. and Barcia. R. M, 1986; Li. K. Y. and Wills. R. J., 1992; Mario. I. N. and M. G. Smith, 1986) that linear scheduling methods could be used in projects such as multiple housing schemes, bridge, high-rise and low-rise buildings, pipelines and highway constructions. The general consensus is that linear scheduling methods are well suited to projects that are composed of activities of repetitive nature. But, their use is believed by some to have faded into obscurity with the advent of network techniques.

In order to meet the objectives of scheduling, a network diagram for one of the many units to be produced is prepared as a first step. Then, the man-hours necessary, as well as the optimum crew sizes are estimated for each activity. This information yields a natural rhythm for each activity (i.e., number of units/day) defined as the optimum rate of output that differ from a multiple of the natural rhythm is bound to yield some idle time for labor and equipment. That is why the number of crew necessary for the entire project is so arranged that the rate of output, a multiple of the natural rhythm, is as close to the target rate as possible.

Once the number of crews, and the actual rate of output have been computed for each activity, the LOB diagram can be drawn. The number of units to be produced is plotted against time. Two oblique and parallel lines, whose slope is equal to the actual rate of the output, will denote the start and finish time respectively of each activity in all the units, from first to last. An example of an activity that has duration of 0.5 days and a natural rhythm of 2 units of production per day is given in Fig. 7.1. In first case only one crew of optimum size is used and an actual rate of the production of 4 units per day is achieved. The vertical arrows show the movement of the crews in either of the cases. The actual rate of production is the slope of the line of balance joining the start times of the repetitive activity in each unit, and is calculated as

$$M = \frac{Q_i - Q_j}{t_i - t_j} ; \quad i < j \quad (7.1)$$

where M = rate of production (units of production per units of time); Q_i, Q_j = number of units started (i and j) ; and t_i, t_j = time elapsed between the start of the project and the start of the i th and j th units, respectively.

The slope of the line of balance joining the finish time of the repetitive activity in each unit is also equal to m . If the duration of the activity is known and if the actual rate of output is limited to a multiple of the natural rhythm, then the foregoing equation is effectively reduced to

$$m = \frac{p}{d} \quad (7.2)$$

where p = number of crews used in the activity and d = duration of the activity in one unit

The time ordinate t of the start of an activity is calculated by the following relationship

$$t_i = t_1 + (1/m)(Q_i - 1) \quad (7.3)$$

where t_i = time ordinate of start of activity in the i th unit; t_1 = time ordinate of start of activity in first unit; m = rate of production; and Q_i = number of units produced, i . The value of t_1 , the start time of the activity in the first unit can be obtained from the time calculation performed for the unit network. The finish time of the activity in the i th unit can be calculated by adding the duration of the activity (d) to its start time on unit i (t_i).

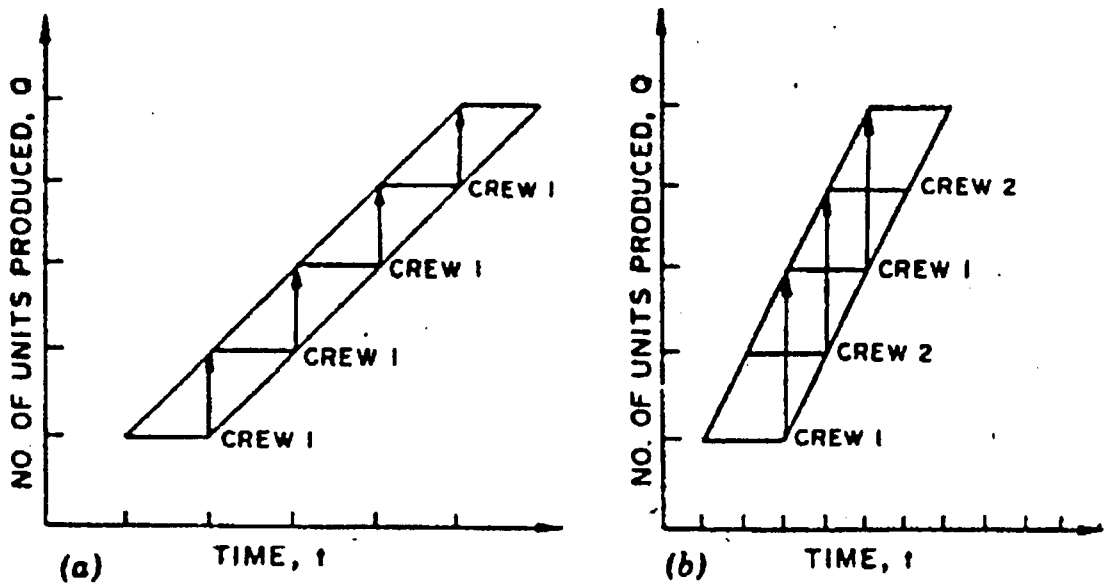


Fig.7.1. Line-of-Balance diagram for activity (activity duration = 0.5 days; natural rhythm = 2 units/day)

Fig 7.1.(a). Number of crew used = 1; actual rate of production = 2 units/day

Fig 7.1.(b). Number of crew used = 2; actual rate of production = 4 units/day

PETRI NETS IN SCHEDULING

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 8

PETRI NETS IN SCHEDULING

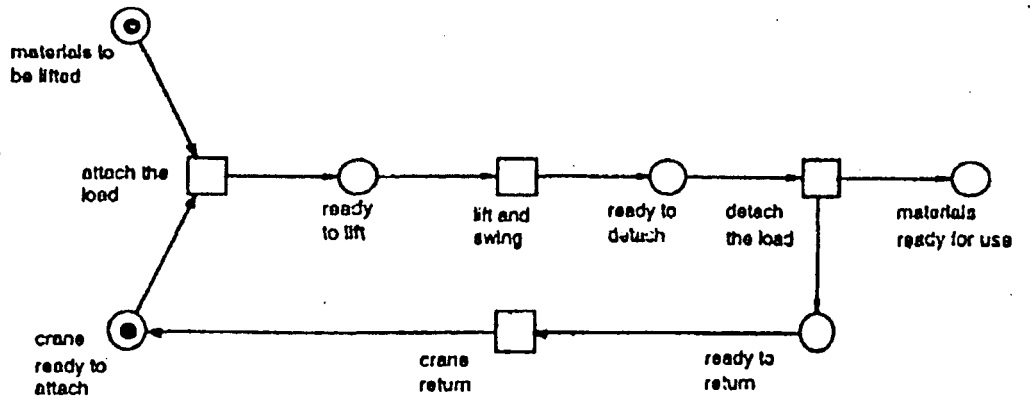
8.1 Introduction

Petri nets (PNs), are a formal graphical modeling tool to model communication systems, computer software and hardware, manufacturing, safety analysis, queuing system and can also be used effectively for modeling construction systems. When compared with other modeling schemes, such as queuing theory and discrete events simulation, PNs are generally considered to be superior for system with concurrency, where several state changes happen simultaneously and where events-driven characteristics are present. Both of these phenomena are present in construction systems. PNs can be converted into algebraic representation that can be analyzed to ensure that logic requirements are met.

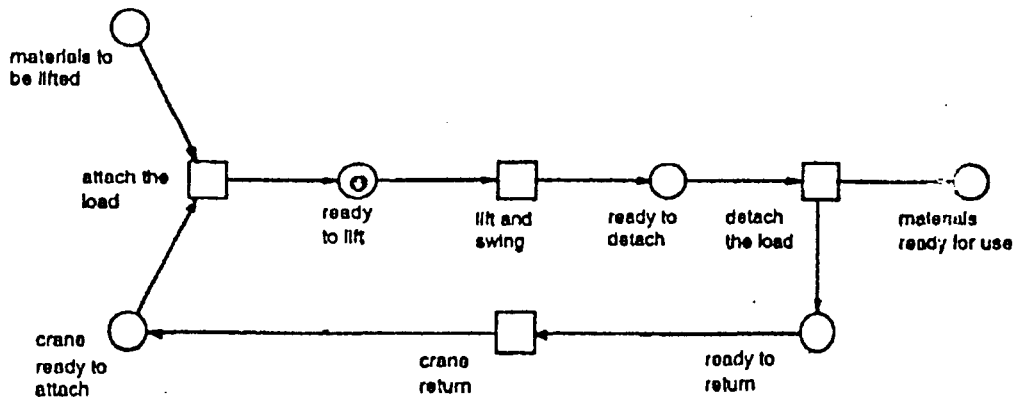
8.2 Petri net approach

C.A. Petri introduced his network in 1962. Since then many advances have been developed by simulation researchers for general simulation applications. Fig.8.1 (a) is a graphical PN model of crane-hoisting materials from the ground to the work face. This network is made up of a number of connected symbols, each with the following attributes:

1. Circle (places) represents states of being. In construction, these are often states of readiness. The crane moves through the following states of readiness: crane ready to attach, ready to lift, ready to detach, and ready to return.
2. Squares (transitions) are actions that change the state of the system. Attach the load, lift and swing, detach the load, and crane return are transitions.



(a)



(b)

Key

○ Places - states of readiness

□ Transitions - actions that change the state of the system

● Tokens - equipment, resources, information

———— Directed Arcs - indicates the direction tokens move

Fig 8.1 PN model of crane hoisting materials

3. Black dots (tokens) are the resources of the operation. In this case the tokens are the materials being hoisted and the crane doing the hoisting.
4. Arrows (directed arcs) indicates the directions the resources (tokens) move when an action (transition) takes place. In Petri net jargon, transitions are said to “fire” as their action takes place.

The location of the materials and crane (tokens) in the network at any point in time is referred to as the “marking” of the net at that instant. Let us start an example with two tokens, the crane and the materials to be lifted, both in their respective circles (places). At this point the transition “attach the load” is enabled and ready to fire. The firing of the transition moves the token from “crane ready” and “material to be lifted” (the input places), and puts a single token in the place “ready to lift”(the output place). The result of this first transition firing is shown in Fig. 8.1(b). Transitions continue to fire when enabled until the supply of tokens is exhausted.

8.3 Mathematical definition

A Petri net, in its most basic form, can be defined as:

$$PN = \langle P, T, IN, OUT \rangle \text{ where} \quad (8.1)$$

$$P = \{p_1, p_2, p_3, \dots, p_n\} \text{ the set of } n \text{ places} \quad (8.2)$$

$$T = \{t_1, t_2, t_3, \dots, t_m\} \text{ the set of } m \text{ transitions} \quad (8.3)$$

$$IN = (P \times T) \rightarrow N \text{ is an input function that defines directed arcs from places to transitions.} \quad (8.4)$$

$OUT = (P \times T) \rightarrow N$ is an output function of directed arcs from transitions to places. (8.5)

The directed arcs can be weighted to indicate the number of tokens that are transferred to or from a place on the firing of a transition.

8.4 Properties of Petri nets

PNs exhibit a number of important properties that are useful in determining the behavior of the modeled system. These properties may also prove useful in analyzing construction systems at the planning stage. A list of the properties, along with the descriptions, are given as follows

1. *Safeness* – a place is safe if the number of tokens in that place never exceeds one. A PN is safe if all its places are safe. The crane places in the forgoing example must be safe as there is only one crane. Safeness can be used to check the logic of the net.

2. *Boundedness* – a place is k -bounded if the number of tokens in it cannot exceed k . For example, if a storage area on site can only store three pre cast units, it would be necessary for the place representing the storage area to be three bounded.

3. *Conservativeness* – a net is strictly conservative if the number of tokens within it remains constant. This concept is useful when tokens represent resources. Some times, however, tokens represent counters or other non-resource items. Under these circumstances it might not matter if tokens are created or destroyed. To provide the

benefit of conservativeness without this problem, one can consider the net conservative only for tokens representing resources. This problem can also be overcome by using colored tokens.

4. *Liveness* - a transition is live if it is able to fire and is not dead locked. If a transition is live it is not necessarily enabled, but is potentially enabled under some attainable marking. This property can be used as a check on the logic of the system. If all transitions are live, given the correct conditions, they can fire and the stage change represented by the transition can occur.

5. *Reachability*- under the marking μ , if it is possible to obtain the marking μ' , μ' is said to be reachable. In Fig.8.1, the fact that marking of one token in "ready to lift" is reachable from the initial marking means the logic of the transition is correct.

These properties are not all applicable to all forms of PNs. Basic PNs are useful in investigating qualitative or logical properties of dynamic systems, such as *boundedness* and *liveness*.

8.5 Petri net constructs useful in project modeling

Several PN constructs that represent particular characteristics of activities are illustrated in Fig. 8.2. as follows

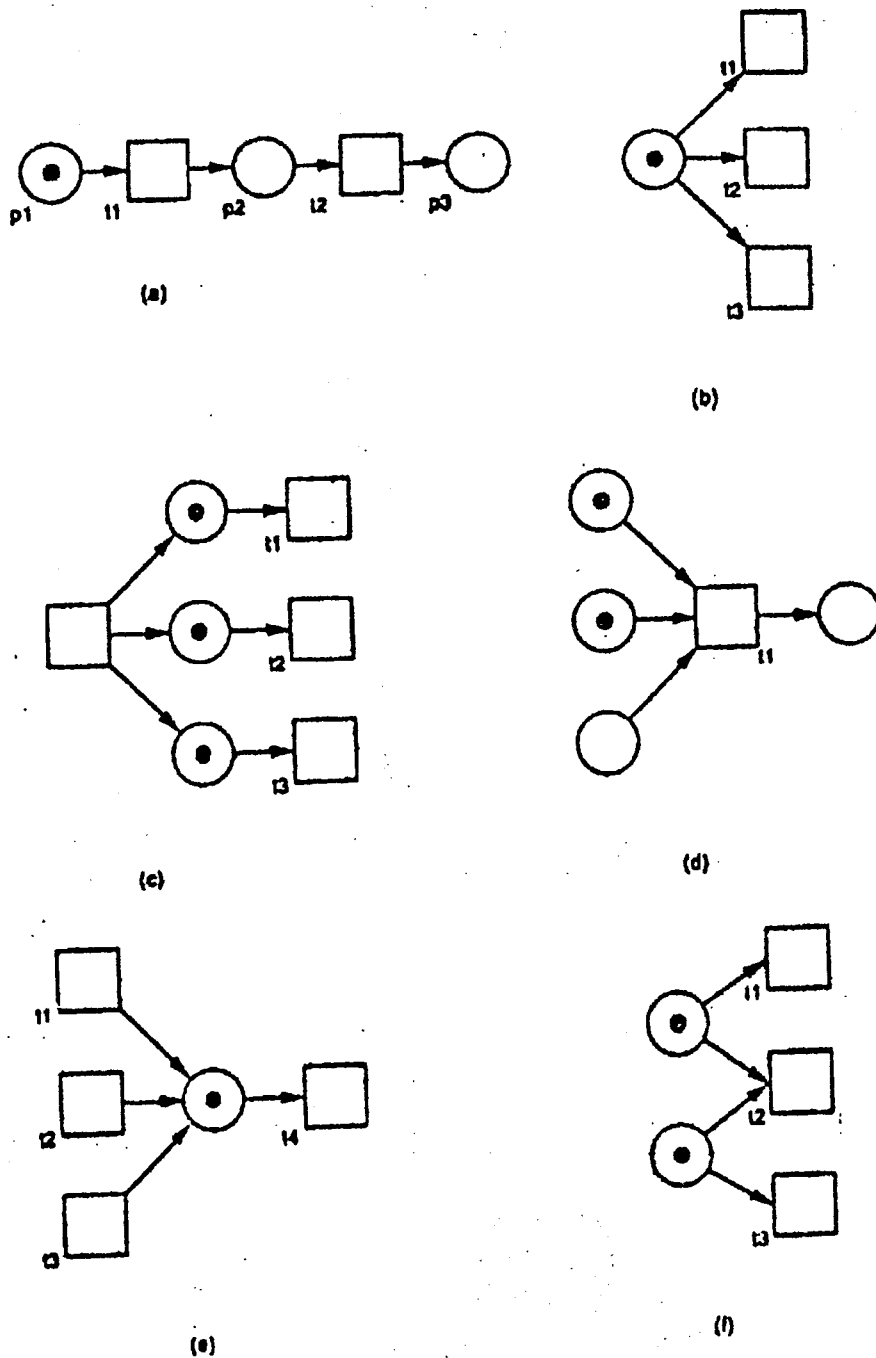


Fig 8.2. Typical PN constructs useful in construction project scheduling

(a) Sequential execution
(b) Conflict

(c) Concurrency
(d) Synchronization

(e) Merging
(f) Confusion

Fig. 8.2(a). *Sequential execution*-transition t_2 can fire only after the firing of t_1 . This imposes a precedence constraint that is typical of some construction activities. The construct can also be used to model causal relationships between activities. For example, concrete cannot be placed until the reinforcement is fixed and the framework is completed.

Fig. 8.2(b). *Conflict*-transitions t_1 , t_2 , and t_3 are in conflict. All are enabled but firing of any leads to the others being disabled. Such situations arise when a resource is being shared between a number of activities. The conflict can be resolved by assigning probabilities or priorities to the transitions. An example of such a conflict occurs when a single tower crane is needed to carry out two separate tasks simultaneously. Performance of one-task stops the other task being undertaken at that time.

Fig. 8.2 (c). *Concurrency*-transitions t_1 , t_2 , and t_3 are concurrent. This construct allows us to represent construction activities that proceed in parallel.

Fig. 8.2(d). *Synchronization*-when several resources and pieces of equipments need to be available for an activity to proceed; this type of construct can be used. The activity can proceed only when a token arrives in the place without a token, thereby synchronizing the fire of t_j to the token arrival. An example of synchronization occurs in Fig. 8.1 where the places "material to be lifted" and "crane ready to attach" each have a token and therefore enable the transition "attach the load" to occur.

Fig. 8.2(e). *Merging*-this structure arises when several materials arrive for use at the same activity. For example, the constituent materials of concrete arrive at the concrete mixer.

Fig.8.2(f). *Confusion*-this is the situation when conflict and concurrency exist together. In a project this some times occurs when resources are shared. It can be resolved by associating priorities or probabilities with the relevant transitions.

It is possible to build a logic model of most project systems using a combination of these basic PN constructs. However, for quantitative modeling and analysis of construction system the concepts of time and decision branching need to be added to basic PNs.

8.6 Time Petri nets (TPN)

To effectively model project systems, we need to be able to associate times with transitions (activities). This means that transition will, when all the input conditions are met, take t time units for the firing process to deposit tokens in the output places. The time delay can be deterministic or represented by any distribution function. It is also possible to associate times with other PN elements, but here we will consider time to be associated only with transitions.

8.7 Extended transitions

Extended transitions allow the modeler to put decision branches into the net. When token pass an extended transition, a decision must be made according to a given condition within the transition. The extensions to basic transitions that we use in construction modeling are probabilistic transitions and priority transitions. These extensions are necessary when two or more transitions are in conflict, that is, they both are enabled.

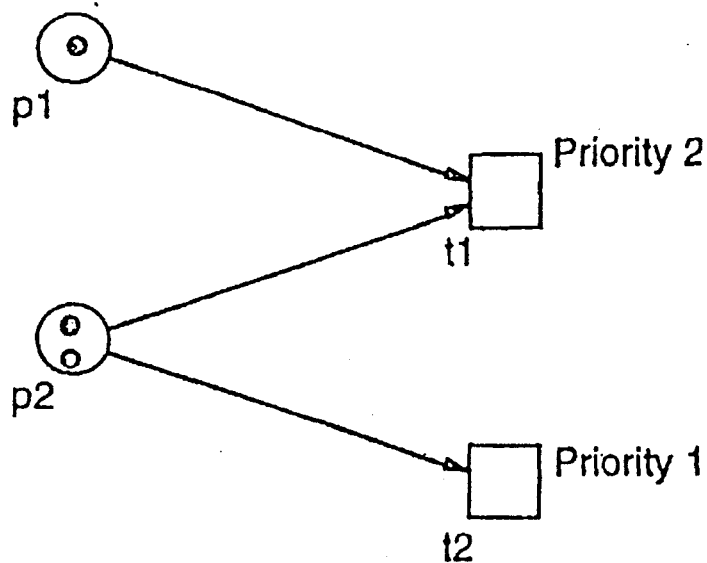


Fig 8.3 Example of use of extended transitions

It is possible to give transitions a priority order for firing in such situations. For example, in Fig. 8.3 transitions t_1 and t_2 are in conflict. If we associate a higher priority with t_2 and lower one with t_1 , and assuming there is an adequate number of tokens in the places (as shown in figure), the transitions will fire in priority order. If there are an

inadequate number of tokens in the places the firing of the higher priority transitions may preclude the firing of a lower priority transition.

When several transitions have the same priority and are in conflict it is possible to associate a probability with each that will determine which transition will fire first. For example, in Fig. 8.3, t_1 and t_2 are in conflict. We can associate a probability of 0.3 with t_1 and probability of 0.7 with t_2 ; t_2 will then fire first on 70% of the occasions that the two transitions are in conflict.

8.8 Coloured Petri Nets(CPNs)

CPN are extensions of PNs with identical modeling power but with increased graphical conciseness. CPNs allow different attributes (colour) to be associated with tokens and for different coloured tokens to be handled differently by the PN elements. For many modeling problems in construction it is necessary to distinguish between different information, materials, and resource flows. CPNs allow us to do this. Coloured tokens are useful when several tasks need to share a scarce resource. For example, a crane lifting different materials on a construction site: each material can be represented by different colour tokens and handled differently by the crane. Materials, based on their colour, could be delivered to different processes or locations on the site. CPNs are also useful for modeling heterogeneous queuing situations in projects, in cases where equipment fleets are made up of different types of equipment. In this situation non-deterministic timed transitions can, if required, have different time distributions associated with different coloured tokens.

Coloured PN's are defined formally as a quintuple

$$CPN = \langle P, T, C, IN, OUT \rangle \quad (8.6)$$

$$\text{Where } P = \{p_1, p_2, p_3, \dots, p_n\} \text{ the set of } n \text{ places} \quad (8.7)$$

$$T = \{t_1, t_2, t_3, \dots, t_m\} \text{ the set of } m \text{ transitions} \quad (8.8)$$

Where $C(p)$ and $C(t)$ = set of colours associated with place $p \in P$ and $t \in T$

$$C(p_i) = \{a_{i1}, a_{i2}, a_{i3}, \dots, a_{iq}\} \text{ where } i = 1, 2, 3, \dots, n$$

$$\text{and } q = |C(p_i)| \quad (8.9)$$

$$C(t_j) = \{b_{j1}, a_{j2}, a_{j3}, \dots, b_{jr}\} \text{ where } j = 1, 2, 3, \dots, m$$

$$\text{and } r = |C(t_j)| \quad (8.10)$$

Where a and b = associated colours.

$$IN(p, t): C(p) \times C(t) \rightarrow N \quad (8.11)$$

is an input function and

$$OUT(p, t): C(p) \times C(t) \rightarrow N \quad (8.12)$$

is an output function where N = set of all nonnegative integers. The input arc from a place p_i , with respect to the colour a_{ih} , to a transition t_j , with respect to the colour b_{jk} , is denoted by the scalar $IN(a_{ih}, b_{jk})$ for some hand k . similarly, an out put arc is denoted by the scalar $OUT(a_{ih}, b_{jk})$.

The firing rules for a CPN are as follows

1. A transition fires if and only if the token colours marking the input places are members of the colour set associated with the transition.

2. When a transition fires, the token colours defined by the input functions, $IN(p, t)$ are removed from the input places.
3. Similarly the colour sets of the output places as defined by the output function, $OUT(p, t)$ are marked.
4. Colours associated with functions are allowed to change across transitions.

8.9 Queuing disciplines for places

This extension to basic PNs allows queues of tokens that form at places to be governed by a queuing discipline. Such a facility is important in modeling some construction systems when resources are shared or where it is necessary to distinguish between queue members (if the queue members have different attributes) and process them accordingly. An example occurs in many earthmoving systems when first-in-first-out queues are used for loading and different capacity trucks take different times to be loaded.

GRAPHICAL EVALUATION AND REVIEW TECHNIQUE (GERT)

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 9

GRAPHICAL EVALUATION AND REVIEW TECHNIQUE (GERT).

9.1 Introduction

GERT is mostly concerned with the generalizations of the basic CPM/PERT type of network to permit probabilistic activities and nodes. The components of GERT networks are directed branches (arcs, edges, transmittances) and logical nodes (vertices).

Two parameters are associated with the branch:

1. The probability that a branch is taken, p , given that the node from which it emanated is realized.
2. A time, t , required, if the branch is taken, to accomplish the activity, which the branch represents.











The time, t , can be a random variable. If the branch is not part of the realization of the network then the time for the activity represented by the branch is zero.

A node in a GERT network consists of an input (emitting, distributive) function and an output (emitting, distributing) function. Three logical relations on the input side and two types of relations on the output side will be considered. This yields six types of nodes, which are described in Table 9.1.

First, consider the AND node as depicted by Node.3 in Fig.9.1. This is the standard node type in PERT networks. By definition Node.3 will be realized only if both a and b are realized. The probability and time associated with the realization of Node.3 are $p_a p_b$ and $\max(t_a, t_b)$ assuming activities a and b are independent.

TABLE 9.1

Node Characteristics and Symbols

	Input	Exclusive-or	Inclusive-or	and
Output				
Deterministic	D			
Probabilistic				

Exclusive-or: The realization of any branch leading into the node causes the node to be realized; however, one and only one of the forward branches leading into this node can be realized at a given time.

Inclusive-or: the realization of any branch leading into the node causes the node to be realized. The time of the realization is the smallest of the completion times of the activities leading into the *inclusive-or* node.

And: The node will be realized only if all the branches leading into the node are realized. The time of the realization is the largest of the completion times of the activities leading into the *and* node.

Deterministic: All branches emanating from the node are taken if the node is realized, that is, all branches emanating from this node have a p -parameter equal to one.

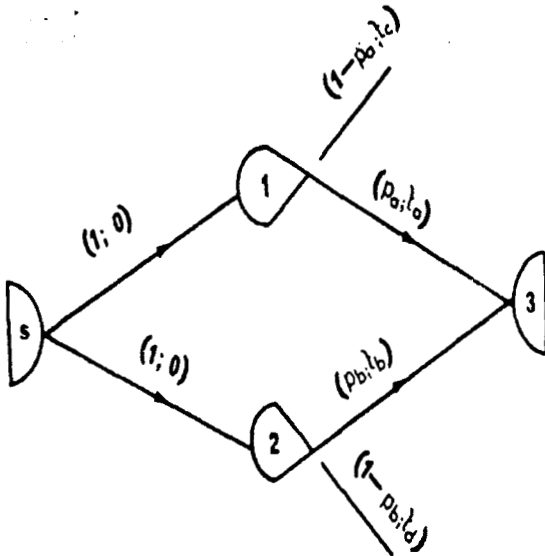


Fig 9.1 A simple network with an *AND* node

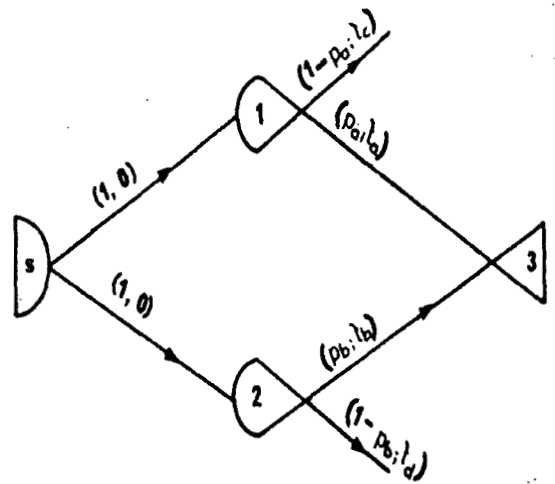


Fig 9.2. A simple network with an *inclusive-or* node.

Probabilistic: At most one branch emanating from the node is taken if the node is realized. For the *inclusive-or* relation, the analysis proceeds as in the *AND* case. The *inclusive-or* node is shown in Fig.9.2.

The reduction process involves the enumeration of all mutually exclusive alternative methods of realizing node 3 from node S. These are listed in Table 9.2. For the *Exclusive-or* relation, the probability and time parameter can be combined into a single parameter as shown in Equation 9.1.

$$w(s) = pM_t(s) \quad (9.1)$$

TABLE 9.2

Possible Events for *inclusive-or* Nodes

Branch occurrence	Probability	Equivalent time
a but not b	$p_a - p_{a \cap b}$	t_a
b but not a	$p_b - p_{a \cap b}$	t_b
a and b	$p_{a \cap b}$	$\min(t_a, t_b)$


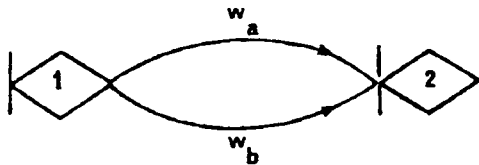
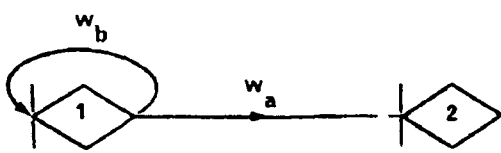
where $M_t(s)$ is the moment generating function (or other exponential transform) of the time parameter. For net work consisting of *exclusive-or* nodes, $w(s)$ can be calculated using (9.2), the topology equation or Mason's rule of signal flow graph theory

$$w_g(s) = \frac{\sum (\text{path } i) \left[1 + \sum_m (-1)^m (\text{loops of order } m \text{ not touching path } i) \right]}{\left[1 + \sum_m (-1)^m (\text{loops of order } m) \right]} \quad (9.2)$$

Equation (9.2) defines the equivalent w -function, $w_g(s)$ for any net work consisting only of *exclusive-or* node:

In Equation (9.2), a loop is defined as a sequence of branches such that every node is common to two and only two branches of the loop, one terminating at the node and the other emanating from that node. In a first-order loop every node can be reached from

every other node. A loop of order n is set of n disjoint first-order loops. Disjoint loops are loops, which have no nodes in common. The parameter of the loop is the product of the parameters of the branches of the loop. A forward path is a sequence of branches from one node to the other such that every node except the two specified is common to two and only two branches of the forward path. The difference between a forward path and a loop is that the start node of a path has no input branch and the terminal node of a path

<u>Network type</u>	<u>Graphical representation</u>	<u>Paths</u>	<u>Loops</u>	<u>Equivalent function w_E</u>
Series		$w_a w_b$	--	$w_a w_b$
Parallel		$w_a w_b$	--	$w_a + w_b$
Self-loop		w_a	w_b	$\frac{w_a}{1 - w_b}$

has no out put branch.

Fig. 9.3 Calculation of the equivalent w - function

In Fig. 9.3, the equivalent w - function is calculated for three basic network types.

From the definition of $w(s)$ we have

$$p = w(o) \tag{9.3}$$

and
$$M(s) = \frac{w(s)}{p} \tag{9.4}$$

since $M(o) = 1.$

Thus, the probability of realizing a node or the moment generating function of the time to realize a node can be obtained from knowledge of $w(s)$. An example of how GERT is used for analyzing the research and development expenditures is given in the following example (Alan B. Pritsker, 1968).

9.2 Analysis of research and development expenditures

An example of the analysis of research and development expenditure using the network is shown in Fig.9.4.

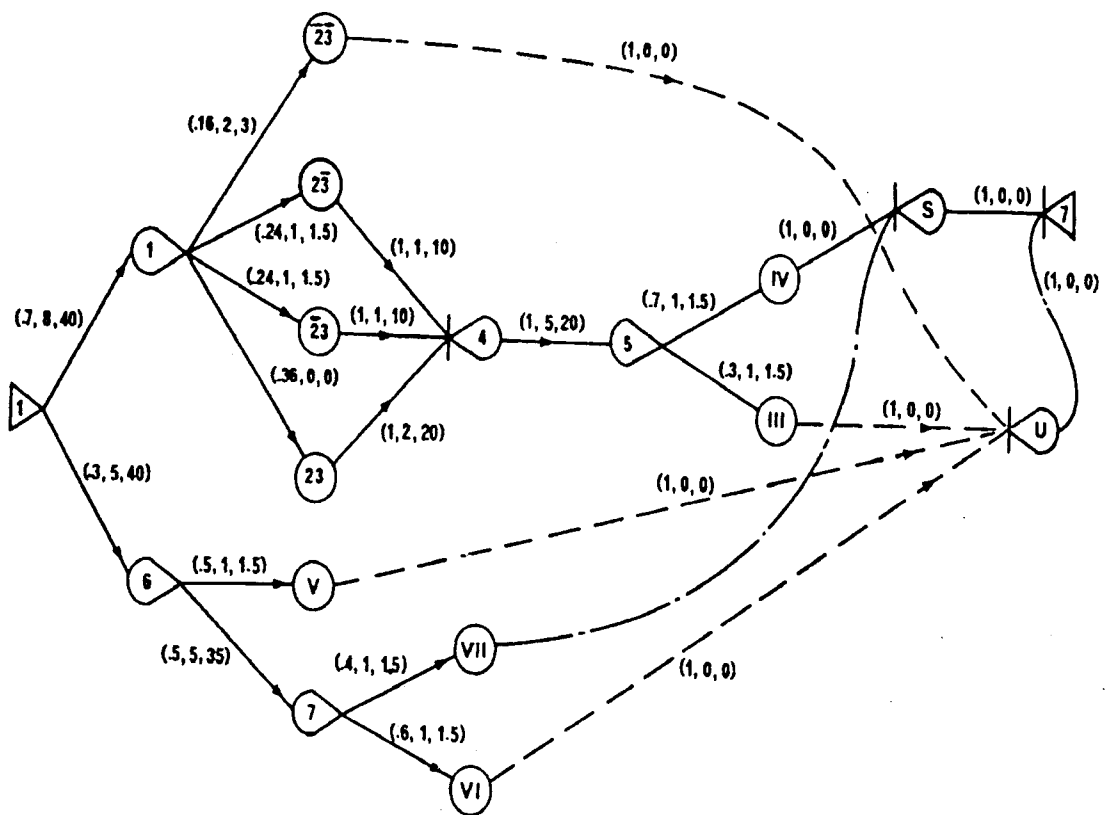


Fig 9.4. GERT network of a research and development process

The definitions of events and outcomes (nodes) and the activities (branches) are given in Table 9.3.

TABLE 9.3

Definitions for research and development Network

Events

1. *Feasibility study indicates electrical control of high temperature is/is not feasible.*
2. *AC control found suitable /unsuitable.*
3. *DC control found suitable /unsuitable*
4. *Optimum integration of AC/DC circuits achieved.*
5. *Unit found to be within/outside potential market price.*
6. *Pneumatic control found to be feasible/ unfeasible.*
7. *Unit found to be within/ outside potential market price.*

Outcomes

- i. *Project dropped.*
- ii. *Project dropped.*
- iii. *Project dropped.*
- iv. *Project put into production and marketed.*
- v. *Project dropped.*
- vi. *Project dropped.*
- vii. *Project put into production and marketed.*

Activities

- A. *Pneumatic feasibility study.*
- B. *AC control investigation.*
- C. *DC control investigation.*
- D. *Report writing.*
- E. *Investigation of optimum AC/DC integration.*
- F. *Report writing.*
- G. *Investigation of optimum AC/DC integration.*
- H. *Economic analysis of system.*
- I. *Report writing.*
- J. *Report writing.*
- K. *Report writing.*
- L. *Economic analysis of system.*
- M. *Report writing.*
- N. *Report writing.*

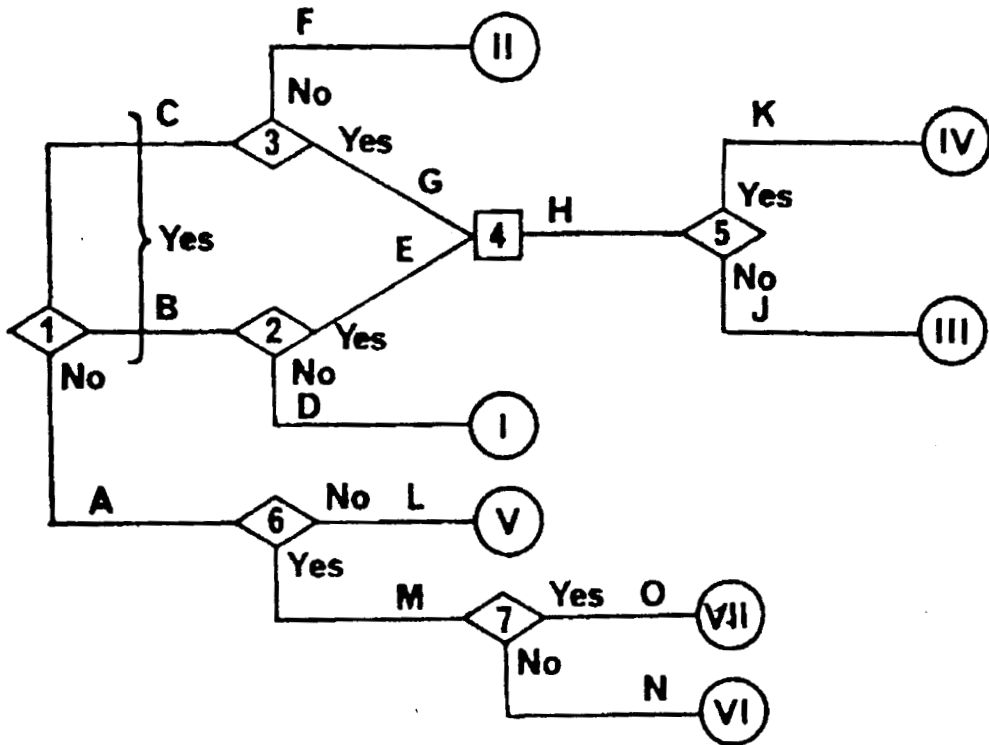


Fig 9.5 Decision box network

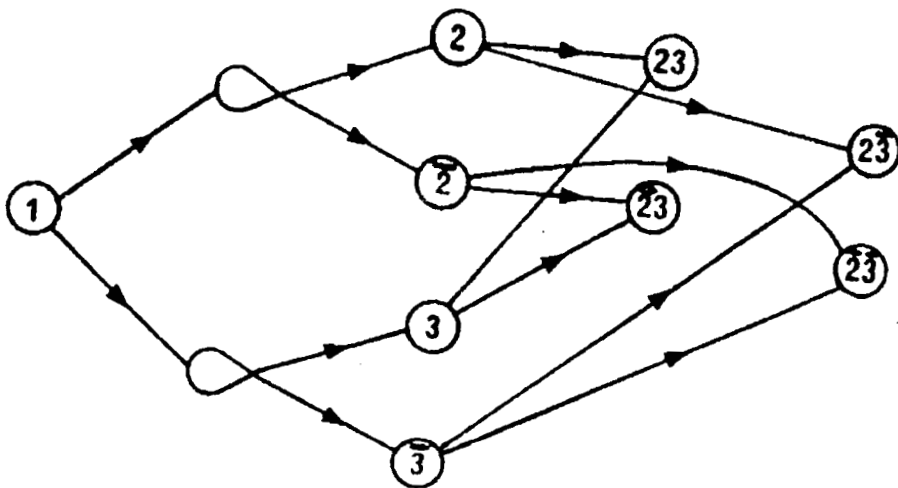


Fig 9.6 Detailed segment of the network

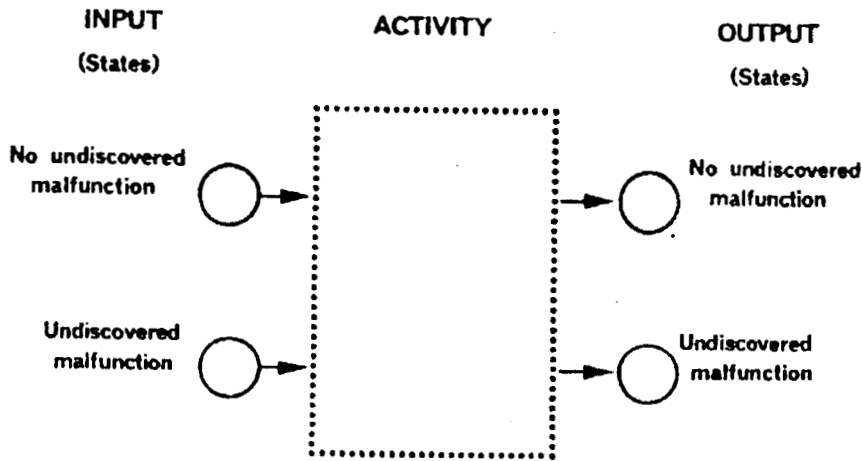


Fig 9.7. A two-port description of an activity

For each branch of the network, it gives the probability that the branch is realized given that the preceding node is realized, and the time and cost associated with the activity represented by the branch if the activity is performed. These values are inserted on the GERT network given in Fig.9.4 by an ordered triple of probability, time (weeks) and cost in Rs.1,000 units, namely, (p, t, c) . Time in this example is not a duration but the amount of effort required to perform the activities measured in weeks.

While constructing the GERT network first the AC and DC control investigations (Activities B and C) are performed simultaneously and this should be indicated on the network without the aid of the bracket. Second nodes I and II do not result in the project being dropped as implied in Fig.9.5. Also, the decision nodes

represent specific events, not *either-or* types of events. For ease of reference between Figs. 9.4 and 9.5, nodes have been labeled with two numbers (2 and 3) and the complements of these numbers ($\bar{2}$ and $\bar{3}$). Thus, node $\bar{2}\bar{3}$ represents the event AC control has been found to be suitable and DC control has been found to be unsuitable. The detailed segments of the network between node 1 and combination of nodes 2, $\bar{2}$, 3 and $\bar{3}$ is given in Fig.9.6.

Third, three terminal nodes, U, S and T, have been added in Fig 9.4. Node U represents the event “project dropped”, S represents “project successful”, and node T represents the event “project terminated” whether it was successful or not.

The GERT analysis for the network presented in Fig.9.4 requires the extension of the w -function to handle two additive parameters. If t and c are independent or information is only desired about them separately, the w -function for a branch becomes

$$w(s_1, s_2) = pe^{s_1 t + s_2 c}. \quad (9.5)$$

For an event of interest, say Event IV, one has

$$\begin{aligned} w_{I-IV}(s_1, s_2) = & (17e^{8s_1 + 40s_2}) [0.24e^{2s_1 + 11.5s_2} + 0.24e^{2s_1 + 11.5s_2} \\ & + 0.36e^{2s_1 + 20s_2}] (e^{5s_1 + 20s_2}) (0.7e^{s_1 + 1.5s_2}) \end{aligned} \quad (9.6)$$

The performance measure associated with event IV are computed by

$$p_{1-IV} = w_{1-IV}(0,0) = (0.7)(0.84)(0.7) = 0.4116, \quad (9.7)$$

$$E\{t_{1-IV}\} = (\partial/\partial s_1) \left[(1/p_{1-IV}) w_{1-IV}(s_1, 0) \right]_{s_1=0} = 14.90 \text{ weeks,}$$

$$E\{C_{1-IV}\} = (\partial/\partial s_1) \left[(1/p_{1-IV}) w_{QIV}(0, s_2) \right]_{s_2=0} = \text{Rs. } 76,643 \text{ (thousands)}$$

9.3 Queuing analysis using GERT

An example of a machine- spare - repairman problem is considered here for analysis. A plant consists of one machine and a spare machine. The time between failures for the machine is exponential with mean rate λ . A single repairman is employed and his service time is exponential with mean rate μ . The GERT network describing this queuing system is given in Fig.9.8 where the node represents the number of failed machines.

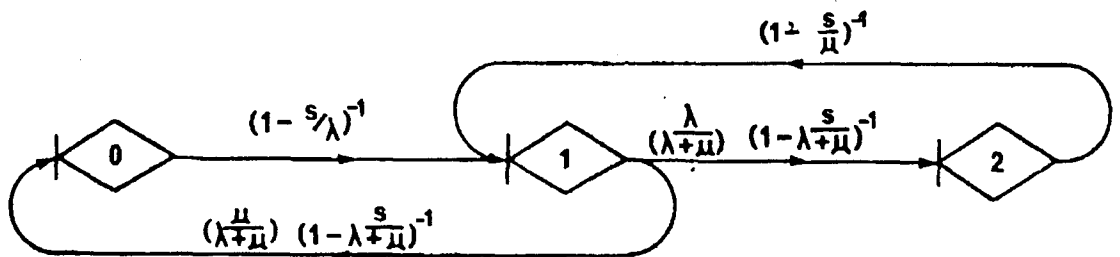


Fig 9.8. GERT representation of machine – spare – repairman problem

The branch from node 1 to node 2 represents the activity that the spare fails before the machine is fixed which has a probability equal to $\lambda / (\lambda + \mu)$ and the time to fail, given failure occurs before service, is exponential with mean rate $(\lambda + \mu)$. A similar analysis is required for the activity between node 1 and node 0.

The information obtainable from the GERT analysis is given in Table 9.4.

In addition to the information presented in Table 9.4, the steady-state probabilities can be obtained from the following equation:

$$p_i = \frac{\text{Expected time in state } i \text{ during a cycle}}{\text{Expected cycle time}}$$

TABLE 9.4
GERT Analysis of a Queuing problem

Description	Network Modification	Moment Generating Function
Time to plant failure	Delete Branch 2-1	$M_{0,2}(s) = \frac{[\lambda / (\lambda + \mu)] [1 - s / (\lambda + \mu)]^{-1}}{1 - [\mu / (\lambda + \mu)] [1 - s / (\lambda + \mu)]^{-1} (1 - s / \lambda)^{-1}}$
Duration of plant failure	_____	$M_{2,1}(s) = (1 - s / \mu)^{-1}$
Busy period	Delete Branch 0-1	$M_{1,0}(s) = \frac{[\mu / (\mu + \lambda)] [1 - s / (\lambda + \mu)]^{-1}}{1 - [\lambda / (\mu + \lambda)] [1 - s / (\lambda + \mu)]^{-1} (1 - s / \mu)^{-1}}$
Idle time	_____	$M_{0,1}(s) = (1 - s / \lambda)^{-1}$
State 0 renewal time	Split node 0 into node 0 and node 0'	$M_{0,0}(s) = M_{0,1}(s) M_{1,0}(s)$

It can be said that GERT is a general procedure for the formulation and evaluation of systems using a graphical approach. The GERT approach to problem solving utilizes the following steps:

1. Convert a qualitative description of a system or problem to a model in stochastic network form.
2. Collect the necessary data to describe the branches of the network.

3. Determine the equivalent function or functions of the network.
4. Convert the equivalent function into the following two performance measures of the network:

The probability that a specific node is realized.

The moment generating function of the time associated with an equivalent network
5. Make inferences concerning the system under study from the information obtained in 4 above

VENTURE EVALUATION AND REVIEW TECHNIQUE (VERT)

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

NB4487

101

658.4032

711
SR:10

CHAPTER 10

VENTURE EVALUATION AND REVIEW TECHNIQUE (VERT)

10.1 Introduction

The strategic analysis of projects requires that the modeling process captures high degree of verisimilitude. The Venture Evaluation and Review Technique, as presently portrayed by the VERT program, offers the user a wide range of options: the ability to specify one of 14 probability distributions, any one of which may be used to describe one, two or three measures on each arc. Typically these measures are called 'time', 'cost' and 'performance', in many project environments – but these measures could be any description important for the analysis. Furthermore, the user may specify several logical node descriptions: *AND*, *PARTIAL AND*, *OR* and *MONTE CARLO*. There are also *COMPARE* and *FILTER* nodes, which may be used to 'terminate' or allow further processing of the network according to present managerial criteria. These facilities are described as the flexible rules for aggregating data within the network based on 50 pre-set arithmetic operations. As much of the use of VERT has been in the area of US national defense, there has been little published work.

VERT builds up its results via simulation. In each pass through the project, sample data are taken from the specified distribution on each arc. These are aggregated, compared and /or filtered, following the logic of the network. Whatever the outcome of each pass through the network, the data therein are merged with all past data to build up probability distributions of success and /or (sources) of failures. VERT will not provide a direct basis for resource allocation as would PERT/CPM, but it will offer management a flexible tool for the strategic analysis of a project. Once managements are sure that the

sources of uncertainty have been fully researched using VERT it is a simple, though potentially a time consuming task, to translate the network to a CPM or a PERT net within the mini or main-frame computer to explore the resource allocations that would ensure project completion according to the VERT-derived characteristics. Equally, it is simple to transfer the data to a PC-based program to undertake the generation of appropriate graphs of work and cash flow allocations for project control. Some PC-based software allows direct exchange of data between a PC and a mini or mainframe; but none of these packages allow for the very flexible analysis of uncertainty and the realistic input of various data distributions as in the VERT package.

10.2 An example network

As an example, a simple 12-activity network is used to indicate the network output that may be obtained from many PC-based project analysis program. Note however, that all the entered data refer to tasks with a fixed duration; so one may become too confident that the overall project duration will be 39 days exactly, as suggested by the result of the critical path calculation.

Table 10.1. Critical path (and PERT) data

ID.	Task	t	t _o	t _m	t _p	t _e	σ^2
1	M/c1	2	1.5	2	3.5	2.16	0.11
2	M/c2	3	2.0	3	6.0	3.33	0.44
3	Make A	2	1.5	2	3.5	2.16	0.11
4	Make B	2	1.5	2	3.5	2.16	0.11
5	Make final tester	20	15.0	20	35.0	21.62	11.1
6	Test A	3	2.0	3	6.0	3.3	0.44
7	Test B	4	3.0	4	7.0	4.33	0.44
8	Make 1 st assembly	3	2.0	3	6.0	3.33	0.44
9	Test final assembly	5	4.0	5	7.0	5.16	2.50
10	Make C	8	6.0	8	12.0	8.33	1.00
11	Test C	20	15.0	20	35.0	21.62	11.1
12	Make final assembly	3	2.0	3	6.0	3.33	0.44

Figure 10.1 shows the VERT network when the same data was run through the VERT-4 program (Kidd, J, 1989) mounted on a computer. It has been assumed that task durations vary according to a *beta* distribution.

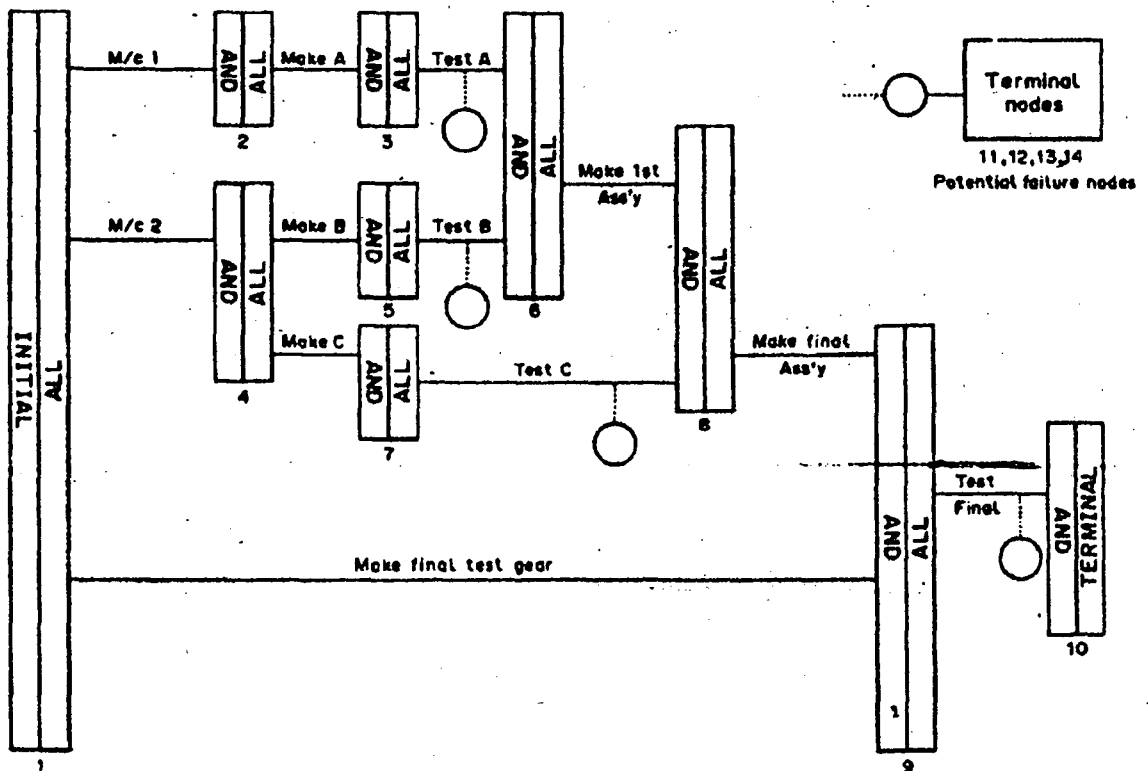


Fig 10.1 The VERT network

Allowance was made for variations that may be typical of real projects. The data of Table 10.1 depict the 'optimistic', 'usual' and the 'pessimistic' estimates of durations. The beta distributions used with in the VERT package was bounded just outside these estimated times to enforce the reality of a '100%' distribution range for the sampling of times. The parameters of the *beta* distribution were 4,8 in all cases – so the skew ness was not too wild: although personal, subjective experience of real tasks would indicate the appropriate parameters to use in a 'real' analysis. The mathematical functions within

VERT were used to link the sampled durations of each activity in each simulation to the cost structures as indicated in Table 10.2.

Table 10.2. Cost data on each task

ID	Task	Duration in days	Variable cost	Fixed cost
1	M/c1	2		
2	M/c2	3		
3	Make A	2	7.5	7000
4	Make B	2	9.5	2500
5	Make final tester	20		
6	Test A	3	12.5	450
7	Test B	4	12.5	450
8	Make 1 st assembly	3	6.75	600
9	Test final assembly	5	8.0	125
10	Make C	8	9.5	2500
11	Test C	20	12.5	450
12	Make final assembly	3	6.75	650

Note-All variable costs are per hour for an eight-hour day, and tasks 1, 2 and 5 have no directly attributable costs for this example.

Table 10.3. Times and costs at terminal node 10 under differing model assumptions.
(From the VERT-4 program) (Kidd.J., 1989)

Model	Time	se	Cost	se
CPM	39.00	-	18,949	-
PERT	40.12	3.11	19,091	307
VERT*	40.41	3.23	19,115	332

* with only 68% success rate given each test activity has a 10% chance of failure.

Table 10.4. Detail of the failure instances

Node	Probability	Time	se	Cost	se
Test A	0.094	7.1	0.8	7880	68
Test B	0.096	9.3	0.9	3515	69
Test C	0.069	32.7	3.6	5717	362
Final Test	0.064	41.2	3.3	19,175	334
		Total	32.3% failure rate overall		

Here a failure rate of 10% has been assumed at each test activity, which results in the overall probability of a successful completion of the project to be reduced to two thirds. The outcome data in terms of times and costs, and the detailed derivations of the sources of the failures to complete the project are shown in Tables 10.3 and 10.4.

10.3 General comment on VERT

The use of a simulation package, like VERT, will allow the manager to explore the strategic aspect of the project management. Naturally this will seem to absorb more precious time – but this will be well spent, if through realistic analyses of uncertainties, more vagaries of the project could be resolved. In a slightly different context this topic can be explored with respect to managerial and organizational effectiveness when related to information management. They inform us that the management of complexity is the corner stone of managerial activity- that is, the variety in the managers' environment has, through the application of cybernetic principles, to be matched by the variety absorption of the available managerial tools. The managers have to be able and willing to absorb and analyze the uncertainties inherent in the projects under their control.

The example network given above was intended to illustrate the dilemma of the PC – based manager; the so-called end-user, at whom more and more versatile software is aimed. In addition to *Microsoft project*, there are also available software packages based on a database structure such as *Panorama* (with Oracle, a relational database); and there is a wealth of packages based on a compiled language, such as C. In this category there are for example, *project manager workbench*, *Pert master advance* and *Total project manager* to name but three well-known packages. All will offer, as mentioned above, good input facilities, easy manipulation of data and task logic, resource smoothing and scheduling and many graphical displays and outputs.

The example project, Fig. 10.1, exhibits few difficulties for a PC-based analysis—except perhaps there may be conflicts in the timing of the tests depending on the multiple or singular availability of inspectors. The task durations and their associated costs are fixed, thus the project will be completed on day 39 at a cost of 18,949. The cash-flow graph, Fig. 10.2., shows a potential problem as there is a high early demand for cash, but, given warning, no doubt the firm will cope up with. Many outputs are available from the package, for instance, based on Gantt charts with associated cost structures; there are prints of the network to clearly illustrate dependencies.

Adding one stage of complexity to analysis we may utilize the uncertainties of the task durations. In the second VERT analysis allowed the tasks to be described by their various *beta* distributions, and each sampled time was used to derive the cost for that activity according to the data given in Table 10.2. Table 10.3 shows that the mean duration for the project has increased to 40.12 days with a standard error of 3.11 days about this mean.

The PERT and VERT project durations are expected to be longer than a CPM duration simply because of the effect of the pessimistic estimates of the durations of individual tasks. Table 10.1 indicates the task duration variations that classically in PERT yield a project mean time of 41.77 days and a project variation of 15.48, given the critical path passes through the arcs: Obtain m/c2, Make C, test C, Make Final Assembly and Final Test. It is unusual to find the calculated project duration exceeding that found by simulation, as from VERT for example, since the classical calculation ignores all effects

of non-critical path interactions with the progression along the critical path. The usual underestimate of the duration of a typical "fat" real network by the classical PERT calculation is of the order of 15% against other methods. VERT will estimate the probable duration of a real project with some exactitude. And incidentally, the partial path durations and their associated costs and or performance indices will be similarly well estimated.

If we take another viewpoint we may recognize that any activity has a finite probability of completion. Normally, for most tasks we say this is 100%, but for some tasks there is a discernibly lower probability of completion. In the example, the success of the tests is taken to be only 90%. We have to ascertain the effect this will have on the overall success of the project, and what effect this may have on the project, and what effect this may have on the cash -flows. In VERT this form of description is both normal and easy to input to the program.

In this example the path costs have been computed by VERT, as well as the overall average, median and modal costs of the project. Since all costs up to the instant of failure will have to be borne by the firm there is merit in reviewing the magnitude of the path cost at the various failure points relative to the overall cost: see Table 10.4.

In VERT it is possible to place *FILTER* or *COMPARE* node that will divert the progress to the simulation, possibly to a failure node, or to some other route according to the achievement of the simulation up to that point of the project. In other words, the management may consider that certain time, cost or performance indices should have

been met by, let us say mid-way through the project. These may be entered, as a 'questioning' node so forcing results to be derived that should enable a useful strategic analysis to be undertaken.

If there is only uncertainty in the durations of tasks (perhaps the project is a regular activity) we would suggest using PERT to note the probabilities of cost and time distributions, which may interact with the tactical and /or strategic issues of the firm. Finally, once the network has been well researched, it would be reasonable to use the (instantaneous) power of a PC to obtain lists and schedules of work through one of the many package that are on offer. Thus there has to be a well-managed link between the users of the PC system and, say, the powerful VERT system if one is working within a project management environment that is inherently uncertain - and what R&D system can predict with certainty, its timings and cash flow needs?

FUZZY SETS THEORY

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 11

FUZZY SETS THEORY

11.1 Introduction

Construction projects are normally executed in an environment characterized by varying degrees of uncertainties. The presence of these uncertainties, in the scheduling process, can have a significant impact not only on estimating the duration of construction activities, but also on network calculations. Historical data, if available, may prove useful in estimating activity durations. In practice however, the majority of construction companies do not systematically record the duration of project activities for improving their estimate on new projects. In addition, it is commonly known that no two construction projects are alike. The conditions for executing those projects at the activity level may also vary from one project to another. Accordingly, the duration of construction activities are usually estimated by scheduling experts on a case-by-case basis, using their own judgment, experience, and project information that is available to them. With the presence of uncertainties, it is some times difficult for experts to give a definitive estimate of activity duration. In practice, linguistic terms such as “approximately”, “more or less”, or “about” are commonly found in the statements used by these experts. These terms clearly exhibit some sort of imprecision that naturally leads to a range of position values, rather than a definitive estimate using a single value (e.g. five days). In addition, an expert’s expression such as “plausibly between five and ten days, but most likely between seven and eight days” indicates not only that the expert is not certain about his two estimates, but also that he or she has a different degree of belief in each estimate. This range of possible values together with their associated degree of

belief, if acquired in a non-statistical manner, should be modeled using the fuzzy sets theory, which is specifically developed for this type of uncertainty.

11.2 Fuzzy sets theory

Fuzzy sets theory was developed specifically to deal with uncertainties that are not statistical in nature. The concept of fuzzy sets theory differs from that of the conventional crisp sets mainly in the degree by which an object belongs to a set. In crisp set theory, objects are either included or excluded from a set. In fuzzy sets theory, on the other hand, objects are described in such a way so as to permit a gradual transition from being a member of a set to a non-member. Each object contains a degree of membership ranging from zero to one, where zero signifies non-membership, one indicates full membership, and values in between describe the degrees of partial membership. A Fuzzy set, A , is defined as a set of pairs, $[t, \mu_A(t)]$, where t is an object or an element in the universe of discourse, and $\mu_A(t)$ is the degree of the membership associated with the element t . In the situation where t is a continuous variable, the degree of membership can be represented by a function, commonly known as a membership function. Membership functions can take various shapes and forms. One of the most widely used forms is called fuzzy numbers.

11.3 Fuzzy number

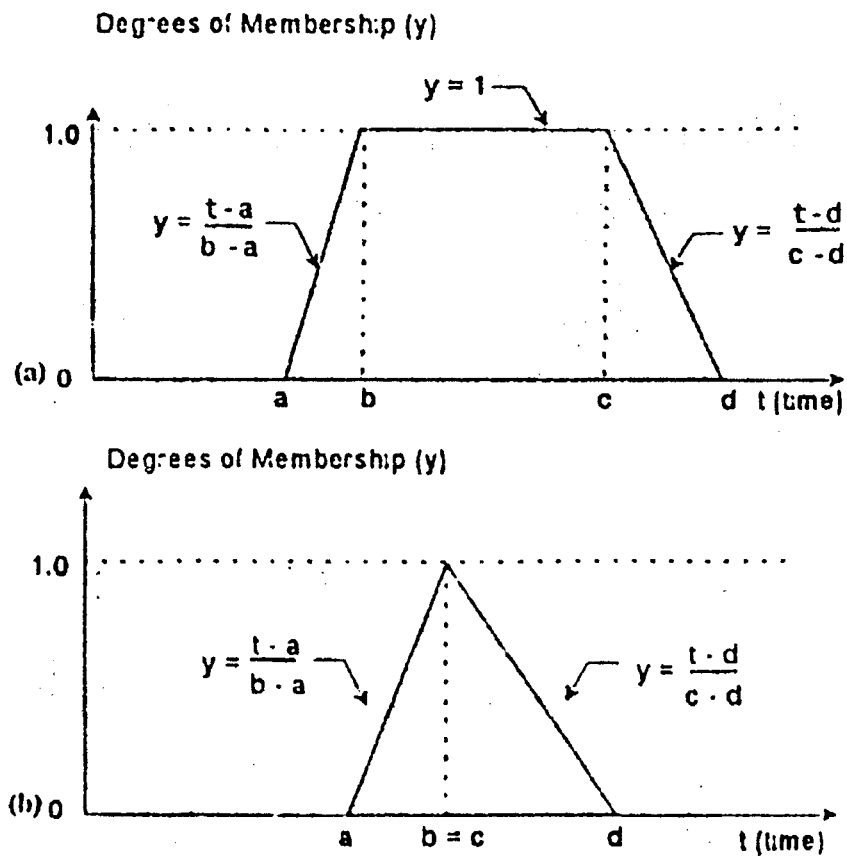


Fig 11.1 Fuzzy numbers (a) Trapezoidal fuzzy number; (b) Triangular fuzzy number

A fuzzy number is a continuous fuzzy set that contains two properties: (1) convexity and; (2) normality. The convexity indicates that the membership function has only one distinct peak, while the normality ensures that at least one element in the set has a degree of membership equal to 1.0 [i.e. definitely possible value (s)]. These two properties make the concept of fuzzy numbers attractive and naturally suitable for modeling imprecise quantities such as “approximately one week,” or “more or less than five days.” Theoretically, fuzzy numbers can take various shapes. In modeling real-life problems, however, linear approximations such as the trapezoidal and triangular fuzzy

numbers are frequently used. A trapezoidal fuzzy number (TFN) can be represented by a quadruple (a, b, c, d) [see Fig. 11.1(a)], where a and d are the lower and upper bounds, b and c are the lower and upper modal values, respectively, and t is an element between a and b . A triangular fuzzy number can be viewed as a special case of the TFN with $b=c$ [see Fig. 11.1(b)].

The generic membership function of the TFN can be expressed as

$$\mu(t) = \begin{cases} \frac{t-a}{b-a} & a < t < b \\ 1 & b \leq t \leq c \\ \frac{t-d}{c-d} & c < t < d \\ 0 & \text{other wise} \end{cases} \quad (11.1)$$

The magnitudes of the left spread, the middle plateau, and the right spread signify, collectively, the degrees of uncertainties associated with each TFN.

11.3.1 Operation of TFNs

There are a number of operations that can be performed on TFNs. Following are operations employed in the development of a method for project network analysis:

Let $M = (a_1, b_1, c_1, d_1)$ and

$N = (a_2, b_2, c_2, d_2)$ be any two TFNs, then:

$$M \subseteq N \text{ iff } \mu_M(t) < \mu_N(t) \quad (11.2)$$

$$M \oplus N = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2) \quad (11.3)$$

$$M \ominus N = (a_1 - d_2, b_1 - c_2, c_1 - b_2, d_1 - a_2) \quad (11.4)$$

$$\text{Max}(M, N) = [\mathbf{V}(a_1, a_2), \mathbf{V}(b_1, b_2), \mathbf{V}(c_1, c_2), \mathbf{V}(d_1, d_2)] \quad (11.5)$$

$$\text{Min}(M, N) = [(a_1, a_2), (b_1, b_2), (c_1, c_2), (d_1, d_2)] \quad (11.6)$$

$$M \int N = \{t, [\mu_M(t) \mu_N(t)]\} \quad (11.7)$$

Where \oplus = fuzzy addition; \ominus = fuzzy subtraction; \mathbf{V} = maximum; and \mathbf{and} = minimum

11.4 Time representation

In scheduling, knowledge about time is usually expressed in terms of dates and durations. Dates are used to specify or indicate when an event or milestone should be accomplished in a project, and durations are estimates of the time required to perform the project activities to overall projects activities. The overall project uncertainties may result collectively from various degrees of imprecision associated with delivery dates and/or activity durations, which impact the start and finish times of the project activities. Imprecision in estimates of time, as commonly encountered in construction, can be represented in this method using TFNs. How TFNs are used to describe a wide range of imprecise estimates is discussed hereafter.

11.4.1 Crisp time and Interval

Crisp time can be regarded as a single time or a set of time elements (i.e., interval) where the degree of membership of those included in the set assume a unit value and zero other wise.

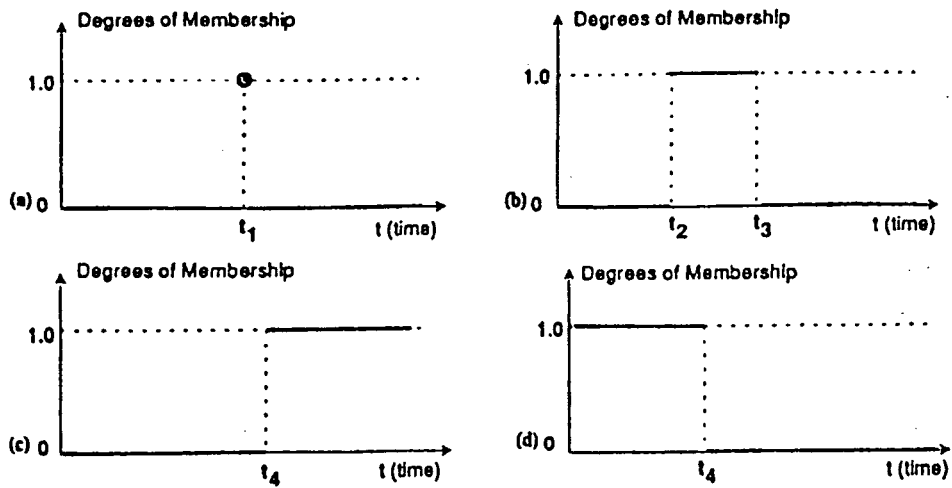
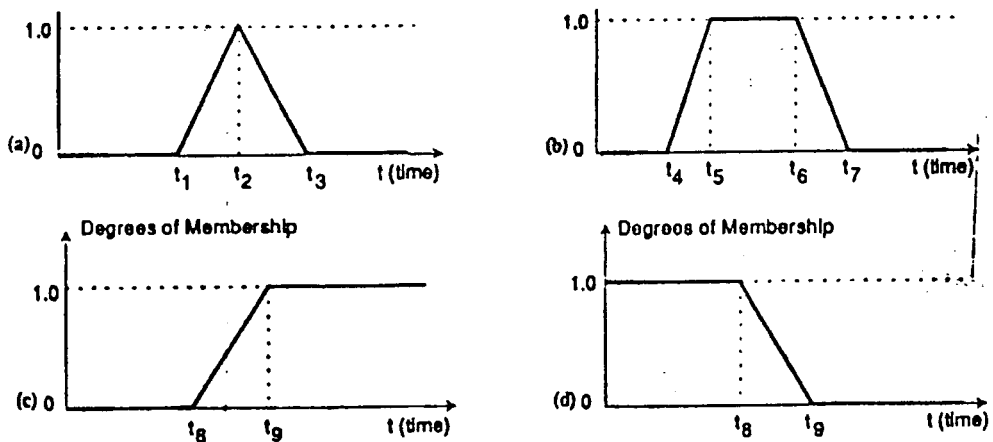


Fig 11.2 Crisp times: (a) Exactly t_1 (b) Between t_1 and t_2
 (c) After t_4 (d) Before t_4



- Fig 11.3 Fuzzy times
- (a) Approximately t_2
 - (b) Approximately between t_4 and t_7
 - (c) Approximately after t_8 but definitely before t_9
 - (d) Approximately before t_9 , but definitely after t_8

Fig.11.2 illustrates four possible crisp times and interval with varying degrees of imprecision. The most definite form of time in which a single value of time, t_1 , is certainly known (e.g., the duration of activity A is 15 days) is shown in Fig 11.2(a). This form of representation is used in deterministic scheduling. In a situation where an exact variable is not representative, a range or an interval of possible values can be specified. A closed interval, (t_2, t_3) as shown in Fig.11.2(b) is suitable for representing this type of variable. By using a closed interval, it is assumed that each time element between (t_2, t_3) , has an equal possibility of occurrence (i.e., the possibility =1.0). In many circumstances, the scheduler only knows that a particular event will occur after or before a certain date. Intervals which are bounded only from one side, such as those in Fig.11.2 (c) and 11.2 (d), are suitable for representing the case in which the scheduler knows for certain that the time in question will definitely occur after t_4 [i.e., (t_4, ∞)]. Fig. 2(d), on the other hand, indicates that the time in question will certainly be prior to t_4 [i.e. $(-\infty, t_4)$].

11.4.2 Fuzzy Time

In the crisp state, the degree of membership of a time element in a set can be either 1 or 0. In real operations, however schedulers might not have, or, in some cases, might not want to specify a sharp boundary between the possible and impossible durations. This phenomena can be regarded as vague or fuzzy. Mathematically, fuzziness can be modeled by assigning the degrees of membership of a variable to take decreasing

value from 1 to 0, allowing a transition from membership to non-membership. This transition naturally represents the way schedulers approximate the values assigned to scheduling variables.

Fuzzy time can be generated by imposing fuzzy boundaries on each of the situations described previously. This results in four different imprecisely known times as in Fig.11.3 (a) - 11.3 (d). The triangular distribution shown in Fig.11.3 (a) represents the situations in which the time in question can assume the value any were between t_1 and t_3 , with the most plausible value being t_2 . Fig.11.3 (b) depicts the trapezoidal distribution, which represents a more imprecise situation. The most plausible values in this case fall between t_5 and t_6 . The degrees of possibility of those elements located outside this interval gradually reduce on both sides to zero at t_4 and t_7 , respectively. All time elements smaller than t_4 or greater than t_7 , are not possible. Fig. 11.3 (c) depicts the time that occurs certainly after t_8 and, most plausibly, after t_9 . Fig. 11.3 (d), on the other hand, represents the time that occurs most plausibly before t_8 , but certainly before t_9 . The distributions shown in Figs. 3 (c) and 3 (d) are called fuzzy bounds, where fuzzy lower bound represents the “after” case, and fuzzy upper bound represents the “before” case. The aforementioned time representation will be used as input in the proposed scheduling method.

Fuzzy operations introduced earlier are used in the following equations to compute fuzzy early times.

$$FES_x = \underset{p \in P}{\text{Max}} (FEF_p) \quad (11.8)$$

$$FEF_x = FES_x \text{ I } FD_x \quad (11.9)$$

$$T_{proj} = FEF_e \quad (11.10)$$

Where FES_x = Fuzzy early start time of activity x ; p = a predecessor activity; P = set of predecessors; FEF = Fuzzy early finish time; FD = Fuzzy activity duration; T_{proj} = Fuzzy project duration and e = last activity in the project.

The *Max* operation performs pair wise comparisons for each and every element in the two quadruples, and accordingly selects the maximum value of each pair to present their respective elements in the new quadruple. This procedure is different from that used in the PERT, where only the expected time values are compared, with no regard or consideration for the standard deviation.

Backward pass calculations using the traditional fuzzy subtraction (11.4) produce unrealistically large uncertainties associated with fuzzy late times (start and finish) of project activities. These uncertainties accumulate rapidly as the backward pass calculation continues to progress. On the completion of the backward pass calculations; some of the earlier activities could end up having a negative fuzzy late time, which, practically, has no meaning. To circumvent this shortcoming, a new technique for backward pass calculations has been developed. This technique is based on the assumption that the right spread of fuzzy late times are at least as uncertain as their respective fuzzy early times. The reason for using only the right spread is explained later.

This assumption stems from the fact that in a CPM network, the uncertainty associated with activity duration directly influence the degree of uncertainties associated with the calculation of start and finish times. For example, the magnitude of the uncertainty associated with the fuzzy start times of an activity is simply the accumulation of the uncertainty associated with the durations of its predecessors. The uncertainty associated with the finish times of those activities, however, can be assumed equal to that of its start plus the uncertainty associated with its own duration. This assumption can be justified by considering, for example, a linear type project network. Since the network has only one path, the early and late start times of any activity obtained from forward and backward pass calculations must be identical, (i.e., equally uncertain). This is true also for the finish times of that network. In most cases, however project network has only one path. In the backward pass calculations, higher uncertainties may be accumulated from the different path (s) than those obtained in the forward pass calculations. This assumption, therefore, allows activity late times to be more uncertain than its respective early times. This adds more flexibility and accuracy for calculating fuzzy late start and finish times of project activities

To ensure that the results obtained are correct and meaningful, the following three conditions must be met: (1) scheduling constraints resulting for forward and backward pass calculations must be satisfied; (2) all four elements in the quadruple must have a positive value; and (3) each element should not exceed its successor [e.g. for the quadruple (a, b, c, d) ; $a \leq b \leq c \leq d$]. These conditions can be satisfied simultaneously through the use of fuzzy bounds such as those shown in Figs.11.3 (c) and 11.3 (d). This is

mainly due to the ample time window given by each fuzzy bound. This adds more flexibility in calculating the fuzzy late times without violating the constraints produced from the forward and backward pass calculations. Consider a fuzzy late time expressed by a closed TFN indicating the latest possible time period by which an activity must be finished. All time elements generated from the backward pass calculations having value beyond the upper bound of that TFN [i.e., element d in the quadruple (a, b, c, d)] are not possible. Modifying the fuzzy late finish by extending its lower boundary (i.e., the left spread) to infinity physically means that all the time that occur before its lower bound [i.e., element a in the quadruple (a, b, c, d)] are possible. The degree of possibilities for the time elements located after d remains unchanged. As a result, the original fuzzy-late-finish constraint is not violated, and calculated fuzzy duration is still valid. The extended fuzzy late time is called the fuzzy upper bound. An example of the fuzzy upper bound is shown in Fig. 11.3 (d).

The conversion of a TFN to its upper bound is simple and direct. Let A be the TFN characterized by the quadruple (a, b, c, d) . Its upper bound denoted by A^* can be obtained by subtracting an interval $(0, \infty)$, expressed in the form of the quadruple $(0, 0, \infty, \infty)$ from A as follows:

$$A^* = (a, b, c, d) \text{ I } (0, 0, \infty, \infty) = (-\infty, -\infty, c, d) \quad (11.11)$$

The computation of fuzzy late times of each activity begin with the determination of its preliminary late finish (PLF). The definition of PLF is analogous to the late finish time used in CPM. The PLF can be calculated for any activity x as

$$PLF_x = \min_{s \in S} (FLS_s) \quad (11.12)$$

where FLS = fuzzy late start time of its immediate successor (s); and S = set of immediate successors of x .

The PLF is then converted to the upper bound for the late finish (FLF^*) using (11.11). Based on the calculated FLF^* and FEF , and the assumption about the propagation of uncertainty stated earlier, it is possible to compute the fuzzy late finish time (FLF). Suppose that the FEF is characterized by (a, b, c, d) and the FLF^* is represented by $(-\infty, -\infty, e, f)$, a procedure to calculate fuzzy late times for project activities is described as follows:

(1). Determine which of the two fuzzy quantities has a greater right spread by comparing the results of $(f - e)$ and $(d - c)$.

(2). Calculate Y , the largest fuzzy number that satisfies the following condition.

$$FEF \cap Y \subseteq FLF^* \quad (11.13)$$

If the right spread of FEF is more uncertain [i.e., $(d - c) \geq (f - e)$], the right spread of FLF is set equal to that of FEF . In this case, all the quadruple elements of Y are equal to the difference between the elements of the upper bounds of the two fuzzy quantities

$$Y = (f - d, f - d, f - d, f - d) \quad (11.14)$$

On the other hand, if the right spread of FLF^* is greater, the right spread of FLF assumes its value from that of FLF^* . The left spread and plateau of FLF , on the other hand, take their respective values from those of FEF . In this case, Y is calculated as

$$Y = (e \ c, e \ c, e \ c, f \ d) \quad (11.15)$$

The comparison of uncertainties is made using the right spreads of the two TFNs being considered. The right spread is the only information available at this time for both FEF and FEF^* . The plateau and left spread of FLF^* are assumed to extend to the infinity.

3. The FLF of the project activities can be calculated as follows:

$$FLF = FLF \oplus Y \quad (11.16)$$

4. Fuzzy late start (FLS) can then be computed by substituting FLF and FD into the following:

$$FLS \oplus FD = FLF \quad (11.17)$$

The foregoing procedure is repeated for all project activities.

11.5 Compatibility between fuzzy events

In addition to the common scheduling results of start and finish times for each activity, information such as: (1) expected project completion time; (2) degree of criticality for project activities; (3) possibilities of meeting specific intermediate milestones; and (4) possibility of violating project constraints, are also of interest to the project team. This information can be used in conjunction with the calculated start and finish times of project activities to provide the project team with the necessary information for assessing the acceptability of the generated schedule. In this regard, two concepts, namely (1) the possibility measure and (2) the agreement index are used to

perform such assessments. Description of these two concepts is given in the following paragraph.

The possibility measure (PM) is used for evaluating the degree of belonging of a fuzzy number to another. In the scheduling context, these two numbers are the calculated and the expected events' times. In fuzzy network calculations, the calculated scheduling event time is usually fuzzy. The expected event time, on the other hand, could either be fuzzy or crisp. Fig., 11.4 (a) and 11.4 (b) show two fuzzy events A and H described by continuous piecewise linear functions. The PM of these two events takes a value equal in magnitude to the maximum membership function of the elements that are common to A to H. Graphically, PM is the highest point in the intersection area produced by the two events being examined [see Figs. 11.4 (a) and 11.4(b)]. Analytically, PM can be calculated using the following steps:

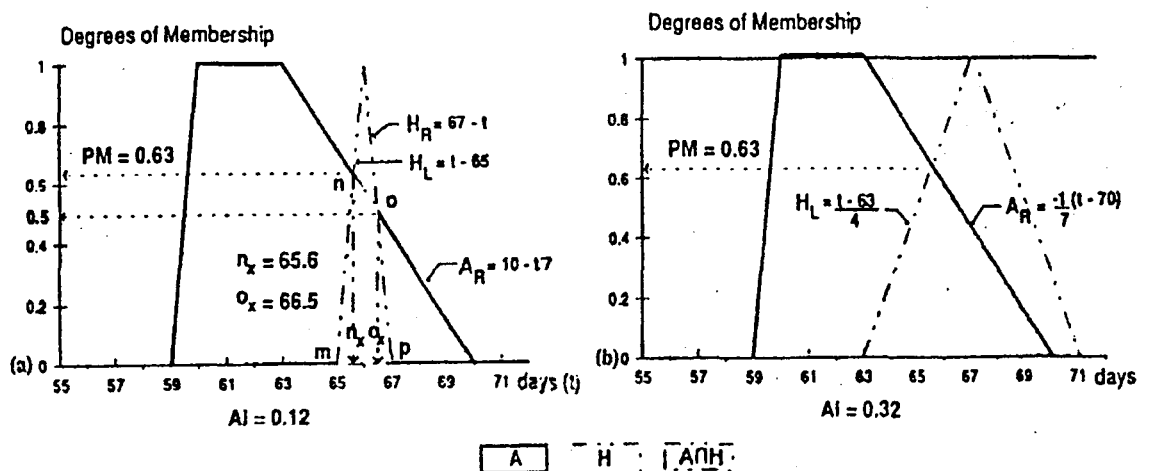


Fig 11.4 Demonstration of possibility measure and agreement index
(a) Less agreement scenario (b) More agreement scenario

1. Identify all points of the intersection generated by the two events using their membership functions [e.g., points n_x and o_x in Fig. 11.4 (a)]. For example, n_x can be calculated using the membership functions H_L and A_R [see Fig. 11.4 (a)] to be 65.6. Similarly, o_x can be computed using H_R and A_R [see Fig. 11.4 (a)] to be 66.5
2. Substitute these points into their respective membership functions to obtain the membership value associated with each intersection point. For example, $\mu_H(t = n_x = 65.6)$ can be calculated, by substituting $t = 65.6$ into H_L , to be 0.63. Similarly, $\mu_H(t = o_x = 66.5)$ is calculated to be 0.50 [see Fig. 11.4 (a)].
3. PM takes the greatest $\mu_{H}(t)$ generated in the previous step. For the situation depicted in Fig. 11.4 (a), PM is set equal to 0.63.

The possibility measure gives only the highest point in the intersection area. No consideration, however, has been given to size of the intersection area. Fig. 11.4(a) and 11.4(b) illustrate an example in which PM alone is inadequate. It can be observed that PM for the two scenarios is identical (i.e., $PM = 0.63$). The events depicted in Fig. 11.4 (b) are, however, more in an agreement with each other than those shown in Fig. 11.4 (a). The possibility measure, in this circumstance, does not provide an insightful assessment of the compatibility between the two fuzzy events. The agreement index measures the ratio of the intersection area between two fuzzy events (i.e., the calculated and expected events) and the entire area of the calculated events. Accordingly, if A and H are two events being considered, the agreement index of A with respect to H, $AI(A, H)$ is defined as

$$AI(A, H) = \frac{(\text{area } A \cap H)}{\text{area } A} \quad (11.18)$$

The intersection area can be determined first by identifying the corner points or the limits [e.g., points m , n , o , and p in Fig. 11.4 (a) and points q , r , and s in Fig. 11.4 (b)] and the equations of the boundary lines [e.g., A_R , H_L , H_R in Fig. 11.4 (a)] for the intersection area. The intersection area can then be calculated using partial integration, considering the function of the boundary lines and their respective limits. For example, the intersection area for the case shown in Fig. 11.4(b) can be calculated as follows:

$$\int_{63}^{65.6} \frac{1}{4} (t - 63) dt + \int_{65.6}^{70} -\frac{1}{7} (t - 70) dt = 2.24$$

the area of the base or reference event (i.e., event A in this case) is simply calculated considering its trapezoidal shape, as 7.0. The AI of the two events shown in Fig. 11.4 (b) is calculated using (11.18) to be 0.32. Similarly, the AI of the events shown in Fig. 11.4 (a) is calculated to be 0.12. Therefore, it can be said that the events shown in Fig. 11.4 (b) are more compatible with each other than those shown in Fig. 11.4 (a). However, the agreement index is not applicable when any of the two being evaluated is an exact event (i.e., the area of an exact event is zero).

11.6 Criticality measurement

When the durations of project activities are estimated with uncertainty, the determination of their respective degree of criticality is not as obvious as in the case

where the durations are estimated with certainty. The concepts of possibility measure and the agreement index can be used to measure the criticality of project activities as follows:

1. Calculate the duration for each and every path in the network.

$$T_x = \sum_{x \in X} d_x \quad (11.19)$$

where x = an activity that belongs to path X .

2. Determine the possibility measure between the duration of each path and that of the project (PM_1)
3. Calculate the agreement index (AI_x) for each path duration and that of the project.

$$AI_x(T_x, T_{proj}) = \frac{\text{area } T_x \cap T_{proj}}{\text{area } T_x} \quad (11.20)$$

4. Activities that belong only to one path assume their criticality from that path. But in activities that are common to more than one path, criticality takes the largest possibility measure and agreement index of those paths.

In addition to the degree of criticality, it is also possible to calculate an approximate total float (TF) for each activity. The approximate total float signifies the possible slack time for an activity. The geometric centroid (C) of a TFN (a, b, c, d) can be calculated as follows:

$$C = \frac{(-a^2 - b^2 + c^2 + d^2 - ab + cd)}{3(-a - b + c + d)} \quad (11.21)$$

The TF of activity x can then be computed as

$$TF_x = C_x^{FLS} - C_x^{FES} \quad (11.22)$$

or $TF_x = C_x^{FLF} - C_x^{FEF} \quad (11.23)$

FUZZY EXPERT SYSTEM FOR NETWORK RESOURCE ALLOCATION

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 12

FUZZY EXPERT SYSTEM FOR NETWORK RESOURCE ALLOCATION

12.1 Introduction

Resource allocation has been a major concern of research and actual application development since the advent of the CPM / PERT network scheduling techniques. Many algorithms have been developed including theoretical optimization and heuristic rules since 1960. The theoretical approaches generate too many combinations and are, therefore quite expensive to use.

The use of heuristic rules [for example, the eight different rules - minimum activity slack, minimum late finish time, resource scheduling method, greatest resource demand, greatest resource utilization, shortest imminent operations, most jobs possible, and random activity relation - provide a reasonable solution to the "too expensive" problem. To date, however none of these algorithms has the capacity to resolve the impact of the variety of external forces (with weather conditions, change orders, changed conditions, resources delays, and resource availability as typical of these forces) that are unavoidable in any real-life project. This might be due to the difficulty in applying the traditional analytical methods, mathematical models, probability theory, or statistical analysis procedure for evaluating the impact of external forces

For a practical algorithm, neither the effect of internal forces (i.e., network related factors, such as free or total float, down stream resource usage, etc.) nor the impact of the external forces should be ignored. Internal forces are usually available from the data output of a network analysis system using CPM / PERT procedure or else can be derived from the network data.

12.2 Fuzzy set application in measuring external impact

In the process of resource allocation, one important issue is the determination of a priority ranking for all activity that are candidates at that instant for resource assignment. This section reviews the basic concept of resource allocation and shows how fuzzy set theory can be used to establish the priority ranking.

12.2.1 Basic concept of resource allocation

The basic concept of resource allocation can be easily explained with the help of Fig.12.1, where α is the candidate set of activities that have all their logical precedence constraints met, and therefore are candidates for resource assignment and initiation of physical work; S is the set of activities scheduled during a time interval; and C is the set of completed activities.

Assume that a list of criteria has been established for allocating limited resource in a specified time frame. Then, the arrow between α and S indicates that each candidate (activity) in α is checked against the list of criteria to see if the criteria are met. If the criteria are met, it is scheduled according to its network constraints. If not, a check is made of the next candidate in set α . In cases where two or more candidates meet the criteria and they require the same resource that are insufficient to start all of them at the same time, priorities must be set before any one of them can be started. Then, each candidate in α will be eventually scheduled according to the priorities. Determining these priorities thus can be a key issue in the process of resource allocation.

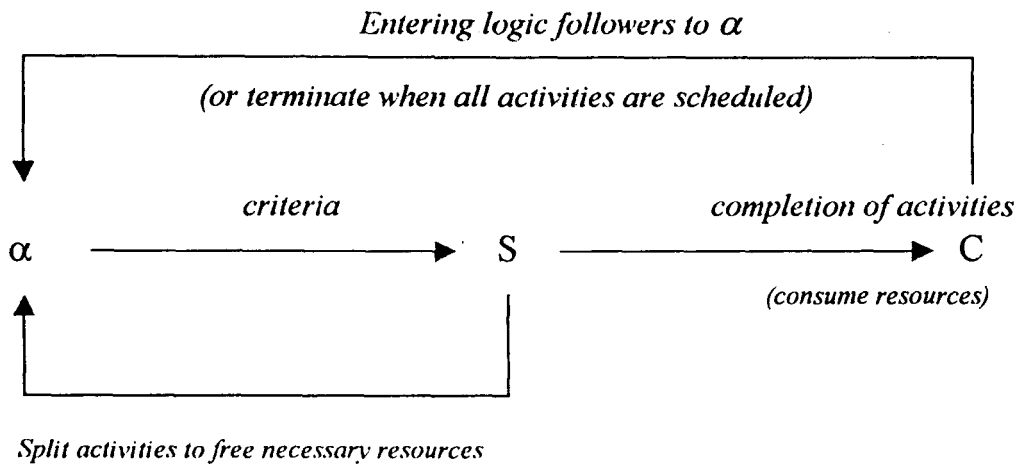


Fig 12.1 Basic concept of resource allocation

The criteria can be categorized into two groups; (1) external criteria (project related), such as weather conditions, resource availabilities, changes, and changed conditions; and (2) internal criteria (schedule related), such as total float, impact of losing float, and the impact of down stream resource requirements.

Assume that n external criteria exist. Let C_i ($i = 1, 2, \dots, n$) denote them. The importance of these criteria depends on time, project location, project characteristics, project manager's preferences, and similar factors. The importance does not simply mean that C_i is either important or not important (what is known as a crisp-set concept). Rather, it can be better treated as a linguistic variable, or as the degree of importance or the weight of importance (fuzzy set concept). Symbolically, let W_i denote the measure of the weight for the importance of criterion C_i . For example, if there exist three criteria C_1 , C_2 , and C_3 , they may be described with criterion C_1 as very important ($w_1 = \text{very high}$), C_2 as more or less important ($w_2 = \text{more or less high}$), and C_3 as less important ($w_3 = \text{low}$).

The possibility theory, which is the extension of fuzzy set theory, can be used here to measure these weights w_i .

From the basic concept of resource allocation, each candidate in α should be checked against the list of criteria to see if the criteria are met. More specifically, this checking determines the susceptibility of each candidate to each criterion. Let s_{ij} denote the susceptibility of activity j to criterion C_i and, let s_{ij} range between zero and one for convenience. So when $s_{ij} = 0$, activity j is not susceptible to C_i . When $s_{ij} =$ any other intermediate value, activity j has some degree of susceptibility to C_i . The susceptibility s_{ij} can be determined using the approximate reasoning concept.

Knowing w_i and s_{ij} , the priority rank P_j for activity j in the candidate set α can be defined as follows:

$$P_j = \frac{\sum_{i=1}^n (1 - s_{ij} \cdot w_i)}{n} \quad (12.1)$$

The term $(1 - s_{ij} \cdot w_i)$ can be interpreted as a measure of activity j 's priority against criterion C_i . For example, assume that a criterion C_i , severity of rain, has been established, and an activity j , placing concrete, will be done in an open area with no protection from rain. Also, assume that there is period during which it rains a lot in the area. Therefore the activity is highly susceptible to rain (s_{ij} is large or close to one) because it is exposed to rain. During the rainy period, the weight of the criterion severity

of rain is high (w_i is high or close to one). Thus, priority to schedule the activity under the probability of impact by rain should be low (w_i is close to zero) during a sunny period.

Table 12. 1. If rain is F_i then it is G_i recommended to consider rain as criterion

Rule (1)	F_i (2)	G_i (3)
1	Heavy	Very strongly
2	Moderate	Strongly
3	Small	Moderately
4	None	Not

Thus, during the sunny period, the priority to schedule the activity should be high. The term $(1 - s_{ij}.w_i)$ reflects these situations and is defined as a relative value to measure the priority against each criterion. Each criterion has its own weight of existence w_i . So, the susceptibility s_{ij} would be diluted by its weights in relation to other criteria. The product of s_{ij} and w_i gives the measure of the dilution and represents the weight susceptibility of activity j to criterion C_i . If there exist n criteria, then the summation of all $(1 - s_{ij}.w_i)$ for $i = 1, 2, \dots, n$ yields a relative value for the priority rank of activity j when all available criteria are taken into account. The priority rank is then normalized by the total number of criteria n to yield a value that is in the range of 0-1.

12.3 Relative weights of criteria

In equation (12.1) w_i can be derived from a set of fuzzy propositions. For example, considering the criteria of rain, temperature, and impact of changed conditions

and changed order, it is not difficult to establish a set of fuzzy production rules as follows:

1. If it is currently raining, then the severity of rain is a very appropriate criterion.
2. If the temperature is low, then the temperature is an appropriate criterion.
3. If the impact of changed conditions is not too serious, then the use of changed conditions as a criterion is somehow appropriate.
4. If the impact of changed order is not too serious, then the use of changed orders as a criterion is somehow appropriate.

Usually, these rules are heuristic and are based on the experience of experts (project team members). A knowledge-based approach is hence a logical extension. The rules listed here can be refined to more detail. For example, the rule regarding severity of rain can be detailed as shown in Table 12.1 or in general, the form of the rules is as follows:

Table 12.2 Probability of rain in specific period

Severity of rain (1)	Probability (2)
Rain is heavy	Not likely
Rain is moderate	Likely
Rain is small	Very unlikely
Rain is none	Unlikely

$$\text{Fuzzy production rule} = \text{if } x \text{ is } F_i \text{ then } Y \text{ is } G_i \quad (12.2)$$

Where x = the name of a criterion, such as rain, temperature, changed conditions, and changed order; Y = a statement in the consequence, such as “the recommendation of

using rain as a criterion": and F_i and G_i = linguistic terms such as *low, moderate, high, heavy, very strong, strong*, that can be consistently manipulated using all fuzzy set theory. These rules can be specified generally for all projects and calibrated specifically for individual projects according to each project's characteristics.

For many external criteria, the probabilities of their occurrence are inherent with their severity. For example, it makes more sense to say that very heavy rain will not be likely next week than to say that rain will not be very heavy next week. So, for the example of the severity of rain, the proposition in Table 12.2 may be established by experience or weather forecast data. Or, in general, this type of proposition has the following form:

$$\text{Probabilistic Fuzzy proposition} = (x \text{ is } F_i) \text{ is } >_i \quad (12.3)$$

Where $>_i$ = linguistic probabilities, such as *likely, not likely, unlikely, very likely*: and x can be the criteria for the resource allocation or anything else with inherent probabilities. Equation (12.3) or Table (12.2) becomes the input to the rule in equation (12.2) or the rules in Table 1 when inferences are performed.

Equations (12.2) and (12.3) form a body of evidence E called type II evidence that is a collection of fuzzy propositions, i.e.,

$$E = (g_1, g_2, \dots, g_n, g'_1, g'_2, \dots, g'_n) \quad (12.4)$$

where g_i and g'_i ($i = 1, 2, \dots, n$) are defined by equations (12.2) and (12.3), respectively. From E , the following can be inferred: what is the degree of possibility that Y is Q ? For example, from the type II evidence of the severity of rain given previously, the following can be inferred: what is the degree of possibility that the recommendation of using the

severity of rain as a criterion is very strong? In this example, Y is the recommendation of using the severity of rain as a criterion, and Q is very strong. For other criteria, such as temperature, changed conditions, change orders, cash shortage, and morale, the same type of inference as previously conveyed can be made. A set of possibilities then can be obtained if we have a knowledge base that consists of the type II evidence regarding these criteria. The set of possibilities can be defined as the relative weight, w_i in equation (12.1), as follows:

For list of external criteria C_i ($i = 1, 2, \dots, n$), the weight w_i is defined as the degree of possibility when the desired level Q is set to an absolute level.

Table 12.3

If activity j tends to be A affected by rain then susceptibility (s_{ij}) of j under criterion of severity of rain is B

Rule (1)	A (2)	B (3)
1	Greatly	High
2	Moderately	Moderate
3	Slightly	Low

For the purpose of solving the construction resource allocation problem, the absolute level in this definition can be set to absolutely strong.



12.4 Susceptibility of activity

Susceptibility, s_{ij} in equation (12.1) is a measure of the sensitivity of an activity j to some external criterion C_i . It can be measured using the concept of fuzzy logic and expert system.

If an activity j tends to be greatly affected by the external criterion C_i , then it is reasonable to believe that the activity is susceptible to the criterion to high degree. For example, assume that a criterion called severity of rain exists, and that an activity j (say, placing concrete) tends to be greatly affected by the rain. Then, by one's judgment or expert advice it is reasonable to believe that activity j is very susceptible to rain. More specifically, this is a type of reasoning based on the following rule:

For activity j , if the degree of impact by C_i is A ,
 then the susceptibility (s_{ij}) of the activity j
 under the criterion of C_i is B

Where A and B can be linguistic terms, such as *high*, *moderate*, and *low*. This type of rule will be referred to as a fuzzy production rule to reflect its unique characteristic of fuzziness. For the example of severity of rain ($C_i = \text{rain}$), the rules may be established by experts as shown in Table 12.3. Note that A in Table 12.3 is the value of degree of impact by rain as defined earlier.

In general, to facilitate the reasoning process it is necessary to translate the expert's knowledge to fuzzy production rules of the following form:

Expert's rules

If X is A_1 then Y is B_1

If X is A_2 then Y is B_2

If X is A_n then Y is B_n

User's input (known): X is A'

Reasoning conclusion: Y is B'

Where X can be, for example, the degree of impact due to rain; Y can be the susceptibility s_{ij} of activity j under the criterion of severity of rain; A_i and B_i ($i = 1, 2, \dots, n$) are fuzzy sets defined to interpret linguistic terms, such as high, moderate, and low; A' is also a fuzzy set, which can be the user's subjective judgment or can be conclusion drawn from other evidence. It need not be equal to any one of the A_i . In theory, it can be any qualitative measurement. B' is the conclusion that is expected from the evidence of expert's rules and user's input. The inference mechanism to conclude that Y is B' will be referred to as *fuzzy modus ponens*, which is an inference logic generalized from the classic modus ponens logic rule.

Table 12.4 If activity j is A_i exposed to rain then degree of impact due rain is B_i

Rule (1)	A_i (2)	B_i (3)
1	Almost entirely	Very high
2	Partially	High
3	Slightly	Moderate

A' can be a conclusion drawn from other evidence. More specifically, this means that the value of linguistic variable X in the expert's rules can be obtained from another set of rules having the same form. For example, the degree of impact by rain for activity j can be concluded from the set of rules in Table 12.4, assuming this set of rules also exists. Consider that activity j , say placing concrete, is entirely exposed to rain. Applying *fuzzy modus ponens* on the knowledge organized in Table 12.4, the degree of impact due to rain can be concluded to be *very high*. Now, *very high* become an input A' to the rules organized in Table 12.3. *Fuzzy modus ponens* can be applied again to conclude that the susceptibility s_{ij} of activity j (placing concrete) under the criterion of impact due to rain is very high. This gives the users more options to select the input information.

In equation (12.1), s_{ij} is a single number (integer or real), not a fuzzy set or linguistic term. However, the *fuzzy modus ponens* always concludes a fuzzy set. Therefore, there is a need to interpret the fuzzy set to produce a plausible number that can represent the fuzzy set and can be used in equation (12.1).

ARTIFICIAL, NEURAL NETWORK

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 13

ARTIFICIAL NEURAL NETWORK

13.1 History of artificial neural networks

Computer scientists were always aware of the fact that in conventional (Von Neuman) computer most of the silicon lies inactive for most of the time and only the CPU and a very small part of the memory is active. The consequence of this state of affairs is that while the digital computer can easily outperform the human beings in computation, it is not able to compete even with a one-year-old boy in recognizing objects, faces, sounds etc. Computer scientists recognized the need for architecture inspired by human brain and hence the recent increased attention on artificial neural network.

The basis of neural modeling in the form considered now may be traced back to the work of McCulloch and Pitts in 1943 (Behnam. M and Y. Q. Zhou, 1994; Chia-Yiu. M and M. A. Shanblatt, 1992; Chinnam. R. B. and Kolarik. W. J., 1997). In 1949, Hebb suggested that the connection between cells which are activated at the same time tend to get strengthened while those which are not activated tend to get weakened – a proposal which had a great impact on learning theory (Jason. P. and Siman. A. R, 1997; Mustafa. K. M. A. and F. Kamoun; Ransing. R. S. and Lewis. R. W, 1997). Rossenblatt introduced a major approach to pattern recognition in his perceptron which could be trained to classify certain patterns as similar or distinct. In 1969, in a book called ‘Perceptron’ Minsky and Pappert criticized the neural network of the day and funding for the area dwindled ^(105, 110). In the next 20 years very little work was done in this area. In the early 1980s, researchers began to take up the study of perception again, perhaps due to

disillusionment with the symbolic processing model of artificial intelligence. The renewed work culminated in the discovery of a learning algorithm for the multi-layer perceptron in 1985 by Rumelhart and others (Shouhong. W, 1997; Yinzhen. L, Kenichi. I, M. Gen and R. Kobuchi, 1997; Zhou. D. N. et. al., 1991). The new approach was to use multiple layer of hidden units to overcome the limitations of earlier perceptrons. In addition, they used nonlinear, continuous and differentiable threshold functions for neuron activations. Over the past few years the interest in ANN has grown rapidly. Professionals from various fields like engineering, philosophy, psychology etc. are intrigued by the great potential offered by this technique and seeking applications in their fields. This interest has been fired by both theoretical and application success.

13.2 Basic concepts

Formally, an artificial neural network can be defined by three elements: a set of processing elements called neurons, a specific topology of weighted interconnections between these elements and a learning law which provides for updating the connection weights.

Neuron: A typical neuron is shown in Fig. 13.1. In this type of neuron, called a perceptron, the output is a nonlinear function of weighted sums of its inputs. The input/output relation is often described as the neuron's transfer function. Some of the commonly used transfer functions are depicted in Fig. 13.2

In Fig. 13.1

x_1, x_2, \dots, x_n are the inputs

w_1, w_2, \dots, w_n are the connection weights.

y is the out put.

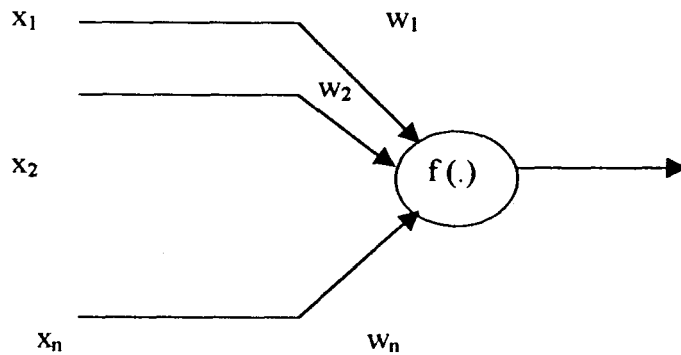


Fig 13.1 Typical model of a neuron

Inter connection topology: there are mainly two classes of network topologies, namely, feed forward and recursive topologies. A typical feed forward network is shown in Fig.13.3. This network has an input layer, two hidden layers and an out put layer. Each out put in a layer is connected to each input in the succeeding layer. A typical set of equations for such architecture may be given by the following equations:

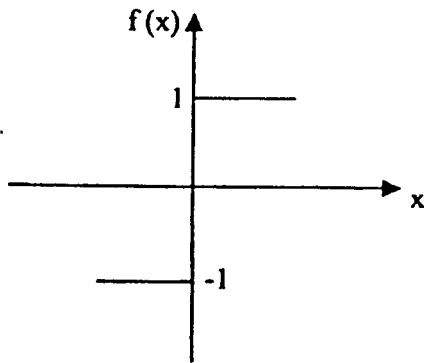
$$P_j = f \sum_{k=1}^n a_{jk} x_k$$

$$q_i = f \sum_{j=1}^m b_{ij} P_j$$

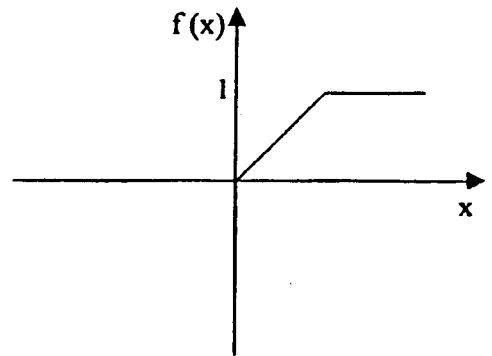
$$y_k = f \sum_{i=1}^m c_{ki} q_i$$

where $f(x) = 1 / (1 + e^{-x})$

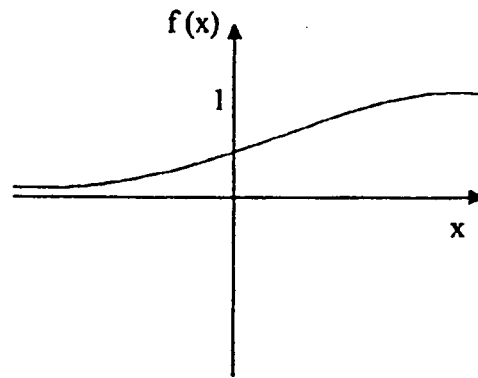
(13.1)



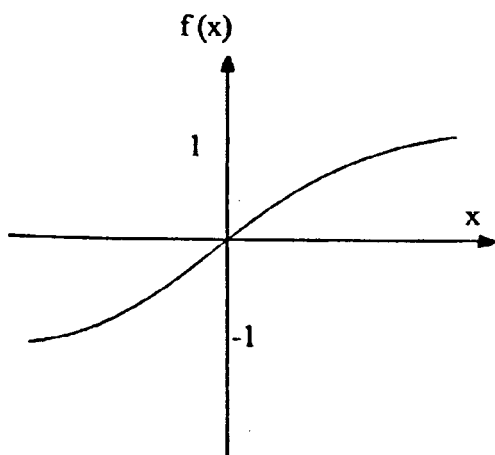
(a) Hard Limiter



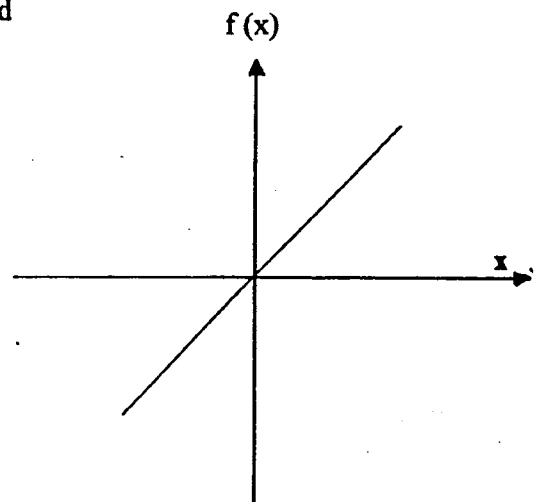
(b) Threshold



(c) Sigmoid



(d) Tan sigmoid



(e) Linear

Fig 13.2 Certain transfer functions

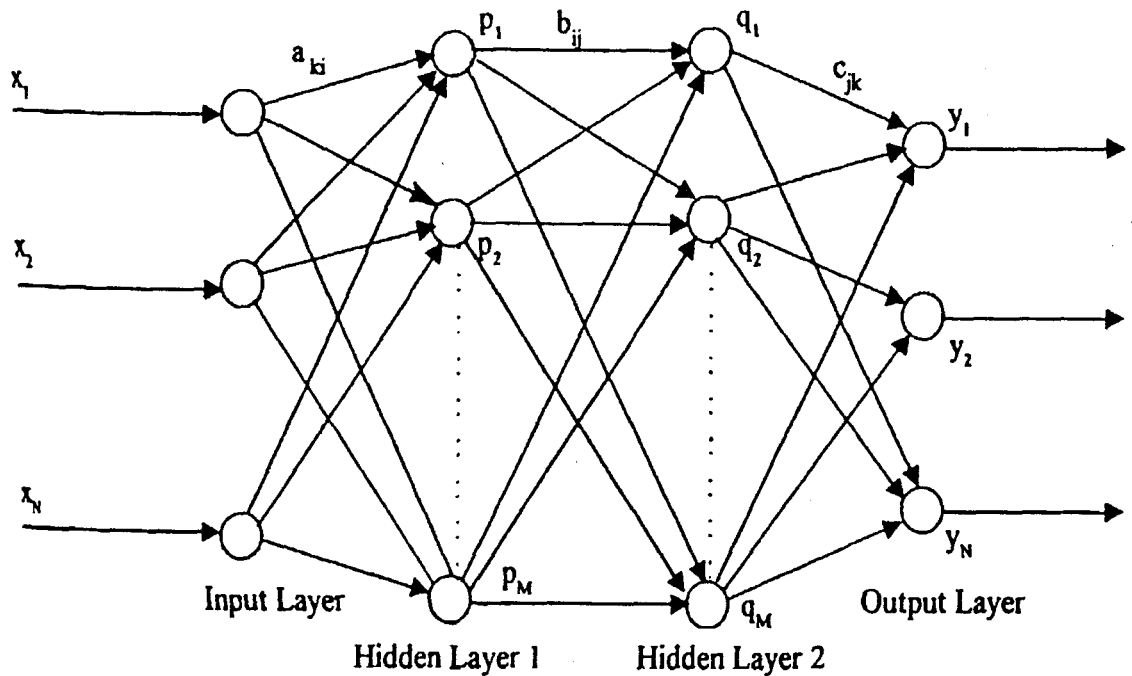


Fig.13.3 Typical feed forward network

For the present case only two hidden layers are considered. In general, the number of hidden layers in the network and the number of neurons in each hidden layer may vary. Also the transfer function of each neuron may change. The effect of feed forward network is to produce a nonlinear mapping between the inputs and outputs. As long as the weights are unchanged this mapping is completely determined. In recursive network topology, each neuron receives as input a weighted output from every other neuron in the network, possibly including itself. Typical recursive architecture is shown in Fig13.4. This system receives an input pattern of x_i s and then interacts according to the equation (13.2). The non-linearity can be a hard limiter or threshold logic function.

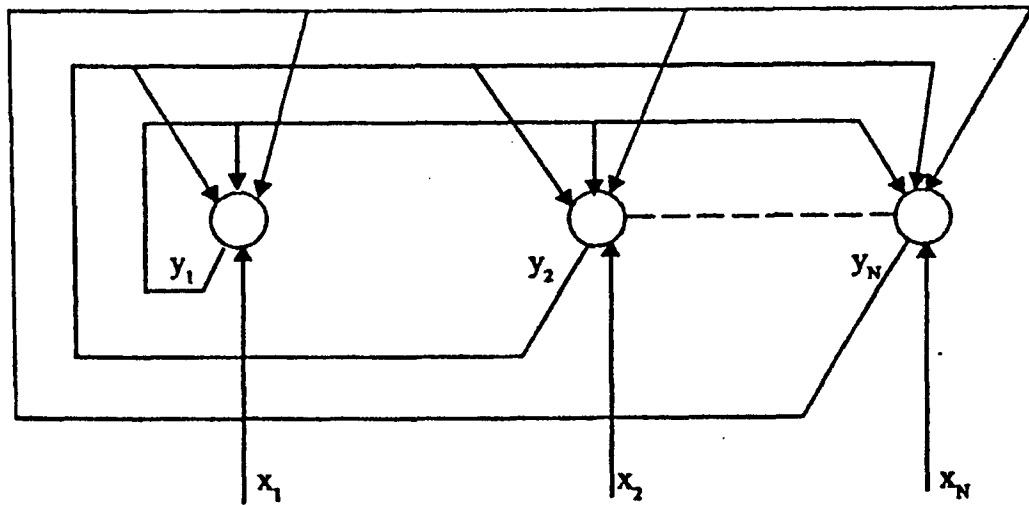


Fig 13.4 Recursive neural network

$$y_i = g \left(\sum_{j=1}^n w_{ij} x_j + y_i \right) \quad (13.2)$$

Learning laws: ANNs are usually used in two ways; operational mode and learning mode. In the operational mode the neural network is used to compute an output in response to a given input. This mode assumes that the interconnection weights have been fixed to some desired values. In the learning mode the objective is to determine the connection weight so that the network output has the desired property. The ANN is first used in the learning mode, with training proceeding until the weights are correctly adjusted for the particular application. Then the network is used in the operation mode. Learning algorithm for weight adjustments can be described either as supervised learning, unsupervised learning or reinforced learning techniques. In supervised learning, it is

assumed that the desired output is known. This is then used to form an error signal, which is used to update the weights. In unsupervised learning the desired output is not known, but learning is based on input/output values. In reinforced learning, weights associated with neurons are not changed proportional to the output error of that particular neuron, but instead are changed in proportion to some type of global reinforcement signal.

13.3 Back-propagation algorithm

The algorithm used for training the multi layered neural network is known as back-propagation. The back propagation (BP) method is at the moment the most effective and widely used learning methods for training multi-layer network. Back-propagation is an approach to adjust the weight associated with the hidden layers, based on the error of the output neurons. To derive the algorithm an error function is defined for the network. Then the partial derivative of the error function is computed. The weight change is then made in the negative direction of the partial derivative. To illustrative this, consider the network shown in Fig 13.3

Let the error functions be defined as

$$E = \frac{1}{2} \sum (y_t^d - y_t)^2 \quad (13.3)$$

Where y_t^d is the element t of the desired output vector y_t^d and y_t is the t^{th} element of the actual output vector.

Normally the hidden neurons and output neurons are provided with bias nodes whose output is unity. The steps to be followed in training are given below:

1. Select training pair from the training set and apply the input vector to the network input.
2. Compute the outputs of the network.
3. Determine the error between the desired and actual outputs.
4. Adjust the weights of connections to minimize the error.
5. Select the next training pair and go to step 2.
6. Continue the steps 1 through 5 until the error for entire set falls below an acceptable value.

13.4 Cost optimization model for a construction project using ANNs

The neural network topology for the neural dynamics construction cost optimization model is shown in Fig. 13.5. The nodes in the network represent the variables and constraints of the problem. The variable layer has N nodes corresponding to the total number of decision variables. The constraint nodes are divided into N_{NT} layers corresponding to non-repetitive tasks, N_{NT} layers corresponding to repetitive tasks, and an initial constraint node. Nodes are grouped within each layer into the constraint categories described in a previous section. Variable and constraint nodes are fully interconnected (interlayer connections). In addition, recurrent and intra-layer connections are also used. Associated with each connection is a weight whose magnitude and sign affect the impulse the connected node will receive. Both excitatory (positive connection weights) and inhibitory (negative connection weights) connections are used in the model. The

coefficients of the constraint functions are assigned to the excitatory connections from the variable layer to the constraint nodes. The gradients of the constraint functions are

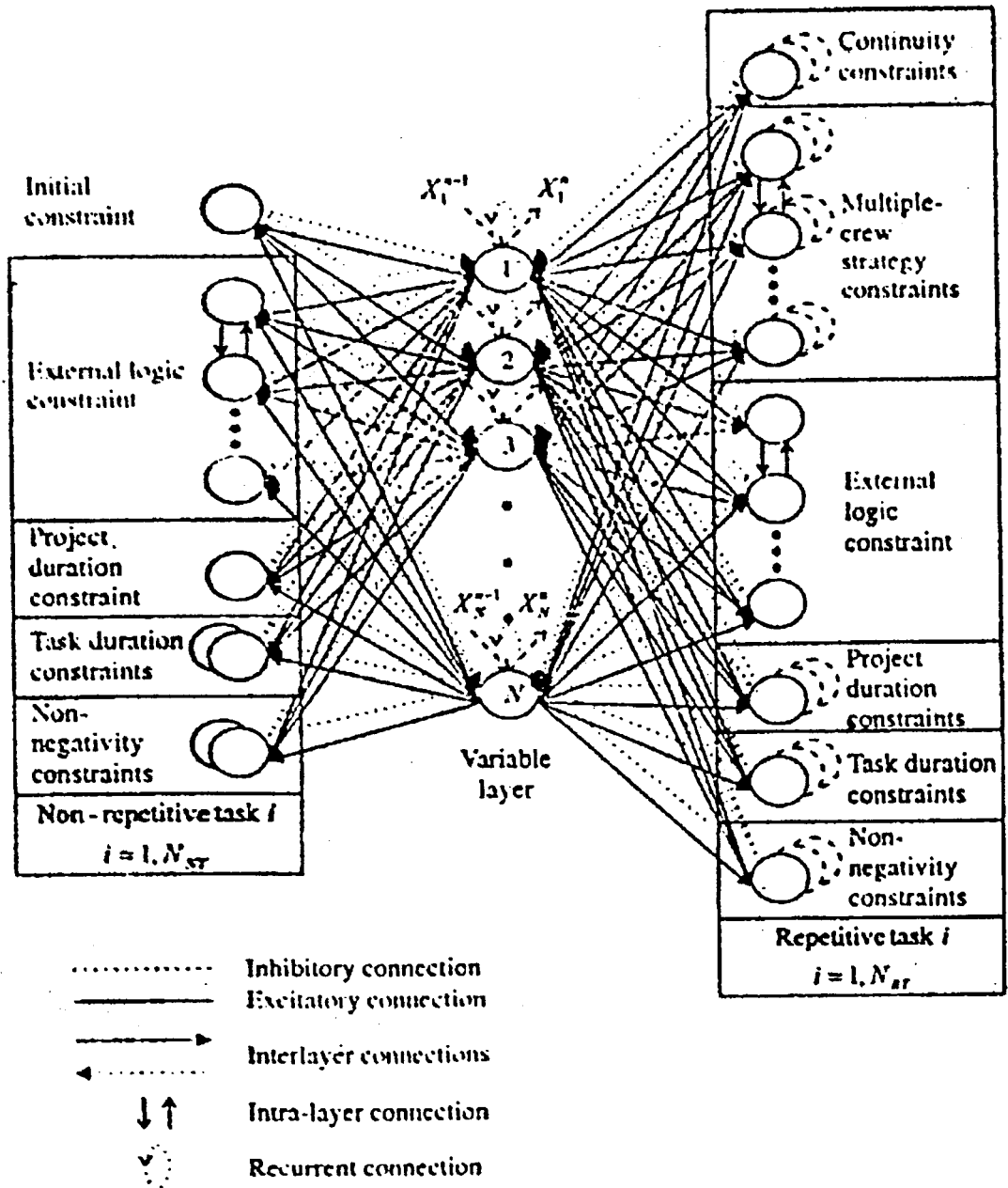


Fig 13.5 Neural network topology for neural dynamics cost optimization model

assigned to the inhibitory connections from the constraint nodes to the variable layer. The gradients of the objective function are assigned to the recurrent inhibitory connections of the variable layer. A weight of one is assigned to the intra-layer connections. This allows the outputs of nodes in competition to be compared.

The output of the variable layer is the current state vector X . Because the coefficients of the constraint functions are encoded in the excitatory connections from the variable layer to the constraint nodes, the input to a constraint node is the magnitude of the constraint at any given state, that is, $g_j(X)$ for an inequality constraint j , and $h_k(X)$ for an equality constraint k . The output of a constraint node will depend on the type of the constraint it represents. For an inequality constraint j , the output is

$$O_{cj} = 0 \quad \text{when } g_j(X) \leq 0 \quad (13.4)$$

$$= r_n g_j(X) \quad \text{when } g_j(X) > 0 \quad (13.5)$$

and for an equality constraint k the output is

$$O_{ck} = 0 \quad \text{when } h_k(X) = 0 \quad (13.6)$$

$$= r_n h_k(X) \quad \text{when } h_k(X) \neq 0 \quad (13.7)$$

Equations (13.4) and (13.5) represent the activation functions. They are chosen such that the output of a constraint node is the penalized constraint violation. When more than one equation is specified for a particular category of constraint, such as external logic constraint, a competition is created between the outputs of the nodes in that group.

For a group of n nodes with outputs $O_{c1}, O_{c2}, \dots, O_{cj}, \dots, O_{cn}$ such that

$$O_{cj} = \max [O_{c1}, O_{c2}, \dots, O_{cj}, \dots, O_{cn}] \quad (13.8)$$

the outputs after competition are as follows:

$$O_{cj} = O_{cj} \quad \text{and} \quad O_{c1}, O_{c2}, \dots, O_{cn} = 0 \quad (13.9)$$

Let w_{ji} and w_{ki} be the connection weight from the j^{th} and k^{th} inequality and equality constraint node, respectively to the i^{th} variable node, and Y_i be the weight of the recurrent connection to a node i in the variable layer. Then, the input to the i^{th} variable node is given by

$$I_{vi} = Y_i + \sum_{j=1}^J w_{ji} O_{cj} + \sum_{k=1}^J w_{ki} O_{ck} \quad (13.10)$$

The new value of the i^{th} decision variable is obtained by the integration

$$X_i^{new} = \int I_{vi} dt \quad (13.11)$$

This integration is done by the Euler or the Runge-Kutta methods. In the construction cost optimization problem, the simple Fourier method also yields accurate results. The network operates until no change in the decision variables occur within a given tolerance, that is, when $X' = 0$, X is the solution to the minimum direct cost construction scheduling problem.

13.5 Application of ANN for prediction of construction labour productivity

During the process of estimating new projects, construction labour productivity is to be predicted in most of the cases. The first step here is to analyze the variety of construction projects and determine which factors that affect the labour productivity should be incorporated in the study. Many factors are to be collected, because it is very difficult to ascertain which of the factors are significant and which are not. The sources of the factors are literature, existing data of participating companies and a survey of the project team. Factors are to be defined as belonging to one of the two categories: quantitative and qualitative. Quantitative data comprise of actual numerical values and qualitative information is mainly ratings and descriptions. Secondly the study is to be conducted to investigate whether there are strong relationship between these factors and productivity. Standard statistical measures scatter plots and correlation statistics are to be compiled. The development of a neural network model for this case is to be done as follows.

1. *Selection of a neural network model-* A number of models are available, including back-propagation, modular, generalized regression, classification and others. This process depends on experience with neural networks and involves trial and error.
2. *Determination of the factors that have significant effect on labour productivity-* These factors form the input to the network.. Obvious ones include the type of work, resources utilized, weather etc.

3. *Determination of the output layer structure*- This determines what the network predict once the input of step 2 is fed into it.
4. *Analysis and scaling of all input and output data to a suitable form*-In general terms, most network structures work best with scaled data.
5. *Determining procedures for maintenance of the experimentation process*- which tends to be data intensive- This step helps in trying various scenarios while keeping track of network performance.

The primary neural network structure is a feed-forward, back-propagation network. Input is to be provided using standard neural network input structures, and the output is a single point estimate of labour productivity. Other network structures included are “modular”, “probabilistic” “generalized regression” and “learning vector quantization (LVQ)”. In many cases the network structure severely restrict what can be inputted and outputted. In others (e.g., generalized regression) the training requires more data than could be possibly collected.

Although estimates from the preliminary neural networks are reasonable, they are deficient in two ways.

1. Failure of predictions are serious and this is unacceptable. In other words, when predictions are wrong, they are significantly wrong. For example, the network could

predict a productivity value of 0.2 m-hrs/sq.m when the actual productivity is 0.7 m-hrs/sq.m (for a range of 0.15 – 1.2)

2. The “one value” estimate (i.e., single point estimate) of labour productivity is not acceptable to estimators, given the fuzzy nature of the problem. In addition, the objective is to improve.

To address these problems a solution is derived based on a network structure with a fuzzy output layer. The suitable neural network is a three-layer back-propagation, feed-forward neural network. The network can use a sigmoid transfer function with a normal cumulative learning rule. There are 55 input nodes representing 30 factors, 30 hidden nodes in one hidden layer, and 13 output nodes. The output formulation of the model is an output scheme that predicts labor productivity on a fuzzy scale (i.e., the productivity is predicted as a set of scores that represent certainty of occurrence corresponding to subset ranges of productivity values). This output is displayed as a frequency distribution, providing a measure of uncertainty associated with the predictions.

13.5.1 Network structure with focus on output layer

The first proposed solution to the problem of determining an output structure is to split the labor productivity values into zones and then allow the network to predict a zone of values rather than a prior estimate, as illustrated in Table 13.1. The zone would contain a sample of productivity values and could be represented by calculated statistics or a fitted distribution. The assumption is that similar productivity values (with in a zone)

would result from similar circumstances. The zone would be predicted and the statistics for the empirical data for the zone would represent the range of possible values. The concept of fuzziness could also be included in the zones, allowing the boundaries of adjacent zones to overlap due to uncertainty. Decisions to be made includes how many zones are to be used (this affects the accuracy of predictions) and, where the zone boundaries would be located.

This model evolved into a "binary zone pattern recognition" method where records are mapped into binary output zones, which represent a small range of productivity values. The first step in the model is to determine the likely range of values. In Fig. 13.6 (a) likely range of value is scaled from 0 to 1. The goal of the model is to predict a subset of the overall range. For example, the subset range could be from a value of 0.10 to 0.40, as shown in Fig. 13.6 (a).

Table 13.1 Productivity values for different zones

Productivity values	Zones
0.10 0.12 0.15	Zone 1
0.20 0.21 0.23 0.25 0.27	Zone 2
- - - -	
0.85 0.89 0.90	Zone

The second step is to determine the number of zones for the analysis, as shown in Fig. 13.6 (b) where the range of all possible values is divided into predefined zone; each zone represents a range of productivity. The third step is to determine the boundaries of the zones and to add an extra zone on the top and bottom of the range of values to ensure that the entire range of possible values is modeled. The fourth step is to mark the records into the productivity zones.

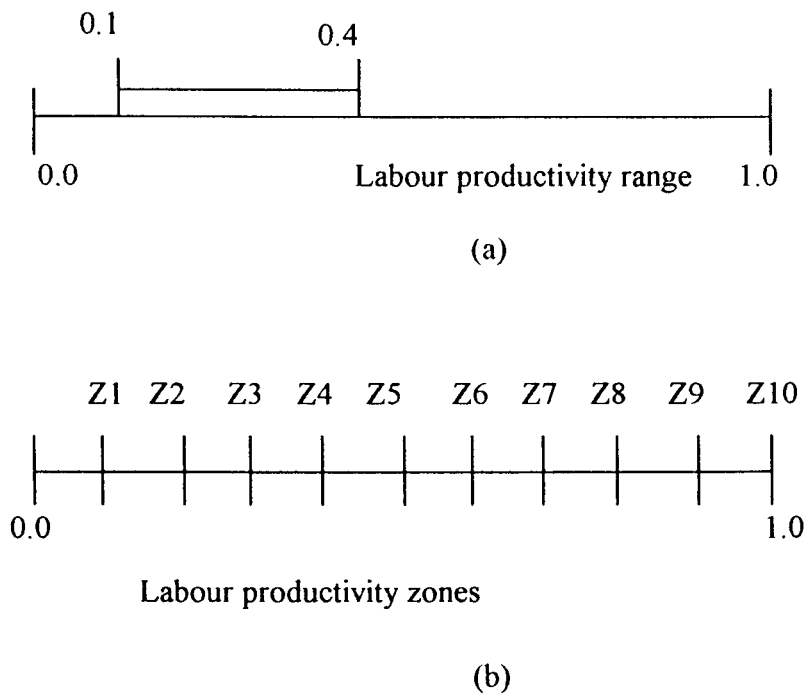


Fig 13.6 Goals of output structure

GENETIC ALGORITHM (GA) APPLICATIONS IN SCHEDULING

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 14

GENETIC ALGORITHM (GA) APPLICATIONS IN SCHEDULING

14.1 Basic concepts

GAs are inspired by the process of natural evolution and the principle of “survival of the fittest”

A typical optimization problem involving GA is

Begin

 Generate a new population of solutions

 While terminating condition not met do {

 Evaluate the solutions

 Select the better solutions

 Recombine solutions using genetic operators

 }

End

As the fragment of pseudo-code shows, the procedure begins with the generation of an initial collection of random solutions. GAs typically works with a collection or population of solutions rather than with a single solution. Each individual solution in a population undergoes the steps of evaluation, selection, and recombination in cycles called generations. An individual may persist across many generations or be replaced in the very next generation depending on its fitness and the generation-gap policy effected by the GA modeler. The GA evolves the population over many generations until a terminating criterion is met. A single string like entity called a chromosome represents each individual solution. A chromosome typically consists of a number of genes, which may be visualized as boxes arranged in a linear fashion. Two attributes are associated

with each gene; its position and its contents, which code for a solution. GAs evolves their populations by recombining partial solutions present in the population. A schema, somewhat analogous to a partial solution, is a singularity template describing a subset of (chromosome) strings with similarity at corresponding gene positions.

1	2	3	4	5	6	7	8	9	10	11
0.70	0.36	0.19	0.24	0.75	0.59	0.85	0.60	0.80	0.02	0.00

Legend:

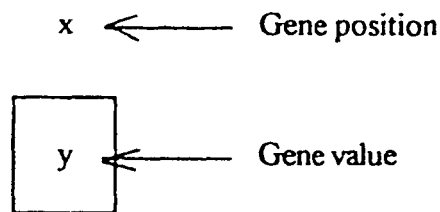


Fig 14.1 Chromosomal representation

For example, 011**001 is an example of a schema that includes all eight place strings having a “011” sequence in the beginning and ending with a “001” sequence. The * is a meta-symbol denoting that the gene can take any value of the (binary) alphabet. Every schema is associated with two properties: its order and length. The order of a schema is defined as the number of string positions that are similar (fixed), and the length is the distance between the first and the last similar (fixed) position. Referring to the foregoing example, the order of this schema is 6 while its length is 7 (8 - 1). Individuals are evaluated quantitatively by applying an objective function on the solution encoded by their chromosomal representation; this is a direct indication of the worth of the solution in

the task environment. The progress of a GA search is often monitored through two performance measures; the best and the average generational performances. Best performance denotes the average of the set of best solutions of each generation while the average generational performance is an aggregate measure of the performance of the population in each generation. Generally, the best performance is a better indicator than the average general performance, and is the measure used in the results presented in literature. Both these measures clearly indicate whether the population is converging to a solution and the degree of diversity that is still present in each generation. Selection is the procedure by which better-than-average solutions are determined for recombination. Among the methods studied in the literature, roulette wheel and ranking are widely used. Essentially, selection places these better chromosomes in an intermediate mating pool wherein the chromosomes are paired randomly. The individuals performing the recombination do not have apriori knowledge of whether the recombination will benefit them individually or collectively, but a famous result known as the schema theorem says that the expected number of a particular partial solution is directly proportional to the ratio of the average performance of individual that contain the particular partial solution to the average of the population as a whole; thus, above-average chance of passing on these schemas to the next generation. An operator called crossover effects recombination; it is the principal mechanism by which the GA arranges for good schemas present on different chromosomes to aggregate on a single individual. Single point crossover involves exchange of a part of each chromosome in a pair across a randomly chosen point. Although crossover is principally thought of as a mechanism that improves the quality of solutions, it is also possible that cross over will disrupt a good schema already

present, especially the long ones. Besides crossover, GAs often apply another operator called mutation. It involves randomly changing the values of certain gene positions, the position and the new value being decided by chance. Mutation is performed at the gene level but not every gene should be mutated. The frequency of mutation is often kept very low to avoid disruption of good solutions; the principal use of mutation is to reintroduce genetic diversity to avoid getting trapped in local optima. It is evident that GAs are probabilistic in nature, from the generation of the initial population to the many operations within the algorithm, which are dependent on chance (implemented through the use of a random number generator). Therefore, several runs of a GA (each called an experiment) should be made to minimize the chance that the results obtained are merely due to the "luck of the draw". This is especially important when evaluating the effectiveness of a particular GA implementation, be it the representation, the GA operators, or the settings of the operational parameters. GAs are not random "generate-and-test" procedures, but incorporate a very effective, though implicit learning element in the algorithm through selection and recombination.

14.2 Scheduling with GAs

In applying GAs to scheduling problems, the issue of the treatment of illegal schedules introduced by the GA operations often comes up. Schedules consisting of ordered lists of activities often produce new schedules with activities duplicated and/or missing or violating precedence constraints as a result of crossover. This can be solved by using a schedule builder (incorporating domain-specific heuristics) to repair the illegal

schedules. This adds a considerable amount of work outside the main GA algorithm and the efficiency of the G-A search is reduced.

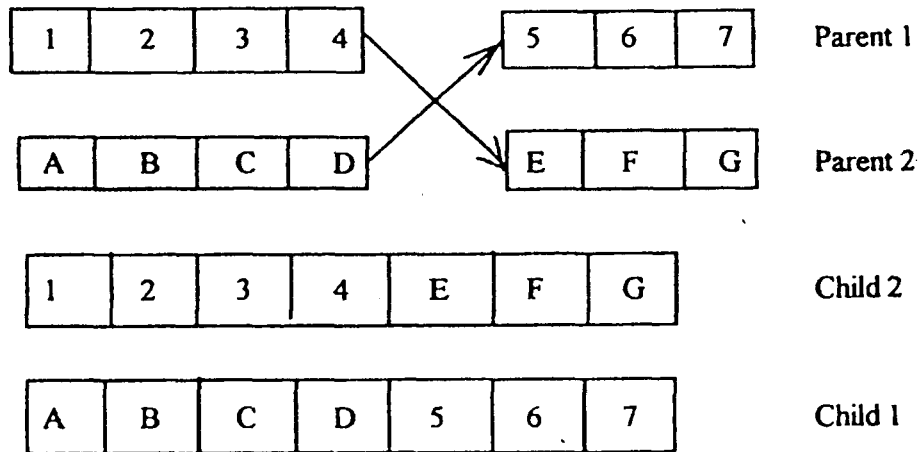


Fig 14.2 Crossover mechanism

Some times, a different chromosome representation can avoid this problem. The concept of sorting random keys can be used to avoid illegal schedules. The keys could be used to represent relative priorities between choices remaining after domain constraints are satisfied. The advantage of using random numbers is that even after recombination, the numbers can be sorted to produce a legal schedule each time. Existing GA theory does not address the problem of infeasible solutions, so ad-hoc work-around that are based on intuitive appeal will distort the feedback to the GA from evaluating its solutions, and the connection between good schema and positive feedback is considerably weakened. The decision variable represented by the genes in the chromosome causes the delay in the starting time of each activity; a binary representation can be used to code the gene values.

14.3 The GA-scheduler

14.3.1 *Structure of GA-scheduler*

By hybridizing the GA with existing problem solving algorithms, the domain expertise embodied in the encoding used by the current algorithm can be preserved. This is reflected in the design of the GA-scheduler. Fig 14.3 depicts the main components of the GA-scheduler implemented, comprising the basic GA engine, a schedule builder, and an evaluation function. The GA engine performs the basic GA processes of selection, recombination and mutation on succeeding populations of solutions. Genesis is a commercially available GA package, which can be used to provide the routines of the GA engine. Generally, Genesis users only have to provide an application-specific evaluation function and link the evaluation function with the GA routines supplied in the package to make a complete application-specific GA program. However, for the scheduling problem, the evaluation function in equation (14.2 (c)) has to be supplemented with a schedule builder. Various aspects of the GA-scheduler are discussed under the appropriate headings.

14.3.2 *Schedule builder*

The schedule builder contains the logic to construct a valid schedule from the decoded chromosome according to the constraints of the application, namely;

1. *Hard constraints*: these constraints can never be violated or relaxed at any cost. Precedence relationships between project activities are prime examples of hard constraints. It is advantageous to code the logic to maintain hard constraints directly into the schedule builder for efficiency reasons.

2. *Soft constraints*: these constraints can be relaxed to a certain extent but with a penalty on performance. The model treats project duration and resource availability as soft constraints. Project durations can be extended or resources borrowed indicating that, at least at the planning stages, these should be treated as decision variables. However, these decision variables are situated at a level higher than those encoded at the gene level, which involve activity - level decisions such as the start of activities. Soft constraints are incorporated into the evaluation function

14.3.3 Chromosome representation

These issues are involved in choosing a suitable chromosome representation; the choice of the decision variable being coded, the mapping from activity to gene position on the chromosome, and the form of coding to be used for gene values. An improper choice of the decision variable causes a lot of ad-hoc rework and weakens the link between good schema and positive feedback

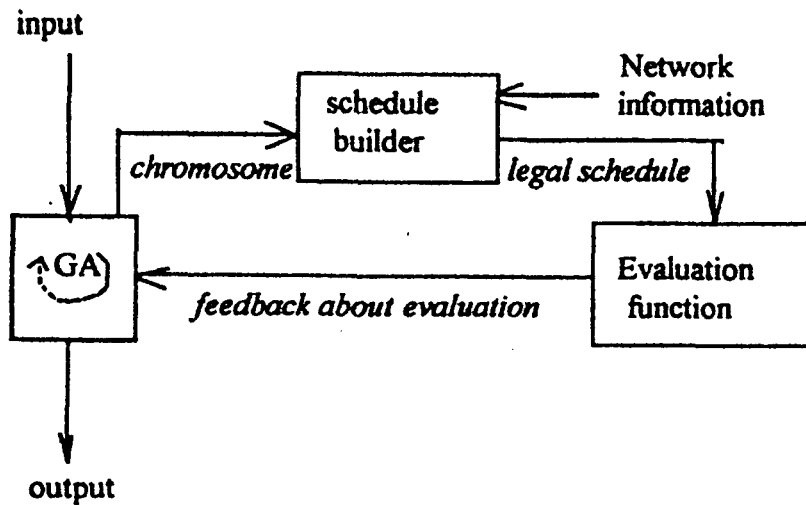


Fig 14.3 GA – Scheduler builder – Evaluation function interface

In construction scheduling, two alternatives suggest themselves: the genes could represent either the start times of activities or the priority with which to schedule activities. If the first alternative is chosen, the precedence relation is often violated, thus requiring extensive external repair. This is because, in this alternative, the precedence relationship between the activities is not encoded in the chromosome and the GA has to learn through exploration with the start times (which is clearly inefficient). Instead, the concept of random keys can be adopted. Accordingly, the gene values are real numbers, which code for two decisions; the scheduling priority of an activity, and the percentage of current float to be utilized before the start of the activity.

The need to consider the activity-to-gene mapping arises because a chromosome is a linear string of genes, whereas a project network, arguably, has some form of

structure, principally induced by the precedence relationships between the activities. It would be straight forward to ignore the precedence structure and use a simple (and arbitrary) mapping of activity to gene position, in such a simple mapping, the first activity would occupy the first gene position and so on. However, such a simple mapping would not perform as well as a mapping based on topological sorting of the activities. In this case, activities are first stored using precedence relationships as the criterion, activities of the same topological rank are then mapped to a contiguous segment of the chromosome. This means that an activity's gene position no longer corresponds to its activity number. This scheme performs better than the simple linear mapping since, in the latter, activity schema (which are expected to be associated with topological rank) would have longer defining lengths and thus become more susceptible to disruption by crossover.

Two widely used forms of coding for gene values are the binary and floating-point representations. GA traditionalists prefer binary coding, arguing that the binary representation explores the most number of schema per trial. With binary alphabets, the parameter range is discretized with enough resolution to enable the output to be affected to the desired level of precision. The discretized value is then represented as a bit string consisting of an appropriate number of 1 and 0 bits. However, coding becomes more difficult when the number of values is not a power of 2. For example, if only 25 values of a parameter are allowed, at least 5 bits are required to cover this range; but this gives a total of 32 slots, of which seven may not have any evaluation. It is problematic to decide how to handle these extra slots since any assignment of them to valid values will bias the

search in an unpredictable way. The use of floating point representation has been a more recent development without much theoretical underpinning, but it has gained increasing acceptance among GA practitioners as it has been shown to work just as effectively as binary coding, and sometimes makes the coding more natural to the application.

14.3.4 Interpretation of the chromosome

In the GA-scheduler, each gene in the chromosome codes for the scheduling priority and delay in the start time of an activity. The precedence constraints are maintained by constructing the schedule rank-wise. An activity with a higher gene value will be put in the scheduling queue ahead of others with lower gene values in the same rank. The actual start time of an activity is not its earliest start, but is determined by adding a fraction of the current float to the earliest start of the activity. The current float (CF) of an activity is defined as

$$CF = LFT - CT - d \quad (14.1)$$

where LFT = late finish time of an activity; CT = current time; and d = remaining duration of an activity. CT , in this model, is defined as the minimum of the finish times of the activities of the previous rank. The start time, ST , of an activity is then calculated as

$$ST = FT_p + \text{random number} \times CF_p \quad (14.2)$$

where FT = maximum of the finish time of the predecessors of the activity.

The use of random keys and a floating point representation enabled solutions to be found for several difficult representations and coding issues. However, their use is not without its cost, since they impose an overhead during the search of schema by the GA to understand why it is necessary to distinguish between genotypes (the list of real numbers encoding priority rankings and current float usage produced by the GA) and phenotypes (the activity schedule instances produced by the schedule builder after interpreting the list of real numbers). When the scheduler uses the real numbers to determine priority, it is only interested in their ordinal, not absolute, values; hence many combinations of real numbers could lead to the same ranking. More significantly, the scheduling delay, calculated as a percentage of current floats, is a multiple of the basic scheduling unit. Some rounding of the result of the multiplication between the gene value and the current float is necessary when determining the delay; depending on the magnitude of the float, several real numbers could map to the same amount of delay. Furthermore, a set of activity schedules could produce the same resource utilization profile, which is evaluated by the objective function. Thus, several genotypes may correspond to one schedule (phenotype), and several schedules may yield the same profile.

14.4 Resource scheduling with genetic algorithms

Early attempts to solve resource-scheduling problems used mathematical models, but their primary disadvantage was that they could not solve the bigger and more complicated problems encountered in practice. Later attempts used heuristic rules to overcome the problem of combinatorial explosion, but the rules themselves proved to be

very much problem dependent. Furthermore, algorithms designed for resource allocation may not perform well on resource leveling problems. A new approach, employing the use of genetic algorithms (GAs) can overcome these drawbacks and provide the basis for a general method of resource scheduling in construction projects. The nature of the solution discovery process used in the GA approach recommends itself to actual everyday application since several good (not necessarily optimum) solutions will be returned by the GA, often under strict time pressure.

Traditionally resource-scheduling problems in construction project networks have been solved either as a resource leveling or as a resource allocation problem. The objective in the resource-leveling problem is to reduce peak resource requirements and smooth out period-to-period assignments within the required project duration, with the premise of unlimited resource availability. The resource allocation problem starts off with the assumption that resource availability is constrained to some maximum value and the objective is to allocate the available resources to project activities in an attempt to find the shortest project duration. Each class of solution has its own specialized solution algorithm that takes into account the objectives, constraints and premises of the problem. The division is to facilitate ease of solution of what would otherwise be a difficult combinatorial optimization problem. In practice, the distinction between the two problem types is not so clear since a decision maker is likely to desire both objectives of resource leveling and minimum project duration, and will certainly consider a trade-off between increasing resource availability and the prospect of shorter project durations. Early attempts to solve these problems used mathematical models such as linear programming

and dynamic programming to obtain an optimal solution. The efficiency of the algorithms in searching for solutions depended on making strong assumptions about the objective functions and constraints employed in the model, which may depart from real-world situations. Furthermore, this approach scaled poorly on larger problems required in practical application due to a phenomenon called “combinatorial explosions”. Other attempts used heuristic rules to overcome the problem of combinatorial explosion. A single rule or a hierarchy of rules (to break ties) is used to decide the order of resource allocation among competing activities. These rules have been shown to perform well over a variety of problems and are widely used in actual practice because of their simplicity and efficiency in application. However, some studies ⁽⁴⁷⁾ have indicated that it is not possible to determine apriori the best set of heuristic rules for a particular problem. Further, a significant implication of the use of heuristic rules is that the optimality of the solutions obtained is not guaranteed, although certain heuristic rules produce very good feasible solutions. It has been reported that the minimum slack heuristic performed well over a wide range of problems. Computationally, this method has the drawback that the network has to be recalculated every time an activity is delayed.

In the ensuing discussions a new approach using GAs to overcome some of the previously mentioned difficulties associated with mathematical and heuristic methods are incorporated. The GA model schedules the starts of activities in a single project (and hence performs resource allocation) using a serial model of allocation, with the objective of minimizing the difference between resource availability and utilization. An ancillary objective is to minimize the project duration under limited resource constraints. By taking

the view that the objective is minimization of the deviation of that required from available resources, the model is able to handle both resource allocation and leveling as well as “due-date” problems without extensive modifications. The key to success in the GA application will lie in the design of the problem representation, good interfacing with the task environment, and determining suitable GA operating parameters. These should be followed by the implementation of an actual GA scheduler and comparison of the results of the scheduler with that of heuristic methods.

Problem definition

The case in view is a multi-resource scheduling problem whose objective is to minimize the deviation of required resources from the available resource profiles subjected to the precedence relationship among project activities

$$\text{minimize } E = \sum_{k=1}^n Rd_k w_k \quad [14.3 (a)]$$

subject to

$$S_x \geq \max_{y + P_x} \{S_y + d_y\} \quad [14.3 (b)]$$

where E and Rd_k = deviations of required resources from the available resource for all n resources and the k th resource respectively; w_k = weightage factor for the k th resource; S_x and S_y = start times of activities x and y respectively; d_y = duration of activity; and

P_x = set of activities which must precede activity x .

The deviation for any one resource (dropping subscript k) is given by

$$Rd = \sum_i Rd_i = \sum (Ra_i - Rr) \quad [14.4 (a)]$$

where

$$Rr_i = \sum_{j=1}^A Rr_{ij} \quad [14.4 (b)]$$

in which Rd_i = difference between available and required resource on day i , Ra_i = resource available on day i , Rr_i = resource required on day i ; Rr_{ij} = resource required on day i by the j th activity and $j + A =$ set of all the activities scheduled on day i . The objective function of (14.3) can be further extended to consider the underutilization and over-utilization of resources and extension of project duration. Thus, if T_a is the target project duration for which the resource is planned and T is the actual project duration, the contribution of any one resource to the objective is

$$\begin{aligned} \sum_{i=1}^r Rd_i &= k_1 \left[\sum_{i=1}^{T_a} (Ra_i - \sum_{j=1}^A Rr_{ij}) \right] \\ &+ k_2 \left[\sum_{i=1}^{T_a} (\sum_{j=1}^A Rr_{ij} - Ra_i) \right] \\ &+ k_3 \left[\sum_{i=T_a+1}^T (\sum_{j=1}^A Rr_{ij} - Ra_i) \right] \quad [14.4(c)] \end{aligned}$$

where k_1, k_2, k_3 = penalty factors (in which k_1 and k_2 are meant to apply to mutually exclusive situations of underutilization and over-utilization, respectively). Resource leveling and limited resource allocation problems can be modeled by setting appropriate coefficients for the first two terms of the equation, while due date problems would be modeled by penalizing resource scheduling beyond the target date [as represented by the last term of [14.4(c)]].

COMPUTER SOFTWARE IN PROJECT SCHEDULING

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 15

COMPUTER SOFTWARE IN PROJECT SCHEDULING

15.1 Introduction

The project scheduling techniques have gained momentum in the fields of application after the advent of the personal computers. A good number of computer software emerged, but only a few of them could withstand the test of time. *Primavera (with project planner)* and *Microsoft Project* are two such software packages, widely used now – a - days. It is attempted here to bring forth the facilities of these packages, with their advantages and disadvantages and a critical analysis of the future potential for software packages

15.2 Primavera project planner

The version studied by the author is the Primavera 3 (P3), release 2 of 1997. (P3 2.0 release). The major features and facilities available in this package are discussed hereafter.

15.2.1 Features available in the P3 2.0b release.

The major utilities available are, customizing standard reports and graphics for new projects (including Gantt/PERT charts), using Microsoft Access 2.0 with P3 for Windows, using Lotus 1-2-3 with P3 for Windows, Creating Custom P3 for Windows Reports Using IQ for Windows, using Excel 5.0 and Word 6.0 with P3, using Lotus Approach with P3 for Windows, creating custom P3 for Windows reports using crystal reports, customizing curves, custom data items and specifications, total float calculations with multiple calendars, showing expected finish constraints on bars, creating free float bars, copying activities from one project to another, printing bar chart layouts and graphic reports to a file, displaying start/finish triangles on bar chart graphics and converting

Microsoft Project data. In addition to these facilities, the following special features are also available in the latest Primavera package.

Non-exclusive scheduling: Exclusive access to schedule a project is no longer needed, whether it is a stand-alone project, a project group, or a member project. When we initiate scheduling, if another user is working in the same project, or a member project of the same project group, P3 displays an advisory message. We can continue scheduling. When scheduling completes, P3 stores new schedule dates for activities that are not locked by other users.

Updated RA Primavera Engine and OLE Automation Server: P3 2.0b includes the most recent versions of the RA Primavera Engine and OLE Automation Server. These updates enhance the interoperability of Primavera products that use RA.

New RA scripts: The update also includes the following new RA scripts that provide additional ways to work with P3 using Excel spreadsheets as the working environment.

ACTIVITY.XLS: Run ACTIVITY.XLS to add or modify P3 activity data in Excel spreadsheets rather than the P3 Activity form. All fields available with the Activity form are available in the spreadsheet.

CUSTOM.XLS: Run CUSTOM.XLS to create a P3 resource or cost control report in an Excel spreadsheet.

PREDSUCC.XLS: Run PREDSUCC.XLS to produce a filtered or unfiltered predecessor/successor report in an Excel spreadsheet.

PROJECT.XLS: Run PROJECT.XLS to produce a project's Activity Code, Resource, WBS, and/or Custom Data Item dictionaries in Excel spreadsheet format. Editing a spreadsheet has the same effect as making edits directly to the matching P3 project dictionary. We can also use this macro to transfer dictionaries between projects.

SPREAD.XLS: Run SPREAD.XLS to produce an Excel histogram showing resource spread. Depict the use of one or all resources in a project.

A utility for updating progress: Use this new Visual Basic script to automatically progress or remove progress from activities. Open a project and specify a new data date, either before or after the current data date. Update Progress adds or removes progress for activities scheduled to work during that timeframe and updates both driving and non-driving resources (This script provides the functionality of Sure Trak's Tools).

Project check-in/check-out: Team members can check member projects in and out of project groups. This utility enables to manage version control.

Open Project Check-in/Check-out by choosing Tools, Project Check-in/Check-out.

Display PERT connector activities as ovals or activity boxes: A new option appears in the PERT activity box configuration dialog box. If we show connectors to excluded activities, mark the display connectors as ovals checkbox to display connector activities as ovals containing the activity ID only. Clear this checkbox to display connector activities as activity boxes with circular endpoints.

Import and export activity logs with MPX Convert: MPX Convert, the utility program that converts P3 project data to and from Microsoft Project format, exports the first 10

log lines of P3 activities to the Microsoft Project notes field, and imports the first 10 lines of the Microsoft Project notes field to P3 as the first 10 log lines.

Summarizing projects with different planning units: P3 enables to summarize projects with mixed planning units. P3 uses the planning unit for the summary project to determine the overall planning unit. For example, if we want to summarize a daily and a weekly project into a daily project, the activity durations from the daily project remain in units of days. P3 calculates the activity durations from the weekly project in days as the numbers of work periods between the summary early start date and the summary early finish date. P3 uses the calendar assigned to the summary activity to determine the available work periods. For example, an activity that has duration of 4 weeks in a weekly project has duration of 20 days if it is summarized in a daily project, based on a project-group calendar of 5 workdays per week. If we summarize a daily and a weekly project into a weekly summary project, P3 uses units of weeks for all durations. All start dates are pushed back to the nearest start day of the week, and all finish dates are moved to the nearest end day of the week. For example, a summary activity with an actual start date of Tuesday in a daily project has an actual start date of the previous Monday if summarized into a weekly project, assuming that Monday is the specified start day of the week in the project-group calendar. P3 uses the summary early start dates and the summary early finish dates to determine the duration of a summary activity. Because P3 moves start dates to the beginning of the week and finish dates to the end of the week, activities from a daily project that are summarized into a weekly project can never have durations of less than one week. P3 always expresses data for a resource in terms of the project-planning unit for the resource. If we summarize projects with different planning units and then load detailed resources, the summarized resource quantities still reflect the planning unit of the source-project resource. If we consolidate, resource use reflects the planning unit of the summary activity.

Consolidating by resource, cost account, or cost category: We can summarize projects to consolidate resource and cost data while leaving the source projects intact.

Consolidate by resource: P3 consolidates resources for a summary activity by analyzing all the detailed activities in the summary group and totaling the quantities for all matching resources. Matching resources are any resources that have the same name, even if their resource designators, cost accounts, or cost-account categories differ. P3 adds the values for the budget, the quantity at completion, the actual quantity to date, and the actual quantity in this period. The new units per time period and quantity to complete are calculated as

Quantity to complete = Summary quantity at completion - summary actual to date

Units per time period = Summary quantity to complete, summary remaining duration

P3 assigns the resources from all the detailed activities in a summary group to the summary activity with the appropriate quantities. P3 assumes that the summary resource requirements start when the summary activity starts and that they finish on the finish date of the summary activity. P3 does not consider resource lags and durations from the source project. P3 does not assign resource designators, cost accounts, and cost-account categories to the summary activity when we consolidate by resource. During the summary process, P3 adds resources to the resource dictionary for the summary project as needed. If a resource that is used in a source project does not exist in the summary project, P3 adds the new resource to the summary project, together with its availability limits and unit prices. Not consolidating by resource may result in a large number of resource assignments for each summary activity. For example, suppose a summary activity is compiled from 10 detailed activities and each uses the same resource

(distinguished in each activity by a resource designator). A detailed analysis results in a summary activity with 10 resource assignments, RESOURCE 1, RESOURCE 2, ... RESOURCE 10. A resource consolidation shows that the summary activity has only one resource (RESOURCE), which represents the total use for all 10 resources.

Consolidate by cost account: A summary of costs by resource is the same as a summary of quantities by resource except that it shows accumulated values of currency instead of units. A cost consolidation by cost account shows only cost accounts and their associated costs. Resources or cost-account categories are not shown. A consolidation by cost account for resource data shows only cost accounts (and not resource names or cost-account categories); resource data are not loaded because P3 requires a resource name to show quantities.

Consolidate by cost category: A summary consolidated by cost category accumulates costs for each category encountered in the group of detailed activities for a summary activity. P3 does not show resources or cost accounts in the summary activities. In place of the resource name in the summary activity, P3 shows the cost-category title. Cost estimates from the source-project activities are also added.

If we update a summary project by summarizing again, and load resources or costs, P3 bases resource/cost data on the current source project(s). P3 replaces existing resource/cost data in the summary project that are not updated with zeros.

15.2.2 *Using the Primavera project planner*

Primavera supplies a default layout, the classic schedule layout, which is displayed when we open a project the first time, add a new project, or create a new layout. The classic schedule layout for the bar chart view includes the activity columns

and bar chart. The activity columns show the activity ID, activity description, original and remaining duration, percent complete, early dates, resources, and budgeted cost. The bar chart contains activity bars for early dates and float, and the timescale spans about four months of the project. Activities are organized by early start dates and total float. The layout for the PERT view shows all activities in the project, starting with the first activity that has no predecessors and ending with the last activity that has no successors. Activities are represented as boxes that show the ID, description, early start and finish dates, original and remaining durations, and total float.

Creating a new layout: We can create an unlimited number of layouts for each project. P3 displays the default layout when we create a new layout. Tailor the layout to our liking, then save it with a unique ID and name. A project group and its projects share one set of layouts.

Saving a layout: P3 automatically saves changes to project data such as adding, deleting, editing, copying, and updating activities in a project. To save changes to the appearance of a layout, such as displaying the Activity form, changing column data, modifying bar sizes and colors, and modifying PERT activities, choose View, Layout, Save. We can also save the layout as the default, which P3 uses each time we add a new project or create a new layout.

The following can be saved in a layout: bar chart view, PERT view, common elements, activity table data and settings, attached and embedded objects, bar and endpoint configurations, bar colors, column titles, font, font definitions, hard page breaks, organization and grouping options, print options, progress bar, progress definition, relationships display, resource/cost profile or table display and settings, rolling dates, row height, screen colors, separation lines between groups and column data, sight lines, Split-

bar positions, summarization settings, timescale settings, trace logic displayed, trace logic settings, WBS outline options, activity box configuration and templates, activity positions, activity shape and color settings, activity shape and color settings for new activities, arrange options, connector block settings, cosmic view displayed, cosmic view position, critical color for activity boxes, end width percentage for activity boxes, print options, progress displayed, relationship format, split bar positions, trace logic options, zoom levels for PERT and trace logic, activity form and detail windows displayed, critical float definition, date format, filter assignment, layout options and toolbar displayed.

To open a layout: P3 opens only one layout at a time for each project. If we share projects with other users, P3 can copy a layout if two or more users need to access it at one time. We must save the copied layout as a unique layout to save our changes. P3 remembers the last layout we opened for the last 25 projects. We can change the current layout while we have a project open. P3 displays the layout in the Bar chart or PERT view, depending on the view opened when we last saved the layout. P3 also runs the layout's associated filter and, if displaying the bar chart view, organizes activities based on the bar chart group and sort criteria. If P3 displays the PERT view, P3 arranges activities based on their relationships and places activities without relationships at the bottom of the network.

Replacing the default layout: Instead of using the classic schedule layout as the default layout, we can create our own default by using the View, Layout, Make Default commands. Once we assign a new default layout, P3 displays this layout each time we create a new layout or add a project.

Snapshot activity list: This layout option is available only from the bar chart view and enables to save the order and grouping of activities in a layout. When we save the layout as a snapshot activity list, P3 creates a list of the activities in the layout. The next time when opening the layout, P3 displays the activities in the snapshot activity list and does not run a filter or reorganize the project. Using a snapshot activity list enables P3 to open projects faster.

Switching to the PERT view when working with a snapshot activity list: The snapshot activity list applies only to the bar chart view. If we switch to the PERT view, P3 displays a message warning that we will lose the snapshot activity list and prompts to continue. If yes is chosen, P3 changes the layout options to run the filter and replace the current set of activities. P3 displays the PERT view and runs the layout's associated filter. Since running the filter also reorganizes activities, we lose the snapshot activity list. If we switch back to the bar chart view, we can see activities based on the layout's associated filter and the current grouping and sorting options. Click No if we want to continue working in the bar chart view with the snapshot activity list.

Multi-user considerations: A snapshot activity list is available only to the user who creates the list; however, we can still open a layout with the layout options set to snapshot activity list. If P3 cannot locate or access the snapshot activity list when we open a layout, P3 reorganizes the layout using the current filter. After we open the layout, keep the layout options set to snapshot activity list and P3 will create a new list when we save the layout; or change the layout options to organize and P3 will open the project accordingly.

Adding activities: When adding activities, P3 automatically assigns an activity ID. P3 assigns a duration of one planning unit, which we can change. P3 automatically sets the

remaining duration equal to the original duration. P3 calculates activity duration based on the type of activity. For some types of activities, we can enter a duration; for other types of activities, P3 calculates a duration. If the layout is grouped and we add an activity to a group, P3 automatically applies the group values to the new activity.

Assigning activity IDs: Activity IDs can consist of up to 10 characters: letters, numbers, or a combination of both. Each activity ID must be unique.

Fragnets overview: Fragnets are sections of a network that can be used to build new projects quickly. Fragnets can be created using data from our own projects or retrieve sample fragnets provided with P3. P3 stores the following data for activities in a fragnet: Activity ID and description, durations, percent complete, float, relationships for activities within the fragnet, logs, resources and costs (and associated dictionary information), constraints, WBS values, custom data items, activity code values and activity type.

Activity and resource calendars: P3 provides base calendars, resource calendars, and a global calendar. Each project has a global calendar and at least one base calendar, calendar 1. The global calendar enables to specify non-work periods, exceptions to non-work periods, and holidays for the entire project and all its resources. Each resource has a calendar that defines when the resource is available without regard to the quantity of the resource. Resource calendars can be used to indicate personnel information, such as vacations, and equipment information like lease dates. Each resource calendar uses a base calendar as a template. Tailor the calendar to the resource by making exceptions to the base calendar, or adding non-work periods to it. These changes become part of the resource's calendar, but they do not affect the base calendar. Calendar 1 is the default base calendar to which P3 assigns all activities and resources unless we add other calendars and specifically assign activities or resources to them. When we define a

calendar for each resource and assign resources to activities, we should also designate the resource as driving and the activity type as independent or meeting. Driving resources determine the remaining duration for activities to which they are assigned. P3 schedules the remaining work of meeting and independent activities using the calendars of their driving resources. P3 schedules work for activities of other types using the base calendar of the activity. P3 ignores resource calendars for non-driving resources.

Resource leveling: this is a process that helps us assure there are sufficient resources available to perform the activities in the project as they are planned. While computerized resource leveling provides with a valid resolution of resource conflicts, it may not necessarily be the optimal solution. Through analysis, it may be able to find alternate solutions, such as changing the project logic to avoid resource conflicts, or reallocating resources. During resource leveling P3 only schedules an activity to occur when its resource demands can be met. To accomplish this, P3 may delay or advance activities to resolve resource availability conflicts. Typically, leveling is done on the forward pass through the project. P3 determines the earliest dates it can schedule an activity when sufficient resources will be available to perform the task. The method P3 uses to forward level a project involves several steps.

1. P3 sorts the entire network into a logical sequence of activities to find the first activity or activities without unsatisfied predecessor relationships. These activities are ready for leveling. In the event of a tie, P3 uses the priority specification as the tiebreaker. In the event that prioritization values are also tied, P3 sorts by activity ID.
2. P3 initially assumes the normal resource limits. P3 deducts from the available resource pool any resources that activities in progress are using.

3 P3 checks that there will be sufficient resources available to satisfy the activity's demands for its entire duration. If so, P3 schedules the activity and returns to step 1. If not, the leveling process continues with step 4. Resource requirements are not always constant and continuous during every work period of an activity's duration. We can split, stretch, or crunch certain types of activities during forward leveling.

4 If the activity has no positive total float, P3 goes to step 5; otherwise, it delays the activity's start by one work period and re-examines the resource availability during each work period of the activity. If there are always sufficient resources available, P3 schedules the activity and goes back to step 1. If not, it repeats step 4 until the total float is exhausted.

5 If P3 previously reached step 5 with this activity, it skips this step and goes directly to step 6. When P3 reaches step 5 for the first time, the positive float of the activity has been exhausted without finding sufficient resources to schedule the activity under the normal resource limits. P3 tries to schedule the activity again, using the activity's original, early scheduled date, but this time assuming a higher resource availability depending on whether or not resource smoothing is in effect, and if it is, which smoothing option is selected in the smoothing section of the resource leveling dialog box. There are three smoothing options: *None*, *Non-time constrained*, and *Time constrained*. All three options lead back to step 3. Choose *None* and P3 assumes the maximum resource availability. Choose *Non-time constrained* and P3 assumes an increase in resource availability of ten percent of the difference between the normal and maximum limits. Choose *Time constrained* and P3 assumes double the maximum resource availability.

6 If the leveling process reaches step 6, P3 could not schedule the activity within its total float, even assuming the maximum resource availability. As a result, P3 delays the activity beyond its total float. This means that the activity will delay the project's completion. P3 continues to assume the maximum resource availability, and delays the activity one work period at a time, each time checking to see if sufficient resources are available during each work period of the activity. If P3 never finds enough resources, looking as far into the future as available memory allows, P3 schedules the activity on its original, early schedule dates, and issues a warning message in the leveling analysis report advising that the activity could not be leveled. If forward leveling delays the project's early finish date, P3 performs a backward pass to recalculate late dates. P3 does not recalculate late dates if there is an imposed project finish date, or if the project's early finish date is unchanged by leveling. P3 also enables to backward level. Backward leveling is the opposite of forward leveling; it schedules activities as late as possible, rather than as early as possible, without exceeding resource availability. P3 performs backward leveling during the backward pass, beginning at the project's latest late finish and working backwards toward the start of the project. P3 calculates late dates for the activities; early dates are not involved in backward leveling. P3 does not perform backward leveling if any activity being considered is an independent or meeting activity. If P3 detects an independent or meeting activity, the leveling run stops and P3 provides an explanation onscreen and in the output file, P3.OUT. Backward leveling can schedule activities to start earlier than their early start dates, resulting in negative total float. P3 searches as far back as the project's calendar start date to attempt to schedule an activity. If available resources are still insufficient, P3 schedules the activity on its late schedule dates and includes a warning message in the leveling analysis report.

Set options for resource splitting, stretching, and crunching: There are activity-level and project-level options related to splitting, stretching, and crunching activities and their

resources. The leveling options available for splitting, stretching, or crunching an activity and its resources depend on the activity type.

Project Groups: P3's project group and project capabilities enable to manage more than one project simultaneously and handle large projects efficiently. A project group contains all activities in the project and provides a comprehensive look at all projects. A project is a portion of a project group that is usually independently controlled. All changes that occur in a project are automatically reflected in its project group, and changes to a project group are reflected in the projects it contains. We can add relationships between projects from the project group to show dependencies. Using the project group/project feature brings data from many projects under one plan. Design an unlimited number of layouts to communicate project status from different viewpoints. For example, one layout can show a summary of schedule dates across all projects, and another can display detailed budget information, display resource profiles and tables to see exactly when and where resources are used. All users of the project share layouts and standard report specifications. Managing multiple projects requires coordination among all project participants. Therefore, before we add project groups and projects, we should develop standard procedures and conventions for the following: Work schedules, coding structures, resources, and unit prices.

15.2.3 A planning example with Primavera

The most important facilities and features of Primavera have been discussed in a nutshell above. A typical highway-widening project is worked out here to bring forth the lucid example of application of the Primavera software. The detailed activities list, analysis and results are given in the Tables 15.1 and 15.2. Figs 15.1 to 15.6 give the different graphical output from the Primavera package for this highway-widening project.

Table 15.1 Activities list of a highway construction (widening) project

Activity ID	Total float	Activity description	Original duration	Remaining duration
P1G098		BEGIN PROJECT	0	0
P1G100		MOBILIZATION	4	0
P1G105		CLEAR SITE	4	0
P1G110		GENERAL CONDITIONS	13	0
P1G200B		PROCUREMENT OF ESW/SL SIGNS	10	0
P1G5200	7	CONSTRUCT CONCRETE BARRIER AGAINST WALL	8	8
P1N1140		PLACE TEMP. MARKING TAPE	1	0
P1N1145		PLACE TEMP. CONST. BARRIER	1	0
P1N1150		EXCAVATE	5	0
P1N1170		PLACE AGGREGATE & ASPHALT BASE COURSE	6	0
P1N1180		REMOVE TEMP. CONST. BARRIER	1	0
P1N1185		PLACE 2" SURFACE COURSE	3	0
P1N1190		DIVERT TRAFFIC ONTO TEMP. ROAD	1	0
P1N1195		REMOVE TEMP. MARKING TAPE	1	0
P1N1200		PLACE CONST. BARRIER & TEMP. STRIPING RAMP	4	0
P1N1202		CONSTRUCT TEMP. BENT #1 & #2	20	0
P1N1210		EXCAVATE RAMP & INSTALL 15" RCC PIPE WITH INLETS	10	0
P1N1220		PLACE AGGREGATE & ASPHALT BASE COURSE	4	0
P1N1230		PLACE 2" SURFACE COURSE	3	0
P1N1235		REMOVE TEMP. CONSTRUCTION BARRIER	1	0
P1N1240		PLACE TEMP. STRIPING & DIVERT TRAFFIC	1	0
P1N2100	-1	PLACE CONST. BARRIER	12	12
P1N2105B	-1	INSTALL EROSION CONTROL DEVICES	15	15
P1N2115A	-1	GRUB & STRIP TOPSOIL	8	8
P1N2190	197	INSTALL SIGN SUPPORT STRUCTURE	8	8
P1N2195	197	ERECT SIGNS	3	3
P1N2200	-1	INSTALL TEMP. SHEETING	12	12
P1N2210	-1	EXCAVATE RETAINING WALL	12	12
P1N2220	-1	CONSTRUCT FOOTING	27	27
P1N2230	-1	CONSTRUCT RETAINING WALL	27	27
P1N2240	-1	PLACE POROUS FILL BEHIND WALL	9	9
P1N2250	-1	BACKFILL RETAINING WALL	7	7
P1N2260	-1	CONSTRUCT CONCRETE BARRIER AGAINST WALL	20	20
P1N2270	-1	CONSTRUCT CONCRETE BARRIER AGAINST WALL	24	24
P1N2300	-1	CONSTRUCT CONCRETE BARRIER AGAINST WALL	25	25
P1N2320	-1	CONSTRUCT CONCRETE BARRIER AGAINST WALL	24	24
P1N3200	-1	CONSTRUCT CONCRETE BARRIER AGAINST WALL	20	20
P1N3220	-1	CONSTRUCT CONCRETE BARRIER AGAINST WALL	6	6
P1N3225	-1	CONSTRUCT CONCRETE BARRIER AGAINST WALL	6	6
P1NP100A	127	PROCUREMENT OF NOISE BARRIER POST & PANELS	10	19
P1NP200C		PROCUREMENT OF SIGN STRUCTURES	10	0
P2G600	-1	PROJECT COMPLETE	0	0
P2N2115B	37	REMOVE GUIDE RAIL	4	4
P2N2125A	42	EXCAVATE FOR ELECTRICAL	10	10
P2N2125B	37	INSTALL TEMP. ELECTRICS	15	15
P2N2130	37	INSTALL ELECTRIC CONDUITS & STRUCTURES	19	19
P2N2140	37	INSTALL POWER & LIGHTING	15	15
P2N2150	37	SUB-GRADE PREPARATION	14	14
P2N2170	37	PLACE AGGREGATE & ASPHALT BASE COURSE	10	10
P2N2180	37	PLACE ASPHALT STABILIZED COURSE	12	12
P2N2185	37	PLACE 2" SURFACE COURSE	10	10
P2N2196	37	STRIPE ROADWAY	9	9
P2N2198	37	REMOVE TEMP. BARRIER	4	4
P2N3100	123	PLACE TEMP. CONST. BARRIER	6	6
P2N3110	123	REMOVE TEMP. PAVEMENT	9	9
P2N3120	123	RE-GRADE AREA	7	7
P2N3225A	134	REMOVE TEMP. CONST. BARRIER	6	6

Table 15.2 Activity particulars of the highway construction project

Activity ID	Lane Closing	Percent complete	Original duration	Rem. duration	Early start	Early finish	Late start	Late finish	Total float	Budgeted cost
PIG098		100	0	0	24SEP98A	24SEP98A				0.00
PIG100		100	4	0	24SEP98A	27SEP98A	24SEP98A	27SEP98A		100.00
PIG110		100	13	0	08OCT98A	04NOV98A	08OCT98A	04NOV98A		19500.00
PIG105		100	4	0	01OCT98A	04OCT98A	01OCT98A	04OCT98A		3583.68
PIN1140	NO7	100	1	0	03NOV98A	03NOV98A	03NOV98A	03NOV98A		288.32
PIN1145	NO7	100	1	0	04NOV98A	25NOV98A	04NOV98A	25NOV98A		1790.52
PIN1202	NO8	100	20	0	17DEC98A	17JAN99A	17DEC98A	17JAN99A		22004.40
PIN1180	NO7	100	1	0	23DEC98A	11JAN99A	23DEC98A	11JAN99A		1790.52
PIN1190	NO7	100	1	0	20JAN99A	28JAN99A	20JAN99A	28JAN99A		248.00
PIN1195	NO7	100	1	0	01FEB99A	09FEB99A	01FEB99A	09FEB99A		288.32
PIN1200	NO7	100	4	0	17FEB99A	17MAR99A	17FEB99A	17MAR99A		2588.12
PIN1235	NO7	100	1	0	13MAY99A	13MAY99A	13MAY99A	13MAY99A		1666.52
PIN1240	NO7	100	.1	0	20MAY99A	02JUN99A	20MAY99A	02JUN99A		248.00
PIN2100	NO7	0	12	12	07JUN99	22JUN99	04JUN99	21JUN99	-1	17799.84
P2N3100	SO3	0	6	6	03SEP99	10SEP99	25FEB00	03MAR00	123	9999.12
P2N3225A	SO3	0	5	5	13SEP99	17SEP99	21MAR00	27MAR00	134	8332.60
P2N2196	SO4	0	9	9	18JAN00	28JAN00	09MAR00	21MAR00	37	10000.08
P2N2198	SO4	0	4	4	31JAN00	03FEB00	22MAR00	27MAR00	37	6666.08
PIG200B		100	10	0	19APR99A	29APR99A	19APR99A	29APR99A		22000.00
PINP200C		100	10	0	02MAY99A	03JUN99A	02MAY99A	03JUN99A		20000.00
PINP100A		0	10	19	02JUN99A	01JUL99	02JUN99A	29DEC99	127	12000.0
PIN2190	NO5	0	8	8	07JUN99	16JUN99	13MAR00	22MAR00	197	6895.36
PIN2195	NO5	0	3	3	17JUN99	21JUN99	23MAR00	27MAR00	197	5371.56
PIN1150	NO7	100	5	0	29NOV98A	07DEC98A	29NOV98A	07DEC98A		2846.60
PIN1210	NO7	100	10	0	18MAR99A	30MAR99A	18MAR99A	30MAR99A		11149.20
PIN2115A	NO7	0	8	8	23JUN99	02JUL99	22JUN99	01JUL99	-1	12533.12
PIN2105B	NO7	0	15	15	06JUL99	26JUL99	02JUL99	23JUL99	-1	3760.32
PIN2200	NO7	0	12	12	27JUL99	11AUG99	26JUL99	10AUG99	-1	9347.04
PIN2210	NO8	0	12	12	12AUG99	27AUG99	11AUG99	26AUG99	-1	7542.24
PIN2240	NO8	0	9	9	27OCT99	08NOV99	26OCT99	05NOV99	-1	11107.08
PIN2250	NO8	0	7	7	09NOV99	17NOV99	08NOV99	16NOV99	-1	7770.84
P2N2115B	SO4	0	4	4	30AUG99	02SEP99	20OCT99	25OCT99	37	3115.68
P2N2125A	SO4	0	10	10	03SEP99	16SEP99	02NOV99	15NOV99	42	6141.20
P2N3110	SO3	0	9	9	13SEP99	23SEP99	06MAR00	16MAR00	123	6772.68
P2N3120	SO3	0	7	7	24SEP99	04OCT99	17MAR00	27MAR00	123	6902.84
P2N2125B	SO4	0	15	15	03SEP99	23SEP99	26OCT99	15NOV99	37	8884.80
P2N2130	SO4	0	19	19	24SEP99	20OCT99	16NOV99	10DEC99	37	42243.08

Table 15.2 (Continued) Activity particulars of the highway construction project

Activity ID	Lane Closing	Percent complete	Original duration	Rem. duration	Early start	Early finish	Late start	Late finish	Total float	Budgeted cost
P2N2140	SO4	0	15	15	21OCT99	10NOV99	13DEC99	04JAN00	37	16954.80
F1N1170	NO7	100	6	0	14DEC98A	22DEC98A	14DEC98A	22DEC98A		107477.52
F1N1185	NO7	100	3	0	07JAN99A	19JAN99A	07JAN99A	19JAN99A		23865.36
F1N1220	NO7	100	4	0	15APR99A	29APR99A	15APR99A	29APR99A		75774.08
F1N1230	NO7	100	3	0	02MAY99A	09MAY99A	02MAY99A	09MAY99A		24147.36
P2N2150	SO4	0	14	14	11NOV99	30NOV99	05JAN00	24JAN00	37	7010.08
P2N2170	SO4	0	10	10	01DEC99	14DEC99	25JAN00	07FEB00	37	189699.20
P2N2180	SO4	0	12	12	15DEC99	03JAN00	08FEB00	23FEB00	37	263867.84
P2N2185	SO4	0	10	10	04JAN00	17JAN00	24FEB00	08MAR00	37	74875.70
P1N2220	NO8	0	27	27	30AUG99	05OCT99	27AUG99	04OCT99	-1	110196.72
P1N2230	NO8	0	27	27	20SEP99	26OCT99	17SEP99	25OCT99	-1	68210.64
P1G5200	NO8	0	8	8	27OCT99	05NOV99	05NOV99	16NOV99	7	14722.56
P1N2260	NO8	0	20	20	18NOV99	15DEC99	17NOV99	14DEC99	-1	38606.40
P1N2270	NO8	0	24	24	09DEC99	13JAN00	08DEC99	12JAN00	-1	16121.28
F1N2300	NO8	0	25	25	03JAN00*	04FEB00	30DEC99	03FEB00	-1	48258.00
F1N2320	NO8	0	24	24	24JAN00	24FEB00	21JAN00	23FEB00	-1	16927.68
F1N3200	NO8	0	20	20	14FEB00*	10MAR00	11FEB00	09MAR00	-1	38606.40
F1N3220	NO8	0	6	6	13MAR00	20MAR00	10MAR00	17MAR00	-1	11581.92
F1N3225B	NO8	0	6	6	21MAR00	28MAR00	20MAR00	27MAR00	-1	11581.92
P2G600		0	0	0		28MAR00	27MAR00		-1	0.00

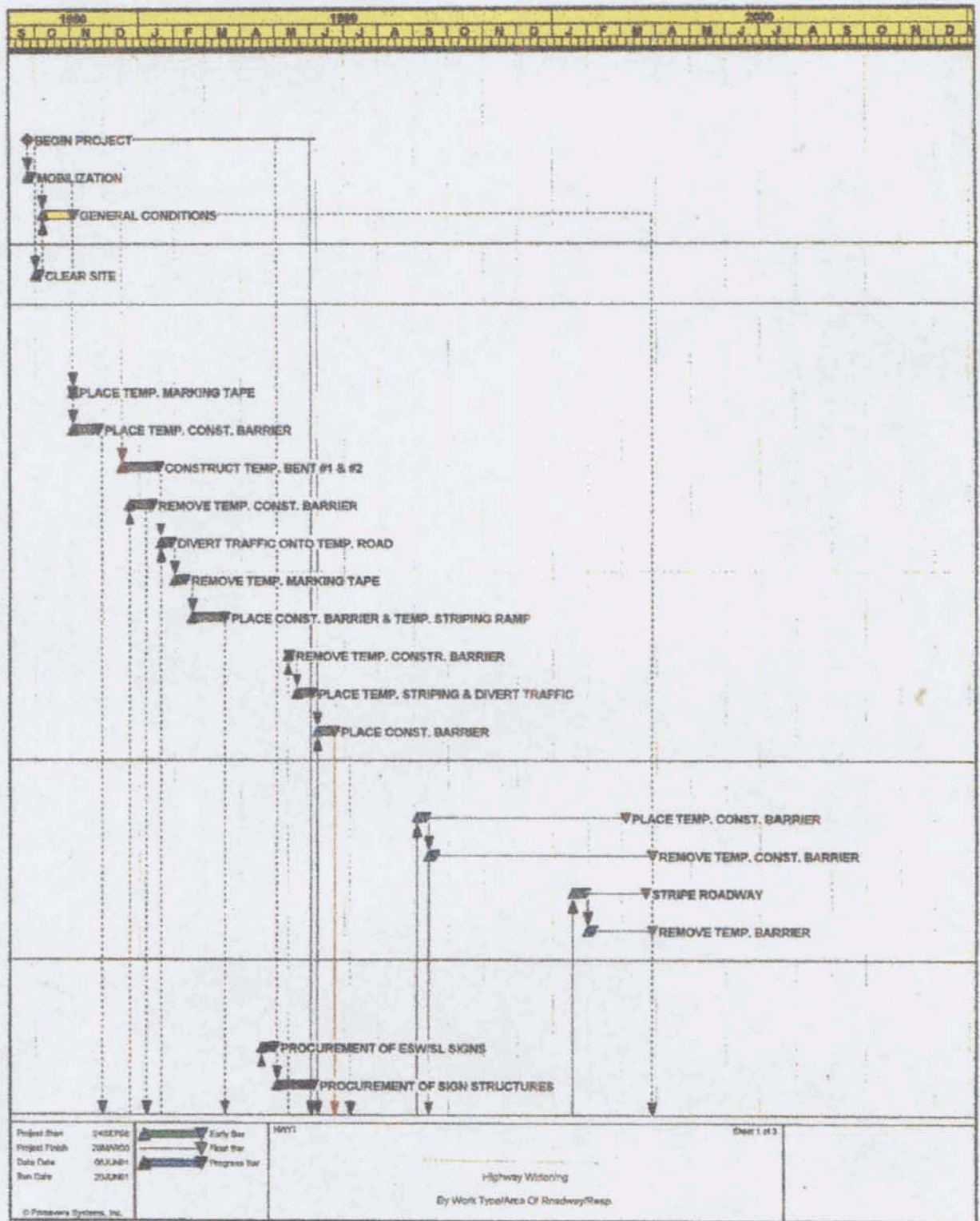


Fig 15.1(a) Bar chart of the highway construction (widening) project (Page 1)

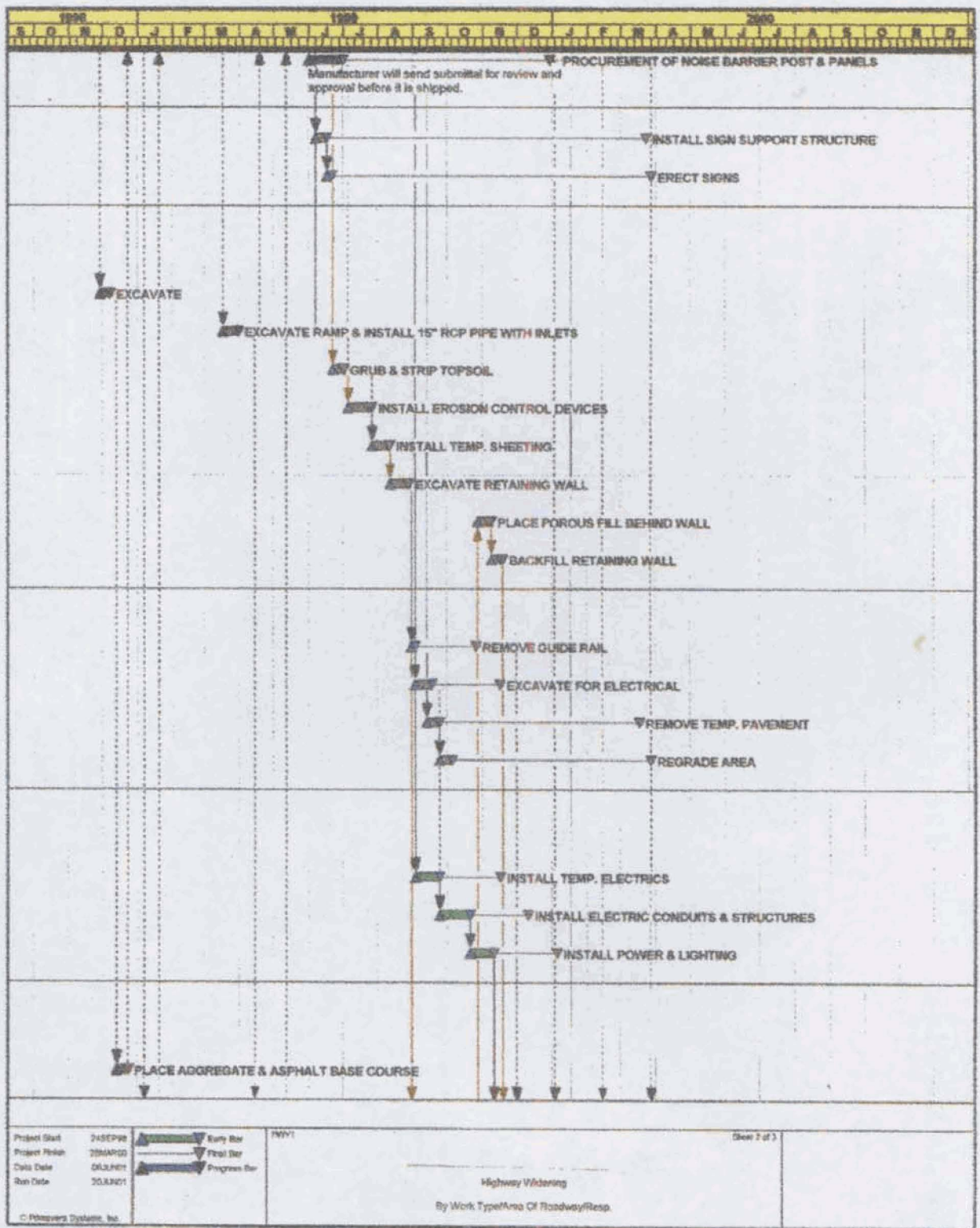


Fig 15.1(b) Bar chart of the highway construction (widening) project (Page 2)

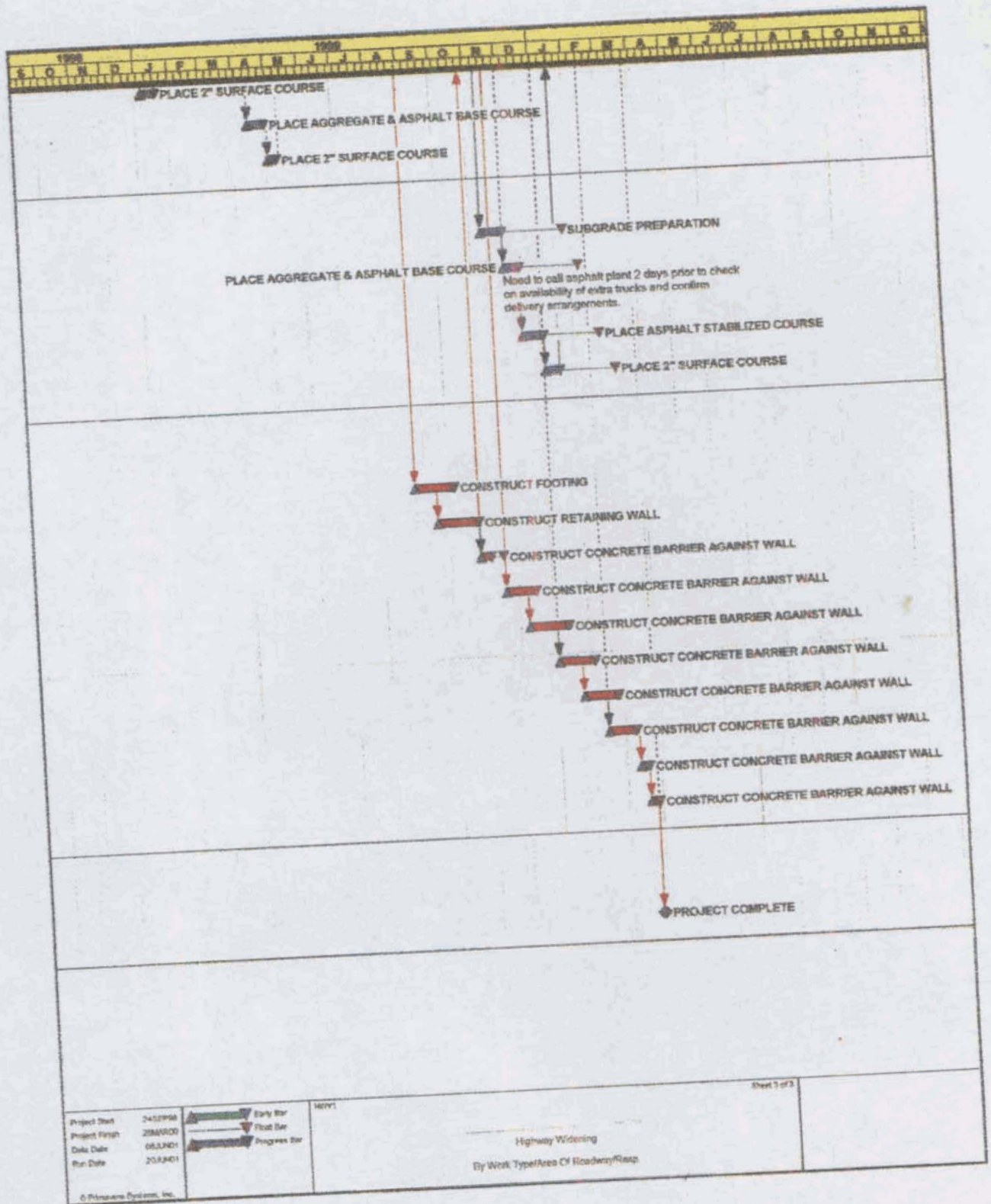


Fig 15.1(c) Bar chart of the highway construction (widening) project (Page 3)

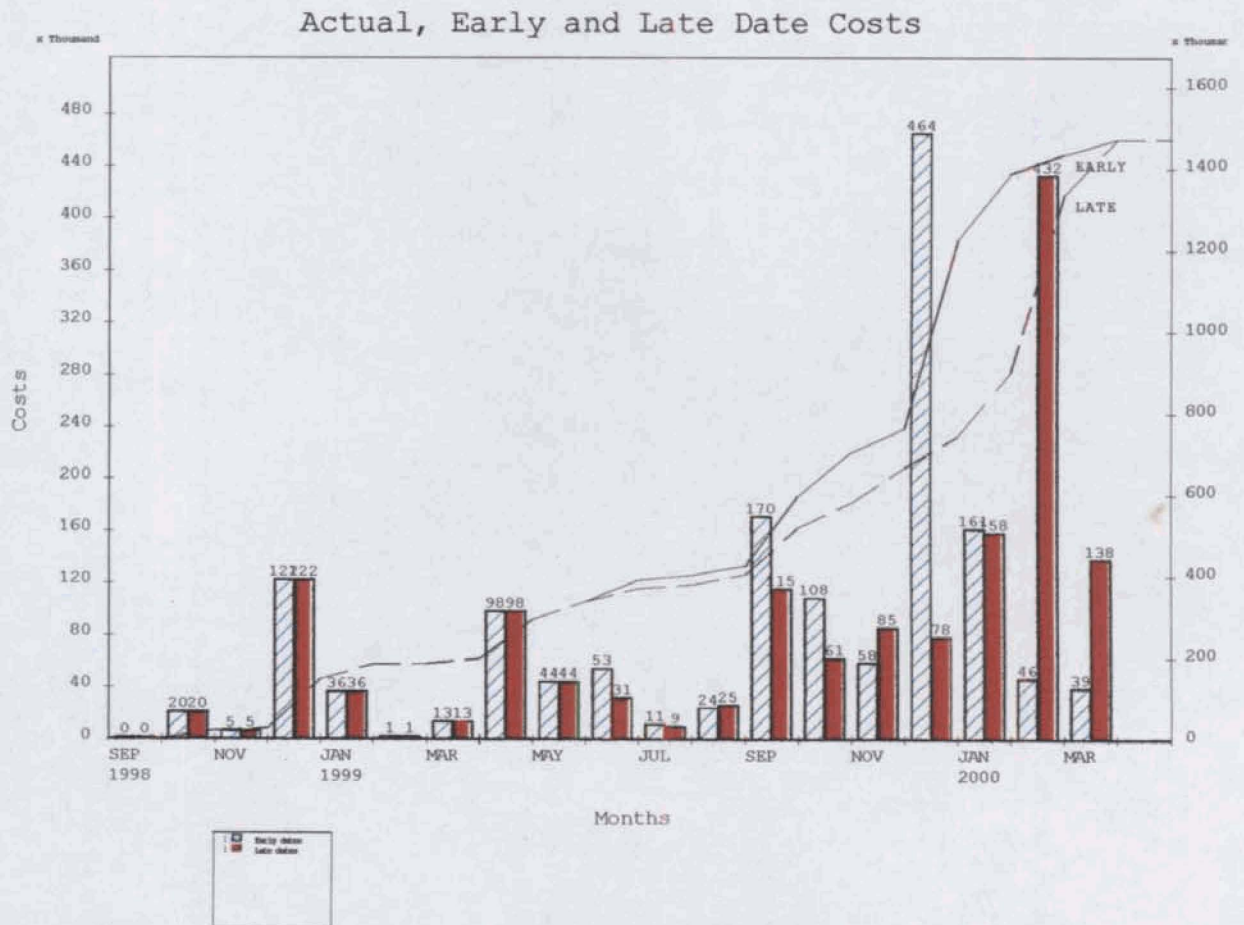


Fig 15.2 Costs graph of the highway construction (widening) project

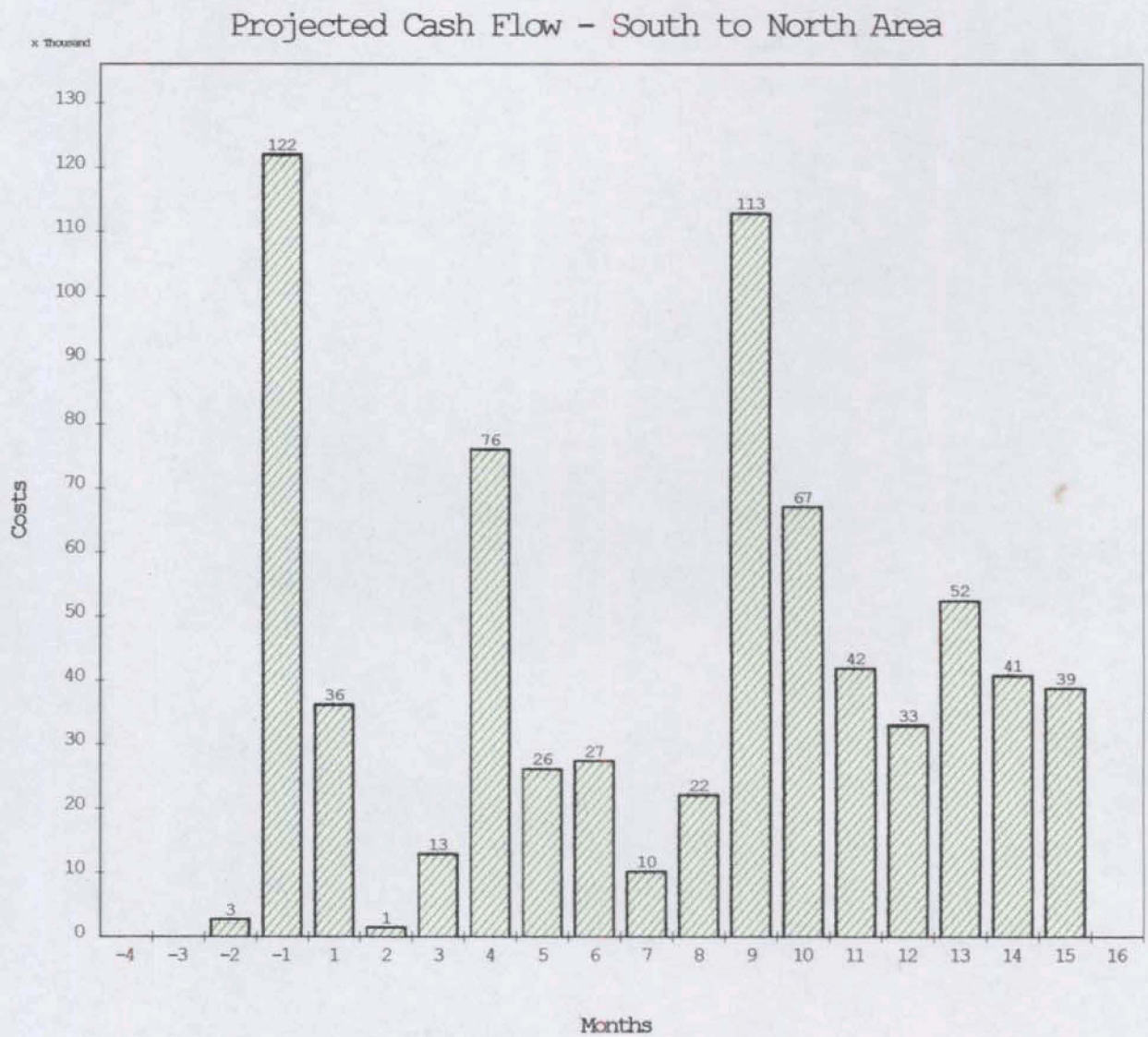
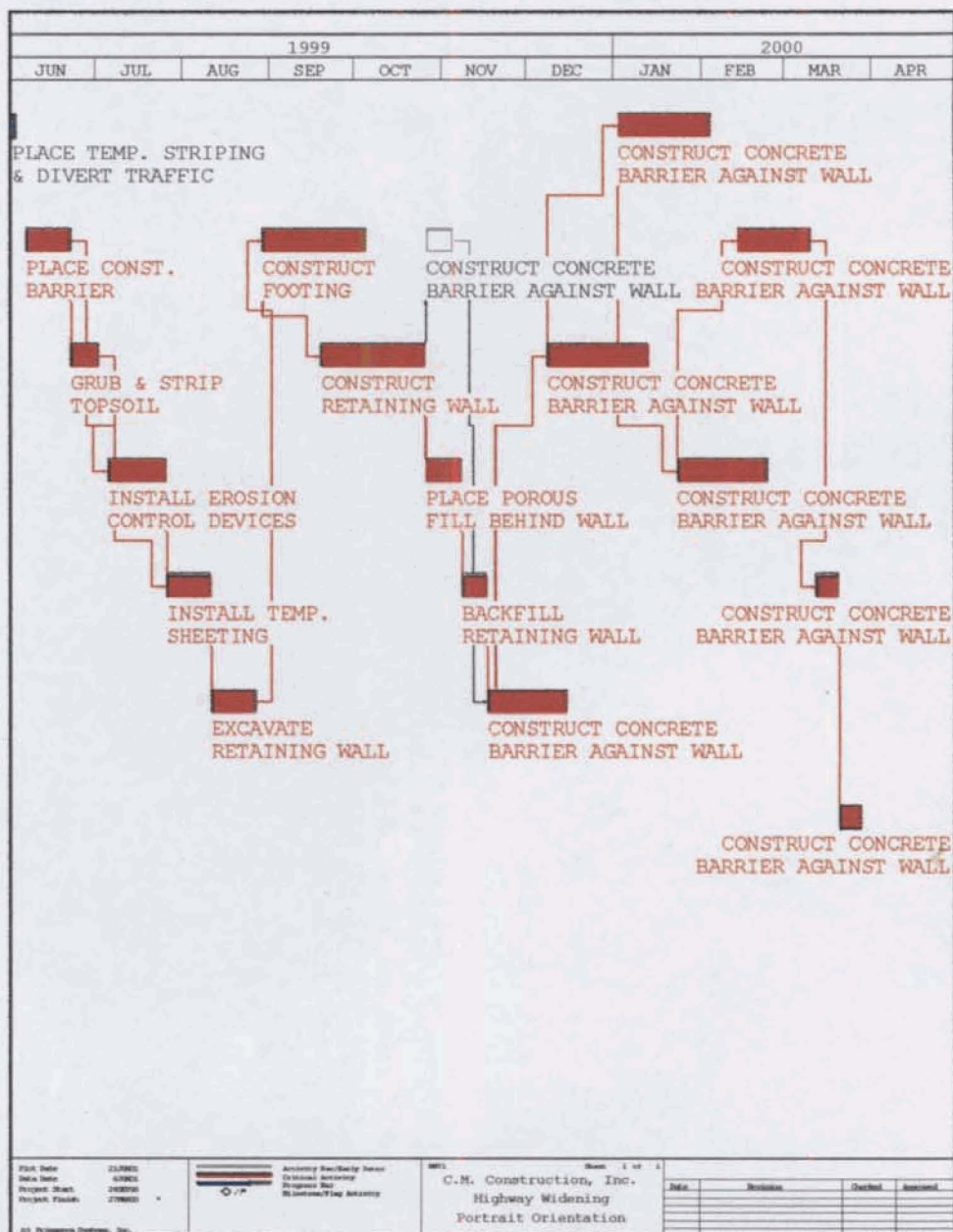


Fig 15.3 Projected cash flow of the highway project (Primavera)



15.5 Time scaled logic diagram (Primavera)

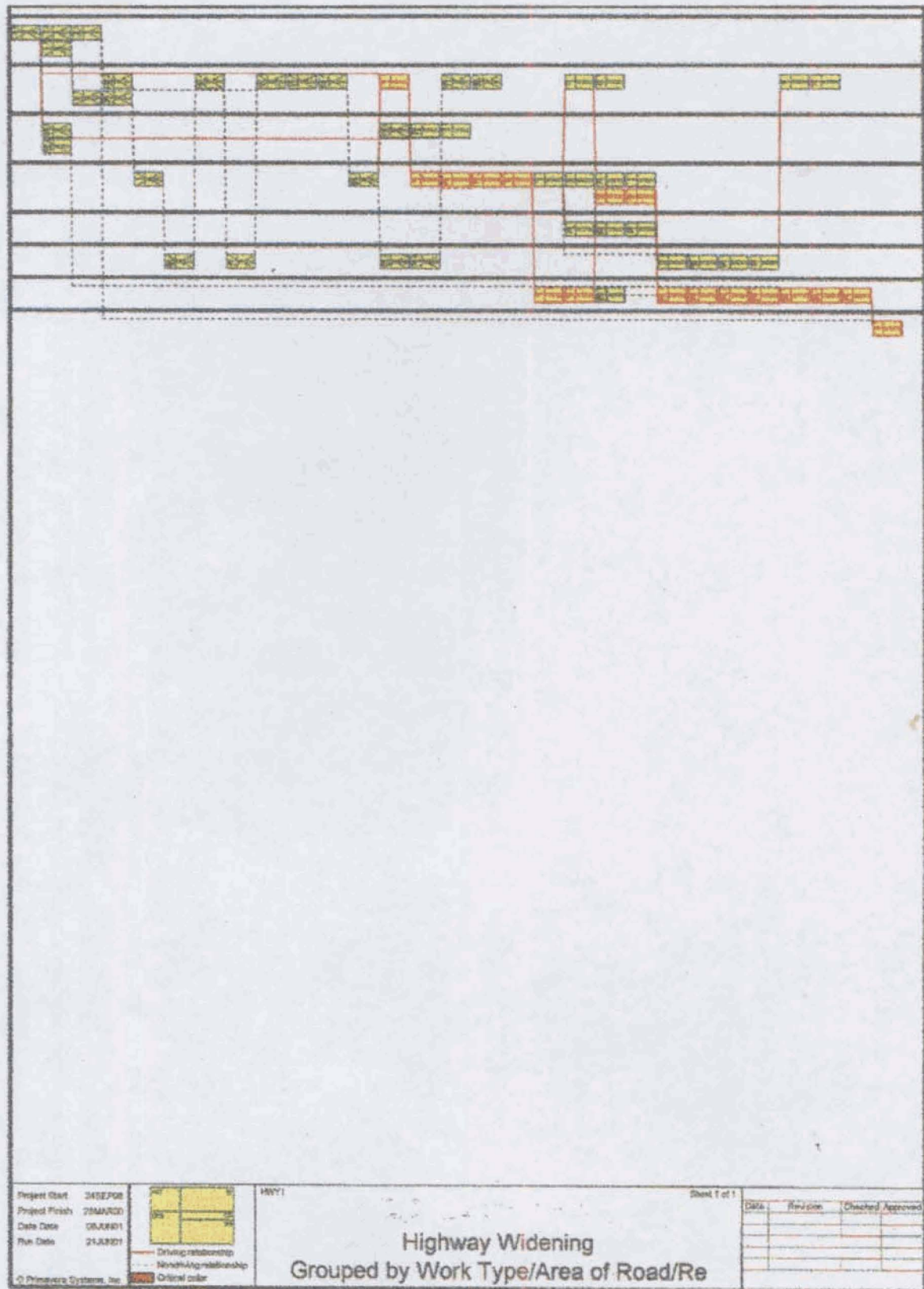


Fig 15.6 Sample PERT chart for the highway project (Primavera)

15.3 Microsoft Project

The *Microsoft Project* was brought out by the Microsoft Corporation in 1990. The package used by the author is *Microsoft Project 98*, released in 1998. This works conveniently on the windows platform and is compatible to Microsoft Office products. The following are the compatible web servers and browsers, and workgroup messages.

O'Reilly Web Site Pro Server

Netscape Fastrack and Enterprise Server

Netscape Navigator Browser

Microsoft Internet Explorer Browser

The minimum system requirements for Microsoft project 98 are

- * Personal or multimedia computer with a 486 or higher processor.
- * Microsoft Windows 95 operating system or above
- * Windows NT Workstation 3.51 with Service Pack 5 or later, or
- * Windows NT Workstation 4.0 with Service Pack 2 or later.
- * 12 MB of memory for use on Windows 95 or above
- * 16 MB of memory for use on Windows NT Workstation.
- * Hard disk space required;

21 MB required for typical installation, depending on configuration and

35 MB for a full installation.

- * CD-ROM drive.
- * VGA or higher-resolution video adapter (Super VGA, 256-color recommended).
- * Microsoft Mouse, Microsoft IntelliMouse, or compatible pointing device

(Microsoft IntelliMouse required for extended navigational capabilities).

15.3.1 Facilities and features available with *Microsoft project*

With *Microsoft Project*, we can display project information in task views or in resource views. Use a task view when we want to enter, change, or display task information. Use a resource view when we want to enter, change, or display resource information. The following table contains all of the *Microsoft Project* views, including those traditionally used by project management professionals (such as the Gantt Chart and PERT Chart), as well as specialized formats for entering and displaying data. We can work with the predefined views or create our own special-purpose views.

Views available in Microsoft Project:

1. *Bar Rollup:* A list of summary tasks containing labels for all subtasks. Use this view with the Rollup Formatting macro to see all tasks concisely labeled on summary task bars.
2. *Calendar:* A monthly calendar showing tasks and durations. Use this task view to show the tasks scheduled in a specific week or range of weeks.
3. *Detail Gantt:* A list of tasks and related information, and a chart showing slack and slippage. Use this task view to check how far a task can slip without affecting other tasks.
4. *Gantt Chart:* A list of tasks and related information, and a chart showing tasks and durations over time. Use this task view to enter and schedule a list of tasks.

5. *Leveling Gantt*: A list of tasks, information about task delays and slack, and a bar chart showing the before and after effects of leveling. Use this task view to check the amount of task delay.
6. *Milestone Date Rollup*: A list of summary tasks containing labels for all subtasks. Use this view with the Rollup-Formatting macro to see all tasks concisely labeled with milestone marks and dates on summary task bars.
7. *Milestone Rollup*: A list of summary tasks containing labels for all subtasks. Use this view with the Rollup-Formatting macro to see all tasks concisely labeled with milestone marks on summary task bars.
8. *PA-Expected Gantt*: A Gantt chart showing schedule's expected-case scenario. Use this view when performing a PERT analysis on a task's duration or on the project's schedule.
9. *PA-Optimistic Gantt*: A Gantt chart showing schedule's best-case scenario. Use this view when performing a PERT analysis on a task's duration or on the project's schedule.
10. *PA-PERT Entry Sheet*: A sheet for entering schedule's best-case, expected-case, and worst-case scenarios for a task's duration prior to calculating the most probable duration. Use this view to compare the disparity between the different task estimates.
11. *PA-Pessimistic Gantt*: A Gantt chart showing schedule's worse case scenario. Use this view when performing a PERT analysis on a task's duration or on the project's schedule.
12. *PERT Chart*: A network diagram showing all tasks and task dependencies. Use this task view to create and fine-tune the schedule in a flowchart format.

- 13. Resource Allocation:* A combination view with the Resource Usage view in the top pane and the Leveling Gantt view in the bottom pane. Use this resource view to resolve resource over allocations.
- 14. Resource Form:* A form for entering and editing information about a specific resource.
- 15. Resource Graph:* A graph showing resource allocation, cost, or work over time. Use this resource view to display information about a single resource or group of resources over time.
- 16. Resource Name Form:* A form for entering and editing the resource name and other resource information.
- 17. Resource Sheet:* A list of resources and related information. Use this resource view to enter and edit resource information in a spreadsheet-like format.
- 18. Resource Usage:* A list of resources showing allocation, cost, or work information for each resource over time. Use this resource view to show cost or work allocation information for each resource and to set resource contours.
- 19. Task Details Form:* A form for reviewing and editing detailed tracking and scheduling information about a specific task.
- 20. Task Entry:* A combination view with the Gantt chart view in the top pane and the Task Form view in the bottom pane. Use this task view to add, edit, and review detailed information about the task selected in the Gantt chart.
- 21. Task Form:* A form for entering and editing information about a specific task.
- 22. Task Name Form:* A form for entering and editing the task name and other task information.

23. *Task PERT*: A network diagram showing the predecessors and successors of one task. In a large project, use this task view to focus on the task dependencies of a specific task.

24. *Task Sheet*: A list of tasks and related information. Use this task view to enter and schedule tasks in a spreadsheet-like format.

25. *Task Usage*: A list of tasks showing assigned resources grouped under each task. Use this task view to see which resources are assigned to specific tasks and to set resource work contours.

26. *Tracking Gantt*: A list of tasks and related information, and a chart showing baseline and scheduled Gantt bars for each task. Use this task view to compare the baseline schedule with the actual schedule.

15.3.2 The Gantt chart view in *Microsoft Project*

The Gantt chart view displays task information about the project as both text and bar graphics. On the left side of this view are columns (containing Microsoft Project fields) in which we can enter and modify task names, durations, and start and finish dates, and other information. On the right side of this view, we can use Gantt bars to graphically display task durations and start and finish dates on a timescale. The relative position of the Gantt bars shows the sequence in which the project tasks are scheduled to occur.

Gantt Chart view can be used to:

- Create a project by entering tasks and task durations.
- Establish sequential relationships between tasks, which allows to see how changing a task duration affects the start and finish dates of other tasks and the project finish date.
- Assign personnel and other resources to tasks.
- Track the progress of the project by comparing scheduled dates with the actual start and finish dates and by checking the percentage of each task that is complete.

To make the Gantt chart view more effective, it can be customized. When saving the project, the customized view is saved with the project file.

15.3.3 The PERT chart view

The PERT chart view displays tasks and task dependencies as a network diagram or flowchart. A box (or node) represents each task and a line connecting two boxes represents the dependency between the two tasks. By default, the PERT chart view displays one diagonal line through a task that is in progress and crossed diagonal lines through a completed task.

PERT chart view can be used to:

- Create and fine-tune the schedule.
- Link tasks to specify the task sequence, as well as determine start and finish dates.
- Graphically show completed, in-progress, and not-yet-started tasks.
- Assign personnel and other resources (such as equipment) to specific tasks.

To make the PERT chart view fit the needs exactly, we can customize its appearance or create new, customized versions. When saving the project, the customized view is saved with the project. When we customize the PERT chart view, we can:

- Display in each PERT box the task information that is most important. For example, instead of displaying the scheduled start and finish dates, display the work and the cost.
- Assign a different border style to tasks of a certain type.
- Change the appearance of the lines that connect PERT boxes and prevent PERT boxes from crossing page breaks.
- Format a category of information to distinguish that type of information from all other information. For example, we can italicize all summary tasks and format all milestone tasks as bold text.
- Change the PERT chart view to display more or fewer boxes, so that we can see more of the project at once or focus on a small portion of it.

- Align PERT boxes to give them an orderly appearance.
- Create a combination view for the PERT chart to display additional information about the tasks shown in the PERT chart view or about the resources assigned to those tasks.

15.3.4 The Resource Usage view

The Resource Usage view displays project resources with their assigned tasks grouped underneath them. Use the Resource Usage view to:

- Enter and edit information on a resource's task assignment, such as cost, work allocation, and work availability.
- See which resources are over allocated and by how much.
- Distribute assignments more evenly between resources.
- Find out how many hours each resource is scheduled to work.
- See the percentage of capacity at which each resource is scheduled.
- Determine how much time each resource has available for additional work assignments.
- Find out how many hours each resource is scheduled to work on particular tasks.
- Review resource costs on a particular task.
- Vary the amount of work a person spends on a task by setting work contours.

To make the Resource Usage view fit the needs exactly, we can customize it or create a new, customized version. When customizing the Resource Usage view, we can:

- Display different resource information.
- Display information in different timescales.
- Format individual information to call attention to it.
- Format a category of information to distinguish that type of information from all other information. For example, italicize all over allocated resources and increase the size of all column titles.
- Change the units of time that are displayed to view the project at the level of detail needed.
- View usage information for a specific time period.
- Change the horizontal and vertical lines, called gridlines, to enhance the readability or clarity of the Resource Usage view. For example, we can specify different line patterns and colors for column and row gridlines.
- Create a combination view for the Resource Usage view to display additional information about the resources shown in the Resource Usage view or about the tasks to which those resources are assigned.

15.3.5 Other views in *Microsoft Project*

Out of the 26 standard views available in *Microsoft Project*, only 3 most popular have been discussed above. The other 23 are also versatile and needs much explanation to

bring out their full facility. It is not attempted here. Instead, some sample views output from a trial run on the package are reproduced.

15.3.6 Examples of using *Microsoft Project*

To illustrate the most popular facilities in the *Microsoft Project*, the same example of the highway construction (widening) project used in the *Primavera* package (as detailed in Tables 15.1, 15.2 and 15.3) is run on the *Microsoft project* package and the sample output are given in Figs.15.7 and 15.8. For fear of bulk, only a few views are reproduced. Since this highway-widening project is a long one, its PERT chart cannot be contained in a single page. To illustrate the PERT chart, another small house construction project (Table 15.4) is chosen and its different output views from *Microsoft project* are also reproduced (Figs. 15.9 and 15.10).

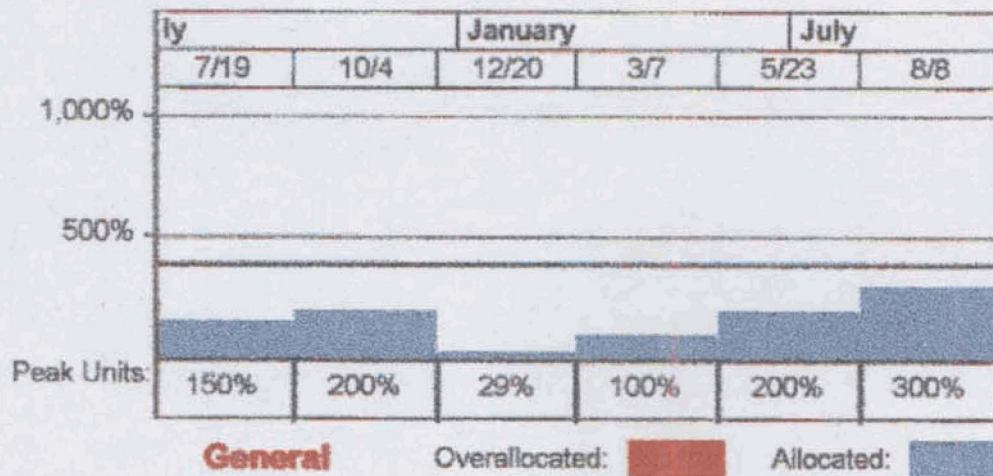


Fig 15.7 Resource graph output from *Microsoft Project* package

Table 15.3 **Data taken for Fig 15.7**

Resource name (Highway widening project of Table 15.1)	Cost	Baseline	Actual cost	Remaining cost
GENERAL LABORER	92,256	92,256	4,712	87,544
CLEAR SITE	372	372	372	0
CONSTRUCT CONCRETE BARRIER AGAINST WALL	2,976	2,976	0	2,976
PLACE TEMP. MARKING TAPE	248	248	248	0
PLACE TEMP. CONST. BARRIER	248	248	248	0
EXCAVATE	1,240	1,240	1,240	0
REMOVE TEMP. CONST. BARRIER	248	248	248	0
DIVERT TRAFFIC ONTO TEMP. ROAD	248	248	248	0
REMOVE TEMP. MARKING TAPE	248	248	248	0
PLACE CONST. BARRIER & TEMP. STRIPING RAMP	496	496	496	0
PLACE AGGREGATE & ASPHALT BASE COURSE	992	992	992	0
REMOVE TEMP. CONST. BARRIER	124	124	124	0
PLACE TEMP. STRIPING & DIVERT TRAFFIC	248	248	248	0
PLACE CONST. BARRIER	1,488	1,488	0	1,488
INSTALL EROSION CONTROL DEVICES	3,720	3,720	0	3,720
INSTALL SIGN SUPPORT STRUCTURE	992	992	0	992
ERECT SIGNS	744	744	0	744
INSTALL TEMP. SHEETING	4,464	4,464	0	4,464
EXCAVATE RETAINING WALL	1,488	1,488	0	1,488
CONSTRUCT FOOTING	10,044	10,044	0	10,044
CONSTRUCT RETAINING WALL	6,696	6,696	0	6,696
PLACE POROUS FILL BEHIND WALL	2,232	2,232	0	2,232
BACKFILL RETAINING WALL	868	868	0	868
CONSTRUCT CONCRETE BARRIER AGAINST WALL	7,440	7,440	0	7,440
CONSTRUCT CONCRETE BARRIER AGAINST WALL	8,928	8,928	0	8,928
CONSTRUCT CONCRETE BARRIER AGAINST WALL	9,300	9,300	0	9,300
CONSTRUCT CONCRETE BARRIER AGAINST WALL	8,928	8,928	0	8,928
CONSTRUCT CONCRETE BARRIER AGAINST WALL	7,440	7,440	0	7,440
CONSTRUCT CONCRETE BARRIER AGAINST WALL	2,232	2,232	0	2,232
CONSTRUCT CONCRETE BARRIER AGAINST WALL	2,232	2,232	0	2,232
EXCAVATE FOR ELECTRICAL	1,240	1,240	0	1,240
REMOVE TEMP. BARRIER	496	496	0	496
PLACE TEMP. CONST. BARRIER	744	744	0	744
REMOVE TEMP. PAVEMENT	2,232	2,232	0	2,232
REMOVE TEMP. CONST. BARRIER	620	620	0	620

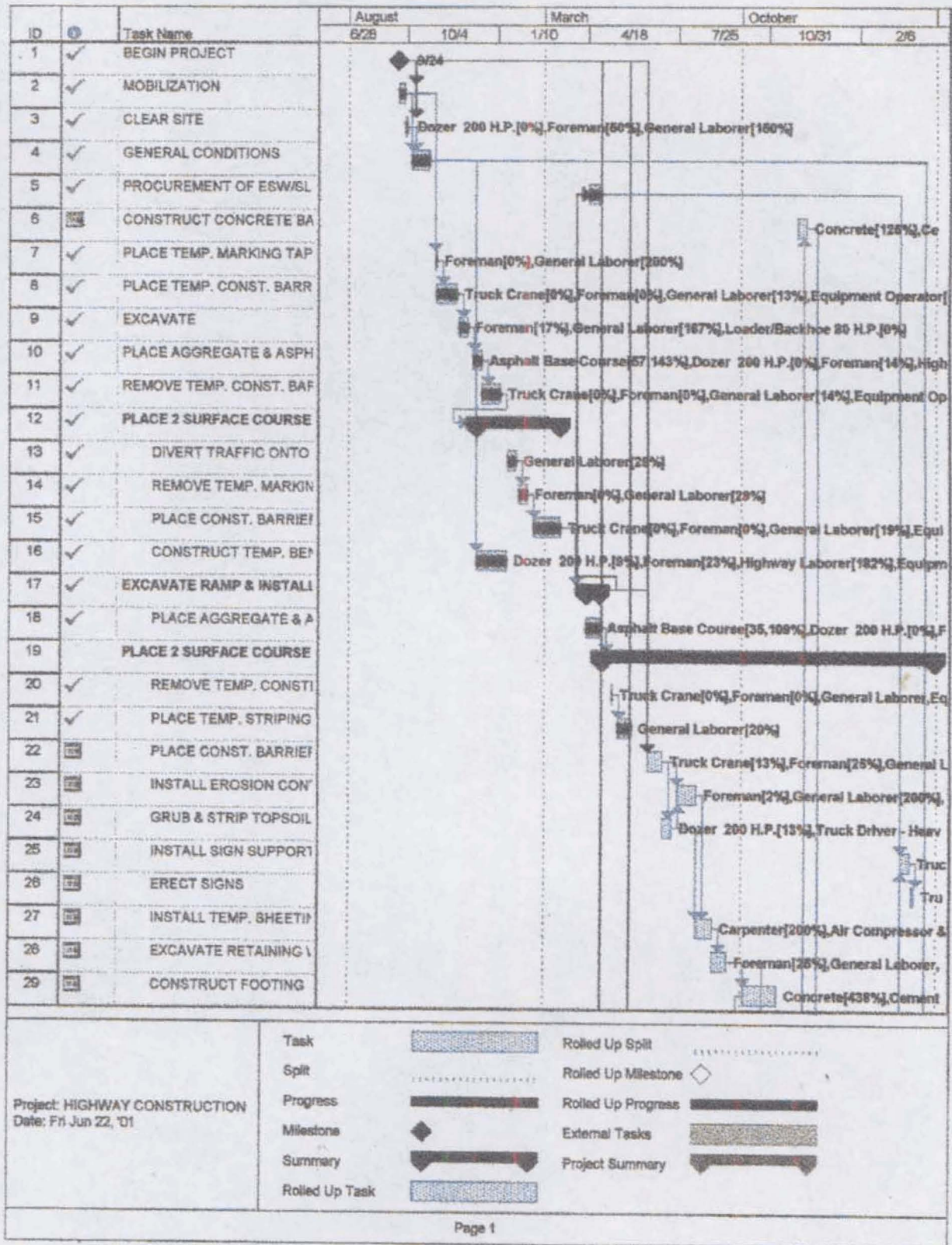


Fig 15.8 (a) Gantt chart for the highway construction (widening project)
Output from Microsoft project (Page 1)

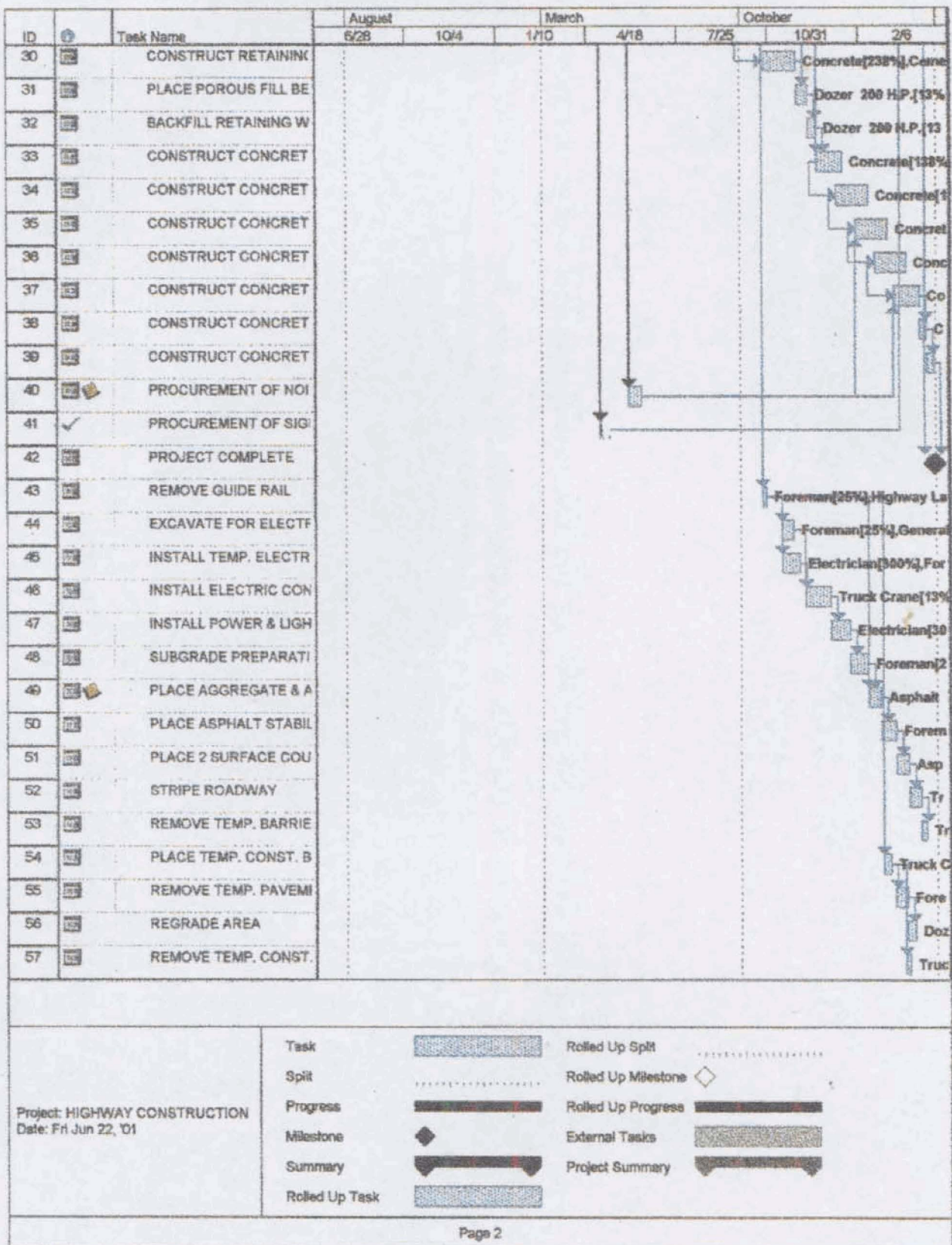
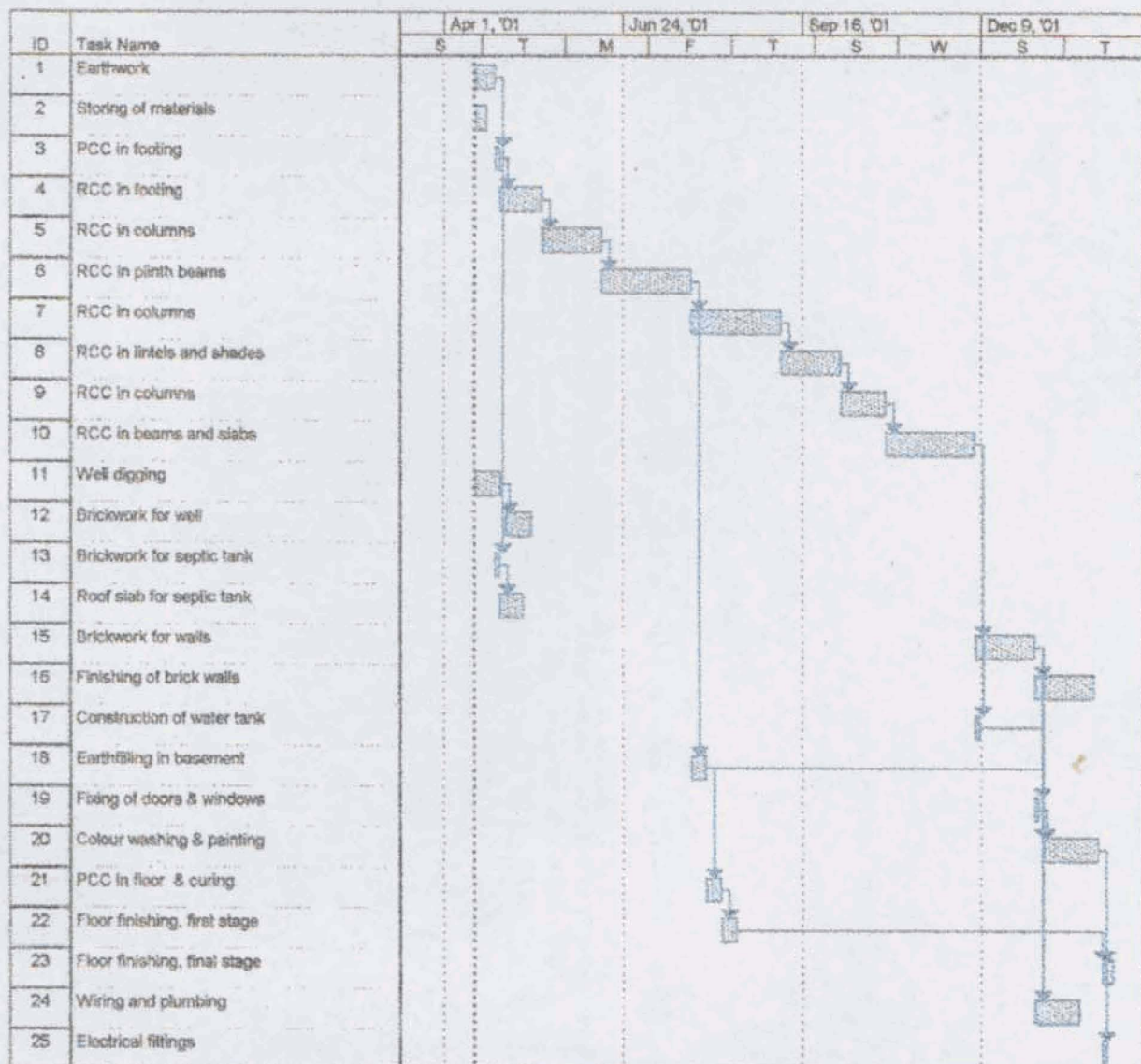


Fig 15.8 (b) Gantt chart for the highway construction (widening project)

Output from Microsoft project (Page 2)

Table 15.4 List of activities of a small house construction project
(For taking output of views from *Microsoft Project*)

ID	Task Name	Duration	Start	Finish	Predecessors
1	Earthwork	7 days	Mon 4/16/01	Tue 4/24/01	
2	Storing of materials	5 days	Mon 4/16/01	Fri 4/20/01	
3	PCC in footing	2 days	Wed 4/25/01	Thu 4/26/01	1
4	RCC in footing	14 days	Fri 4/27/01	Wed 5/16/01	3
5	RCC in columns	20 days	Thu 5/17/01	Wed 6/13/01	4
6	RCC in plinth beams	30 days	Thu 6/14/01	Wed 7/25/01	5
7	RCC in columns	30 days	Thu 7/26/01	Wed 9/5/01	6
8	RCC in lintels and shades	20 days	Thu 9/6/01	Wed 10/3/01	7
9	RCC in columns	15 days	Thu 10/4/01	Wed 10/24/01	8
10	RCC in beams and slabs	30 days	Thu 10/25/01	Wed 12/5/01	9
11	Well digging	10 days	Mon 4/16/01	Fri 4/27/01	
12	Brickwork for well	10 days	Mon 4/30/01	Fri 5/11/01	11
13	Brickwork for septic tank	2 days	Wed 4/25/01	Thu 4/26/01	1
14	Roof slab for septic tank	7 days	Fri 4/27/01	Mon 5/7/01	13
15	Brickwork for walls	20 days	Thu 12/6/01	Wed 1/2/02	10
16	Finishing of brick walls	20 days	Thu 1/3/02	Wed 1/30/02	15
17	Construction of water tank	2 days	Thu 12/6/01	Fri 12/7/01	10
18	Earthfilling in basement	5 days	Thu 7/26/01	Wed 8/1/01	6
19	Fixing of doors & windows	2 days	Thu 1/3/02	Fri 1/4/02	15
20	Colour washing & painting	20 days	Mon 1/7/02	Fri 2/1/02	15,17,18,19
21	PCC in floor & curing	5 days	Thu 8/2/01	Wed 8/8/01	18
22	Floor finishing, first stage	5 days	Thu 8/9/01	Wed 8/15/01	21
23	Floor finishing, final stage	5 days	Mon 2/4/02	Fri 2/8/02	20,22
24	Wiring and plumbing	15 days	Thu 1/3/02	Wed 1/23/02	15
25	Electrical fittings	2 days	Mon 2/4/02	Tue 2/5/02	20



**GANTT CHART FOR THE SMALL HOUSE CONSTRUCTION PROJECT
OUTPUT FROM MICROSOFT PROJECT PACKAGE**

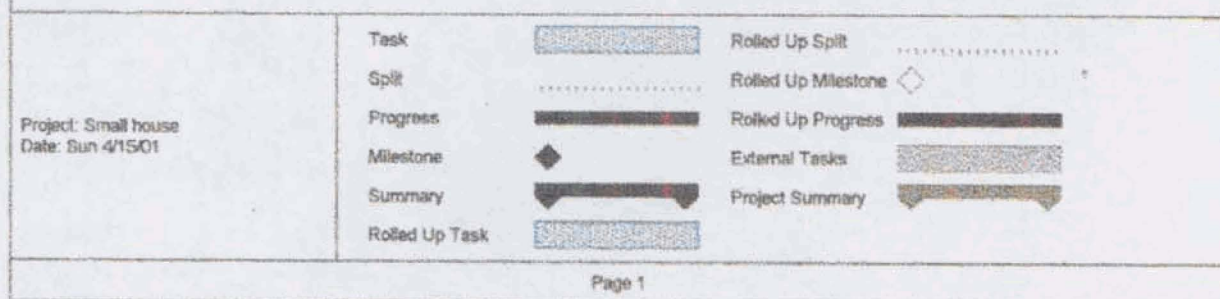


Fig 15.9

Gantt chart from Microsoft Project

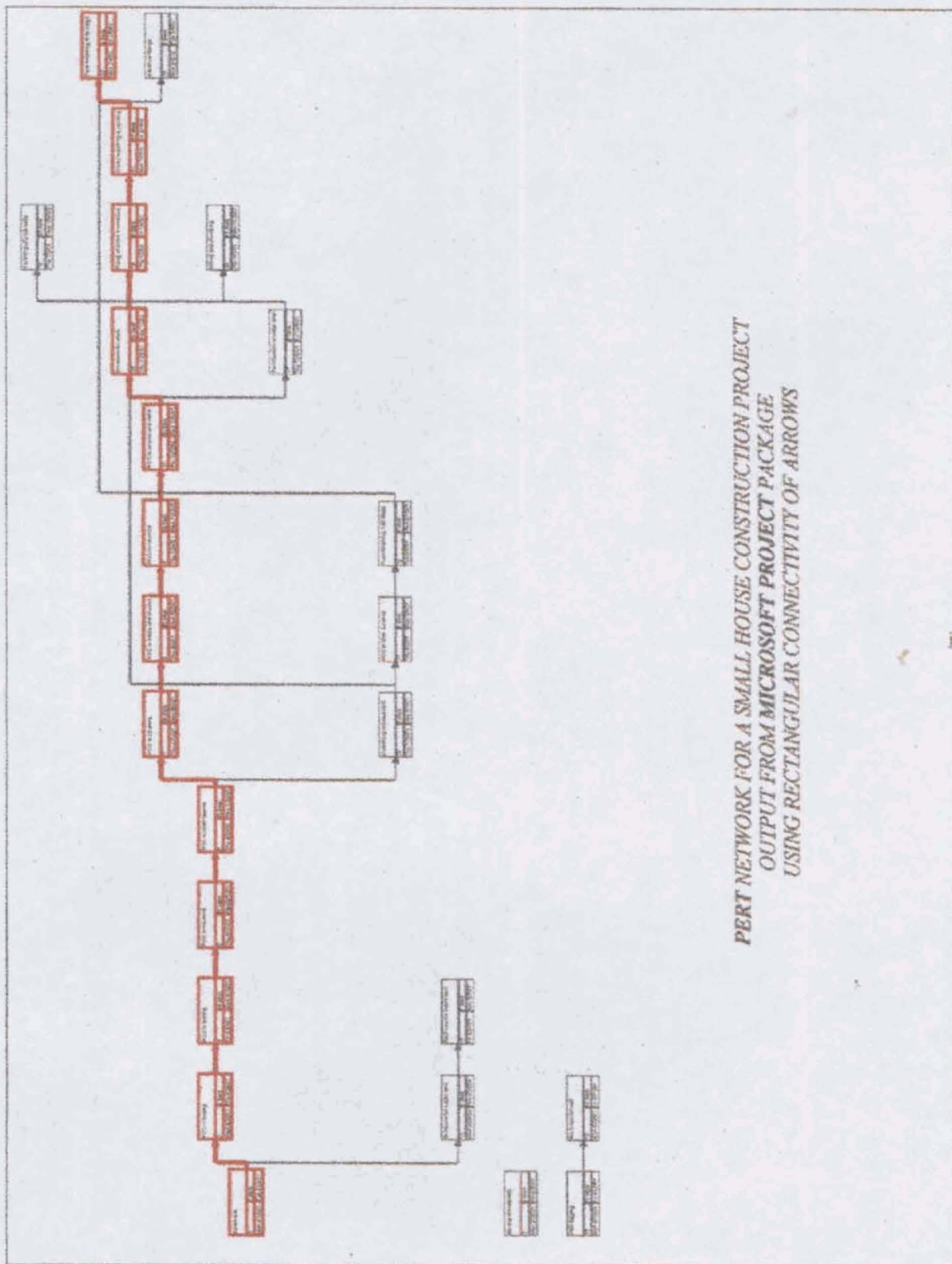


Fig 15.10(a) PERT chart output from *Microsoft Project* (rectangular connectivity)

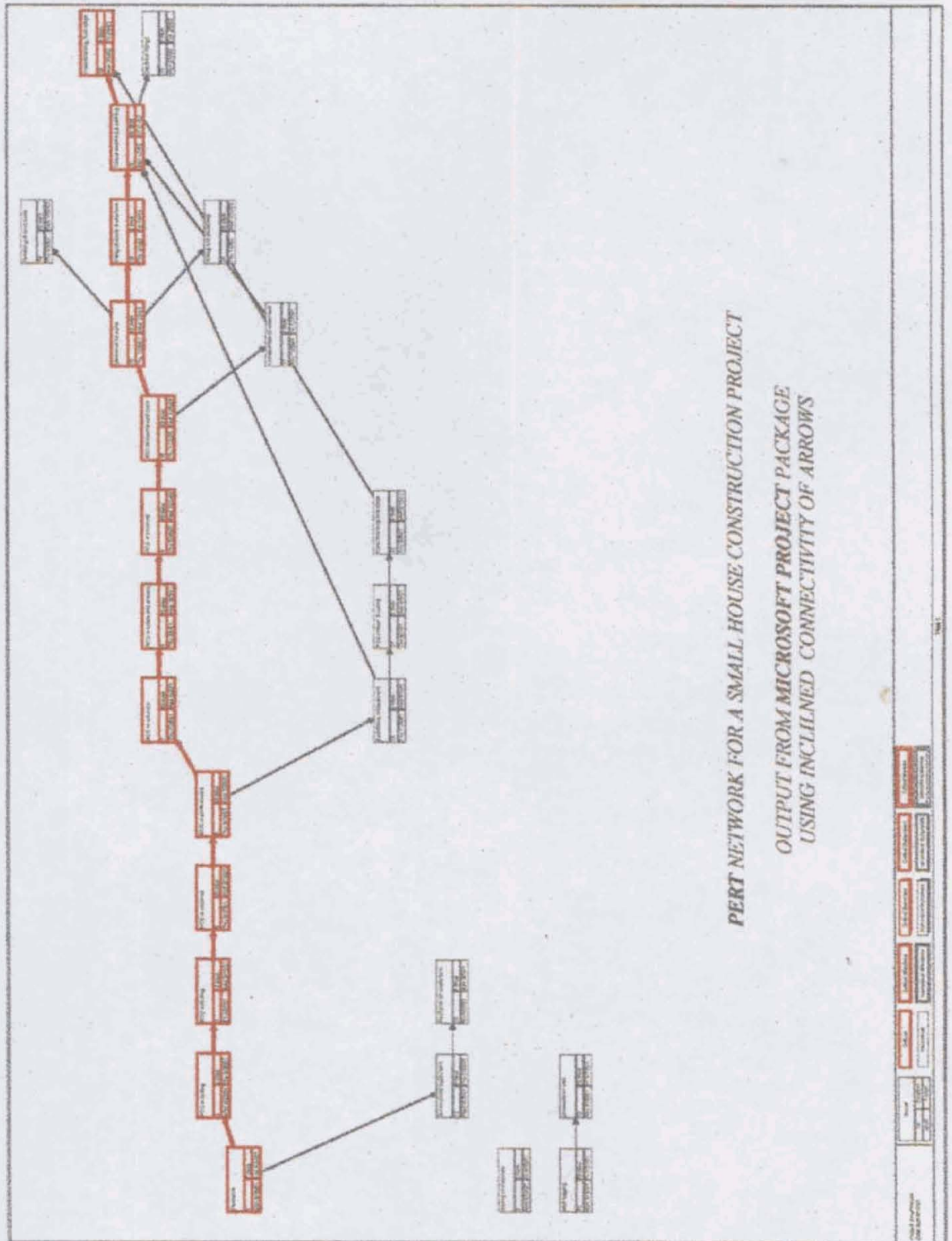


Fig 15.10(b) PERT chart output from *Microsoft Project* (inclined connectivity)

15.4 A comparison of *Primavera* and Microsoft *project* packages

Both *Primavera* and *Microsoft Project* have gained popularity among project planners because of their versatility and varying applications. By consistently using both the packages, the author has made the following observations.

- *Microsoft Project* is more users friendly and has more maneuverability for inexperienced users.
- For handling bulk data, *Primavera* is more helpful to the experienced users
- Though the number of views are more in *Microsoft Project*, most of them are tabular views, where as in *Primavera*, most of the views are graphical ones.
- *Primavera* permits the use of its data on *Microsoft Project* with out much alteration, where as *Microsoft Project* does not permit the transfer of data to *Primavera* in the same format as such.
- Facility for defining the graphical shapes is more in *Microsoft Project* compared to those in *Primavera*.
- Resource handling capabilities and output types are more in *Microsoft Project* and this helps in more interim reports.
- Both *Primavera* and *Microsoft Project* are more concerned about the most popular and basic scheduling techniques with out much mathematical complexities and they do not take care of advanced scheduling techniques like GERT, VERT etc.

TIME-STEP CHART

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 16

TIME-STEP CHART

16.1 Introduction

Time-step chart (TSC) is a new charting technique proposed by the author. This chart is intended to overcome the deficiencies of the Gantt chart, improving up on it and at the same time utilizing the advantages of the Gantt chart such as its simplicity and easiness of comprehension. At the same time, the positive aspects of the networking techniques, both probabilistic and deterministic are also accounted for in this new charting technique. Advantages of the network diagrams are not neglected and this chart is not basically to replace the network diagrams. Still, many of the positive features of the network diagrams are taken care of in the proposed *time-step chart*. The following deficiencies of the Gantt chart and network diagrams like CPM, PERT, GERT etc. have prompted the introduction of such a chart.

1. The Gantt chart doesn't give the interconnectivity of the activities. In the Gantt chart, each activity is taken as a discrete event and its start and end are represented individually, irrespective of the other activities. Hence the Gantt chart doesn't give an idea about the possibilities of one activity affecting another. Especially to a person, who is not familiar with the nature of activities in a project, the Gantt chart doesn't convey the message and the information provided is many times misleading. Though as a simple form of representation it is acceptable, in a comprehensive project planning and analysis, because of these inadequacies, the Gantt chart many-a-time becomes insufficient or rather inadequate.
2. Though the network diagrams like CPM, PERT, GERT, VERT etc. do give the interconnectivity of the activities, for a big project, the respective diagrams become too complicated to draw, because of the interconnecting arrows. Though software packages are available to draw such diagrams, too many activities make the diagram to be spread on many pages in the output of the commonly available software packages. When we try to contain this on a single page, many times the software do not permit compression beyond a

limit. In case the compression is permitted, the readability of the data becomes too difficult. Since the available software do have inbuilt systems for networking, they need not necessarily optimize the space requirement of the diagram. This make the network representation rather complicated in the case of large networks. The network connectivity in many cases is rather too difficult to do manually if the activities are more than that could be easily accommodated. Experience in making the networks also counts in reducing the bulk of the diagram.

3. Both the Gantt chart and the network diagram do not give any indication about the nature of the activity as to whether it is deterministic or probabilistic. Though it may not seem that important at first to determine the nature of the activity, at the time of computation of the total duration of a project, this is of utmost importance, since the near critical activities may cause the delay of the project, much more than the critical activities. Moreover, just by seeing a network, an onlooker is not in a position to judge the underlying implications of the probabilistic nature of activities, if any, in the project. In real experience, it has been pointed out by many that the nature of the activity is more important than the duration of the activity in deciding the completion time of the activity. Even though the probabilistic networks do take into account this factor, its implication is never let known in the simple network, which is the basis for all further calculations. Though PAN and GERT do consider the probabilistic nature of the activities, when it comes to simple networks like CPM and PERT, the nature of the activity is not being properly taken care of.
4. For more number of activities, it is very difficult to make a compact Gantt chart or PERT chart. The compactness will make the activity descriptions illegible, as seen in Fig.15.10. The chart or diagram can be split into different pieces to avoid this difficulty, but then the total picture and message of the diagram is being lost. Since most of the managers are interested in getting a totalistic view of the project, the split diagram does not suffice the requirement. Compressing the diagram becomes virtually a task, when the

interconnectivity of the activities is multi-linked (one activity having links with many activities) and cannot be contained in less number of arrows. The software, if any, used for the preparation of the diagram need not necessarily consider the bulk of the interconnectivity in the layout of the network diagram.

The legends and application of the Time-step chart are explained in Figs 16.1 and 16.2, with the help of a small house construction project.

The legends of the *TSC* can be verbally explained as follows.

The basic diagram consists of two rectangles, the lower one with more height compared to the upper rectangle. The upper rectangle is inscribed with a number representing the duration of the project and the lower rectangle will have either a circle or rhombus inside it. The circle represents that the activity is deterministic and the rhombus represents that it is probabilistic. The circle or rhombus is inscribed with a number representing the activity. A triangle at the top of this rectangle, with the base on this rectangle and pointing towards top represents the following activity, if any, with its number written inside the triangle and a triangle at the bottom of this rectangle, with base at top and pointing downwards represents the preceding activity, if any, with its number written inside it. In case the preceding and following activities are more than one, the numbers of all these activities will be written inside these triangles. If there is no following activity, the upper triangle will be absent and if there is no preceding activity, the lower triangle will be absent.

A vertical line on the right half of the upper rectangle indicates that the activity is critical. The start time of the activity is written at the base line of the bottom rectangle and when there are many activities, a base line with date written on the left end will act as the date indication line for the start time of these activities. The nodes in the *TSC* are not drawn to any scale and hence the top end of any node does not indicate the finish time.

Moreover, the sizes of the nodes can be chosen conveniently such that all the information to be provided in the node can be legibly given in it. Still, it is advisable to provide all nodes of the same size for a uniform appearance. When the *TSC* appears on the same sheet, the uniform size of the nodes becomes rather a requirement.

Standard node combinations may be any of the following cases:

1. Two rectangles with the bottom one having more height and a circle inside this lower rectangle. This node indicates that there is no following or preceding activity for the activity under consideration. The inside circle indicates that the nature of the activity under consideration is deterministic. The number written inside the circle indicates the number of the activity under consideration.
2. Two rectangles with the bottom one having more height and a rhombus inside this lower rectangle. This node indicates that there is no following or preceding activity for the activity under consideration. The inside rhombus indicates that the nature of the activity under consideration is probabilistic. The number written inside the rhombus indicates the number of the activity under consideration.
3. Two rectangles with the bottom one having more height and a circle inside this lower rectangle and a triangle at bottom having its base common with the base of the bottom rectangle and pointing downwards. This node indicates that there is no following activity, but there are preceding activities for the activity under consideration. The inside circle indicates that the nature of the activity under consideration is deterministic. The number written inside the circle indicates the number of the activity under consideration. The numbers written inside the bottom triangle indicates the numbers of the preceding activities.
4. Two rectangles with the bottom one having more height and a rhombus inside this lower rectangle and a triangle at bottom having its base common with the base of the bottom rectangle and pointing downwards. This node indicates that there is no following

activity, but there are preceding activities for the activity under consideration. The inside rhombus indicates that the nature of the activity under consideration is probabilistic. The number written inside the rhombus indicates the number of the activity under consideration. The numbers written inside the bottom triangle indicates the numbers of the preceding activities.

5. Two rectangles with the bottom one having more height and a circle inside this lower rectangle and two triangles, one at bottom having its base common with the base of the bottom rectangle and pointing downwards and another at top, having its base common with the top of the upper rectangle and pointing towards up. This node indicates that there are following and preceding activities for the activity under consideration. The inside circle indicates that the nature of the activity under consideration is deterministic. The number written inside the circle indicates the number of the activity under consideration. The numbers written inside the bottom triangle indicates the numbers of the preceding activities and the numbers written inside the top triangle indicates the numbers of the following activities.

6. Two rectangles with the bottom one having more height and a rhombus inside this lower rectangle and two triangles, one at bottom having its base common with the base of the bottom rectangle and pointing downwards and another at top, having its base common with the top of the upper rectangle and pointing towards up. This node indicates that there are following and preceding activities for the activity under consideration. The inside rhombus indicates that the nature of the activity under consideration is probabilistic. The number written inside the rhombus indicates the number of the activity under consideration. The numbers written inside the bottom triangle indicates the numbers of the preceding activities and the numbers written inside the top triangle indicates the numbers of the following activities.

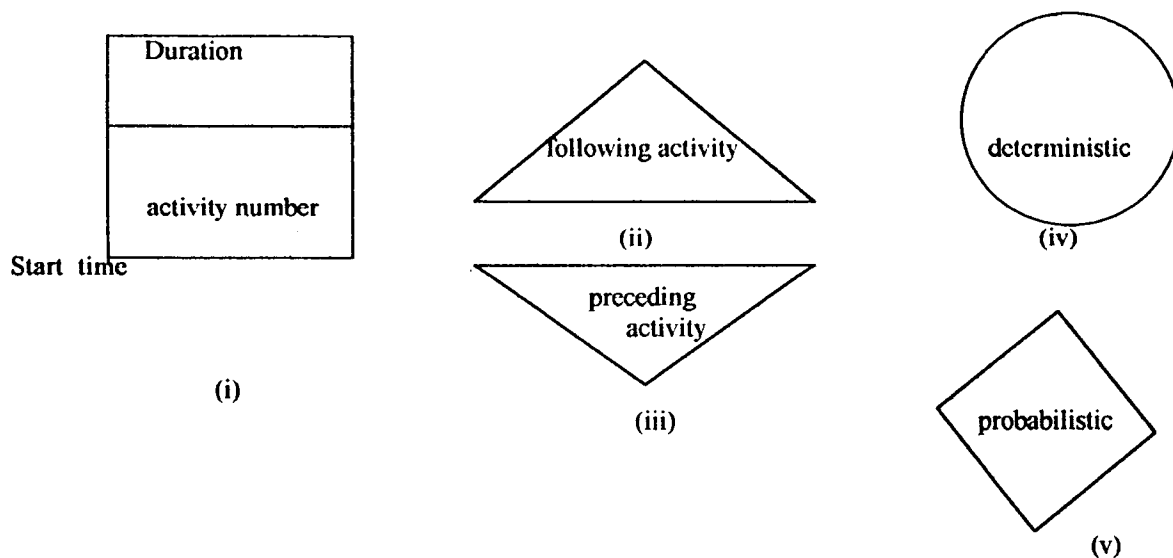
7. This case is for specifically identifying the critical activities. In any of the cases given above, if the upper rectangle is having a vertical line from base to top, in the right

half, it represents a critical activity. In any of the above combinations of nodes, this representation can be added without in any way affecting the configuration of the node.

As an illustrative example for drawing a *TSC*, of a small house construction project is taken with the list of activities and details as given in Table 15.4. The legends used for drawing the *TSC* are given in Fig 16.1. The *TSC* for this case is given in Fig 16.2.

To understand the *TSC* given in Fig 16.2, the following information will be useful.

- # There are 25 activities in this small house construction project, represented by 25 nodes.
- # Starting dates of the activities are as given in Table 15.4 and this is represented on the base lines at the left end.
- # Activities 1, 2, 11, 12, 13, 14, 15, 18, 19, 20, 24, 25 are deterministic in nature and the remaining activities are probabilistic.
- # Activity 2 has no following and preceding activities.
- # Activities 23, 24 and 25 have no following activities.
- # Activity 15 has got four following activities, 16, 19, 20 and 24.
- # Activity 1 has a following activity, 3.
- # Activity 3 has a preceding activity, 1.
- # Activities 1, 3, 4, 5, 6, 7, 8, 9, 10, 15, 19, 20 and 23 are critical activities.
- # The project begins on 16/04/01 and the last activity begins on 04/02/02.
- # Completion of activity 23 decides the completion time of the project.



5 TYPES OF NODES USED IN THE TIME STEP CHART

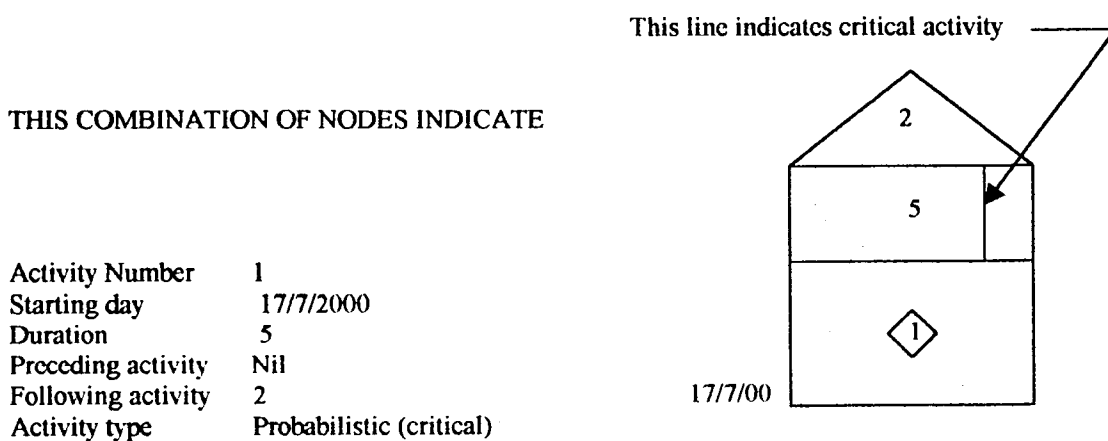
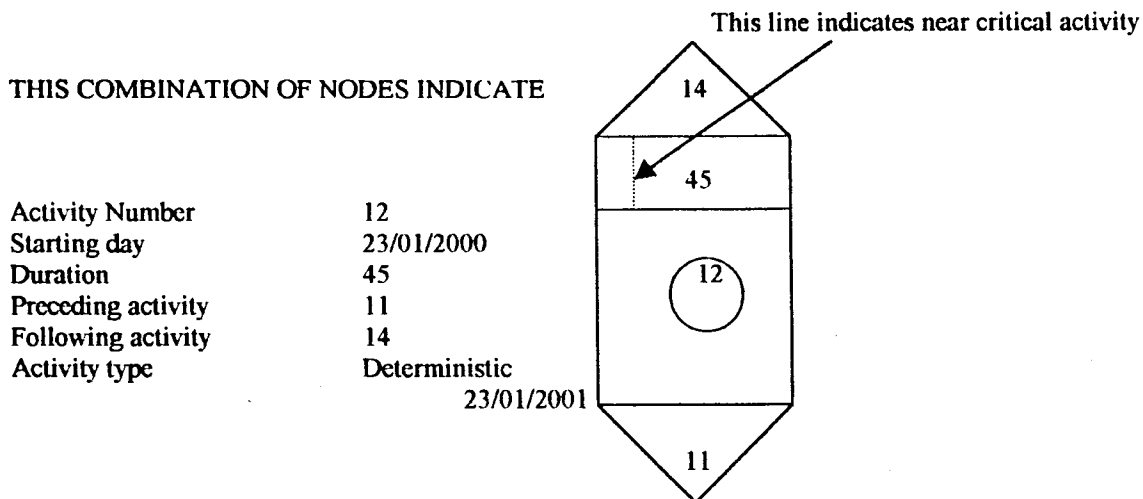


Fig 16.1 Legends of the Time Step Chart

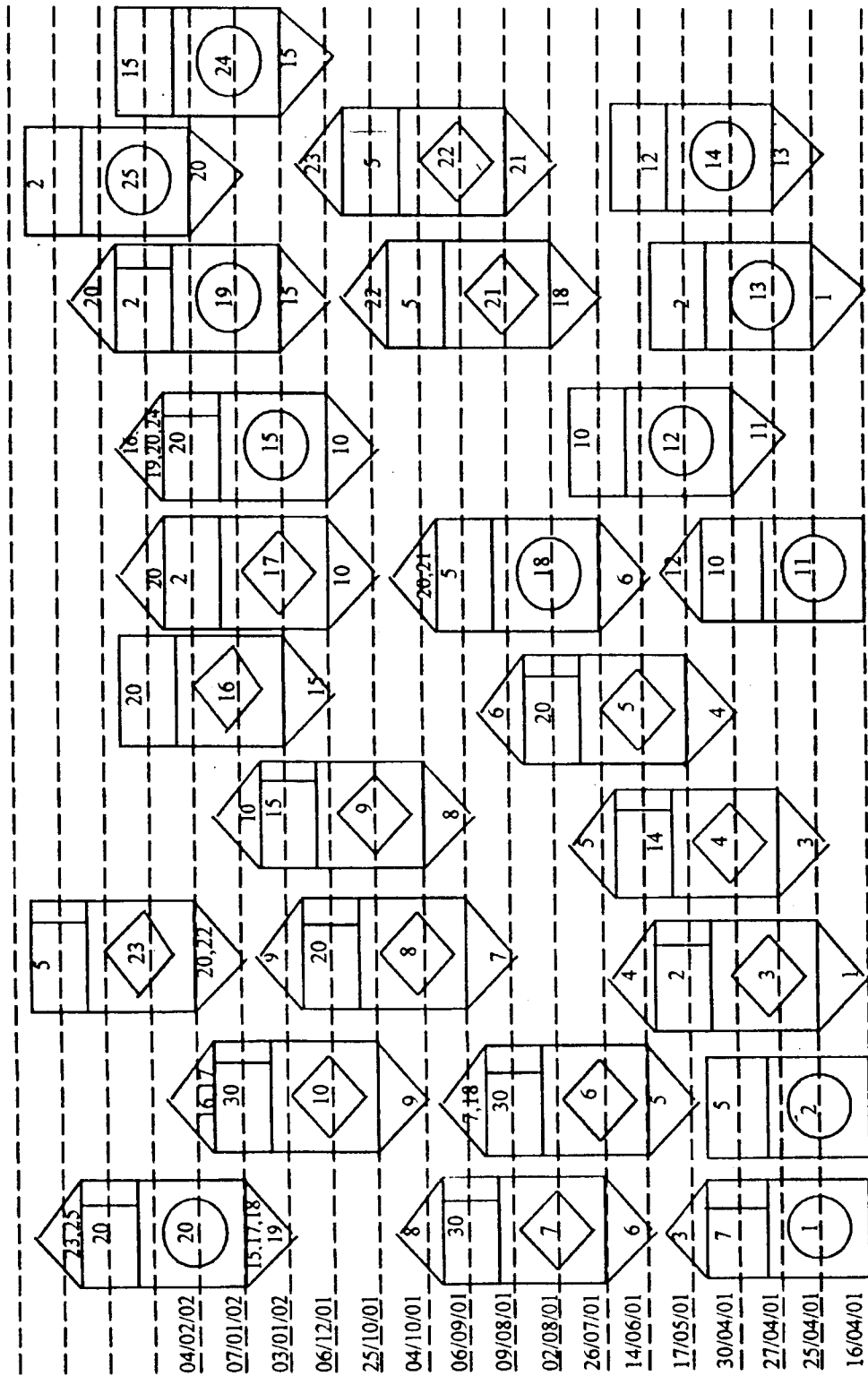


Fig 16.2 Time Step Chart for the small house construction project in Table 15.4

16.2 How to draw the *Time-step chart*?

- Draw the base lines on a sheet of paper, such that the earliest date is on the bottom most line.
- Mark the corresponding date on each base line
- Draw the rectangle for each activity such that the base of the rectangle rests on the base line corresponding to its start date
- After drawing all the rectangles, draw the triangles at the base and top of the rectangles, to represent the preceding and following activities.
- On the top and bottom triangles, inscribe the numbers corresponding to the following and preceding activities.
- At the middle of the bottom rectangle, inscribe a rhombus or circle to indicate whether the activity is probabilistic or deterministic.
- Inside the rhombus or circle, write the activity number.
- Inside the top rectangle, write the duration of the activity
- Find the critical activities by conducting a forward pass, starting from the first base line, as explained in article 16.4.
- Mark the critical activities by drawing a vertical line in the right half of the upper rectangle.
- Near critical paths can be denoted by a dotted vertical line on the left side of the upper rectangle.
- Rearrange the nodes such that the chart occupies the minimum space.

Standard node shapes can be made in stencils and the drawing of the chart can thus be made easy. Color combinations can be suitably used to indicate the nodes and inscriptions to make the chart more attractive. Once the chart is drawn, the node positions can be suitably adjusted to have a definite pattern and systematic look. The size of the nodes can be chosen based on the number of the activities and the overall size of the chart preferred. If the number of following and preceding activities are many and is difficult to

be contained in a triangular node element, a special character like @, #, * etc. can be given there instead of numbers and the corresponding numbers can be given as footnotes.

16.3 Determination of critical path in the TSC

- Start from one of the activities on the bottom most base line.
- Choose the following activity as indicated by the number on the upper triangle, if any.
- Proceed further, choosing the following activities, till there is a node with many following activities given.
- When there is more than one following activity, choose any one of them and proceed further, till the final activity in the chain is reached.
- Add up the durations of all the activities in the chain.
- Take the next chain of activities, with the other branch chosen at the branching point.
- Similarly proceed with the further activities on the base line, such that all the possible chains of activities are taken care of.
- The chain with the highest duration is taken as the critical chain and all the activities on this chain are taken as the critical activities.
- The chain with the second highest duration is taken as the near critical path and the activities on this chain are taken as near critical activities.

As an example, for the *TSC* given in Fig 16.2, finding the critical path is done as follows.

Start with activity 1.

The following activity is 3.

Activity 4 is following the activity 3.

Activities 5 and 6 follow activity 3.

Activity 6 is followed by two activities, 7 and 18.

Take the branch starting with activity 7 and proceed.

Thus we get the following chains of activities starting with activity 1.

1-3-4-5-6-7-8-9-10-16.

1-3-4-5-6-7-8-9-10-17-20-23.

1-3-4-5-6-7-8-9-10-17-20-25.

1-3-4-5-6-7-8-9-10-15-16.

1-3-4-5-6-7-8-9-10-15-19-20-23.

1-3-4-5-6-7-8-9-10-15-19-20-25.

It is seen that since activity 2 has no following activity, the chain doesn't form.

The chain starting with activity 11 also ends with activity 12 and hence no further chain follows.

Thus, for all the chains available, add up the durations of the individual activities and find the chain with the highest duration. In this example, the chain with the highest duration is 1-3-4-5-6-7-8-9-10-15-19-20-23. Hence, all the activities on this chain can be marked as critical activities. The chain with the next higher duration is 1-3-4-5-6-7-8-9-10-15-19-20-25. The only activity in this chain, not coming in the list of critical activities is activity 25 and hence this is the only near critical activity in this project. Since activity 25 is of deterministic nature and of duration 2 only, whereas activity 23 is of probabilistic nature and duration 5, the effect of the only near critical activity can be ignored in this case, since it is not going to affect the total project duration.

Adding the durations of the activities in the chain 1-3-4-5-6-7-8-9-10-15-19-20-23, we get the total duration as

$$7 + 2 + 14 + 20 + 30 + 30 + 20 + 15 + 30 + 20 + 2 + 20 + 5 = 215.$$

The duration of the near critical path 1-3-4-5-6-7-8-9-10-15-19-20-25 is

$$7 + 2 + 14 + 20 + 30 + 30 + 20 + 15 + 30 + 20 + 2 + 20 + 2 = 212.$$

In this case, it is not necessary to find the duration of the other chains, since all of them are incomplete chains and their durations do not come near the duration of the critical path.

16.4 Advantages of the *Time-step chart*

1. *All possible information about activities are available from the chart itself:*
The chart clearly indicates all the necessary information about each activity such as the early start time, early finish time, duration of the activity, nature of the activity, preceding activity, following activity, whether the activity is critical or not, and the identification number of the activity. All these information are contained in the node itself, which represent the activity in full. Since the typical shape of the node itself identifies the above parameters, an experienced user can get an overall idea about the activity from the first look itself, with out going through the legends. Even though legends are not necessarily familiar, it can be easily recognized, because of simplicity.
2. *Correlation of activities are shown without complicated networks:* Since the correlation of activities are shown by relation to a base line and by the numbers written on the node, with out verbal explanation one can easily bring out the inter-connectivity and correlation of the activities. Since there are no interconnecting lines used, the congestion in the network caused by the bulk of the lines is avoided. Since the connectivity is indicated by the numbers of the preceding and following activities as also the dates of commencement are shown by the corresponding date line on the *time-step chart*, the brevity of the chart is enhanced.
3. *Probabilistic and deterministic activities are easily identified in the chart:*
The shape of the node directly indicates whether the activity is deterministic or probabilistic. This reduces the speculations on the outcome of the activity's interaction with other activities and simultaneously provides us a clear vision as to what are the steps necessary for the control of the activity in time and cost.

4. *Highly compact nature of chart makes handling easier:* Since every project manager is interested in getting a comprehensive report on the status of the project, bulky reports tend to reduce the acceptability. The very compact nature of the *time-step chart* makes it handy and preferable for producing the project summary. The interconnecting arrows in the network diagram prevents it from compressing beyond a limit and in the case of the *TSC*, there is no such restricting factor as also there is no minimum size to which the chart can be compressed. The only deciding factor is the readability of the numbers that are written on the nodes. Moreover, the chart can be made either vertical or horizontal, based on the type and nature of the project or the report required.
5. *TSC incorporates all the advantages of Gantt, CPM/PERT, GERT and VERT charts/networks:* The basic pattern for the *TSC* is chosen similar to that of the Gantt chart. The probability indications are similar to that of the GERT chart. The interconnectivity is in many ways similar to the CPM/PERT chart. The box type notations have a resemblance to that of the VERT nodes, though the VERT nodes have logical choices at all the steps, which is necessarily not a feature of the *TSC* nodes.
6. *The PERT mathematics can very well be applied to TSC also:* The equations used in the PERT calculations for optimistic project completion time, pessimistic project completion time, expected project completion time, standard deviation and variance can be applied in the case of *TSC* also without any modification, since the *TSC* is compatible to all the mathematical calculations involved in PERT. Since all the nodes in the *TSC* are provided with the basic mathematical information required regarding a node, the computation of the activity variables will not be difficult at all and with the preliminary information of the legends attached, one will be able to do the mathematical calculations.
7. *TSC is very easy for understanding and drawing:* The drawing and understanding of the *TSC* is very easy, since complicated networking is not involved. *TSC* is developed with the philosophy that the control charts in any

project should be easily discernible by the supervisory staff and not engineers alone. Drawing the *TSC* is possible by the supervisory staff also and hence comprehension is not that difficult. The fundamental knowledge of any charting technique is enough to understand the *TSC*.

8. *Critical activities are immediately identified from the chart itself:* In *TSC* the critical activities are marked by a special identification mark and it doesn't require any effort to identify them. In the CPM/PERT networks also, special identification marks like bold line, double line etc. are used for identifying the critical path in both the AON/AOA diagrams. This specialty of the *TSC* is hence very similar to the other network diagrams cited and is simpler in representation.

16.5 Comparison of Gantt chart with *TSC*

Features of the Gantt chart

- Activity nature – whether probabilistic or deterministic – is not identified from the chart.
- Following and preceding activities for a particular activity are not shown in the chart.
- Interconnection of activities are not specified in the chart
- Critical activities are not known from the chart.
- The chart is continuous and more space is required to contain it.
- The chart should be totally drawn in a suitable scale and hence compressing it beyond a level is not possible.
- Even if there are no activities in a particular period, it has to be included in the chart, since the chart is drawn for the total duration of the project.

Features of the TSC

- Activity nature – whether probabilistic or deterministic – is easily identified from the chart.
- Following and preceding activities for a particular activity are digitally shown in the chart.
- Interconnection of activities is specified in the chart by digits indicating following and preceding activities.
- Critical activities are marked in the chart.
- The chart is discrete in nature and hence minimum space is required to contain it.
- The chart doesn't require a specific scale and hence compressing it to any level is possible.
- If there are no activities in a particular period, that period can be excluded from the chart, since the chart is drawn only for the specific duration of the project, when activities do start.

16.6 Comparison of the CPM/PERT network with TSC

Features of the CPM/PERT network

- Too many arrows may cross each other in the network diagram, making the diagram complicated.
- For large number of activities, drawing the diagram requires much time consuming trial and error.
- For a large number of activities, drawing the diagram is a Herculean task as the crossing arrows will not be easily discernible.
- When the activity number is large, it will be difficult to show the duration, early start/finish etc. on the diagram, since the arrows may hide it.

- Too many activities will necessitate much space for drawing the network and it may not be possible to contain it in a sheet.

Features of the TSC

- Absence of any crossing arrow in the TSC makes it very easy to draw and comprehend.
- For a project of any number of activities, drawing the TSC requires only a few minutes.
- Since the arrows are absent, the nodes will be clearly discernible and visual complication is avoided.
- Even when the activity number is large, it will not be difficult to show the duration, following/preceding activities etc. on the TSC since the absence of arrows help in easily locating all the nodes.
- Any number of activities does not necessitate much space for drawing the TSC and it is possible to contain a good number of activities in a single sheet.

16.7 An example for comparing the CPM/PERT network with TSC

Consider an example for comparing the CPM/PERT network with the TSC. An example taken here is that of the construction of a brick compound wall with top fencing for a security area. The project consists of 17 activities and 2 dummies as given in Table 16.1

Table 16.1 List of activities for the construction of a brick compound wall with top fencing for a security area.

Sl.No.	Activity	Duration (days)	Preceding activities	Following activities
1.	Layout area	2	-	2,3,4
2.	Procure sand	2	1	6,10
3.	Procure rubble	2	1	5
4.	Earthwork excavation for foundation	4	1	7
5.	Procure cement	1	3	7
6.	Dummy	0	2	7
7.	RR work in foundation	10	4,5,6	8,9
8.	Brick work and curing for wall	10	7	11,12,14
9.	Pointing RR work	3	7	13
10.	Procuring steel gate	4	2	17
11.	RCC for gate pillar	1	8	17
12.	Dummy	0	8	13
13.	Plastering and curing brick wall	7	9,12	15,16
14.	Fixing top fencing	7	8	18
15.	White washing brick wall	2	13	19
16.	Fixing steel gates	1	10	19
17.	Painting top fencing	2	14	20
18.	Painting steel gates	1	16,17	20
19.	Clean up	2	18,19	-

The two dummy activities are included in the list for the sake of drawing the CPM/PERT network diagram. These dummies are not essential when using the *TSC*. Still, for the sake of total comparison, these dummies are also duplicated in the *TSC*. The CPM/PERT network for this project is shown in Fig. 16.3

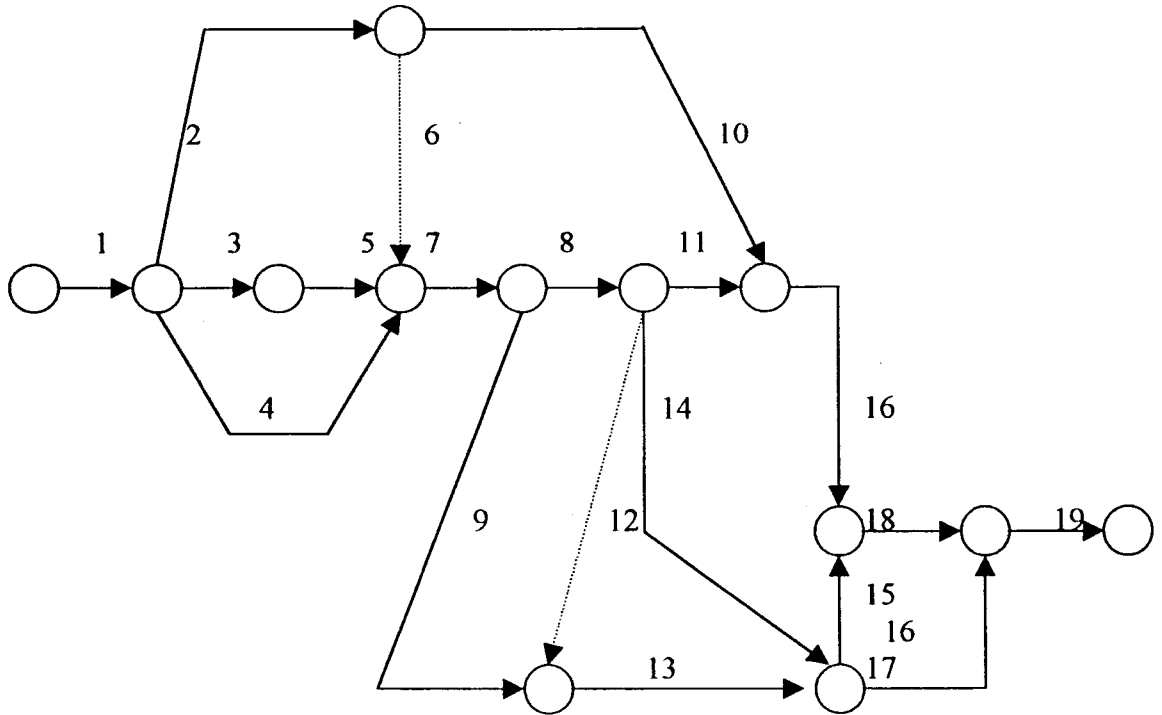
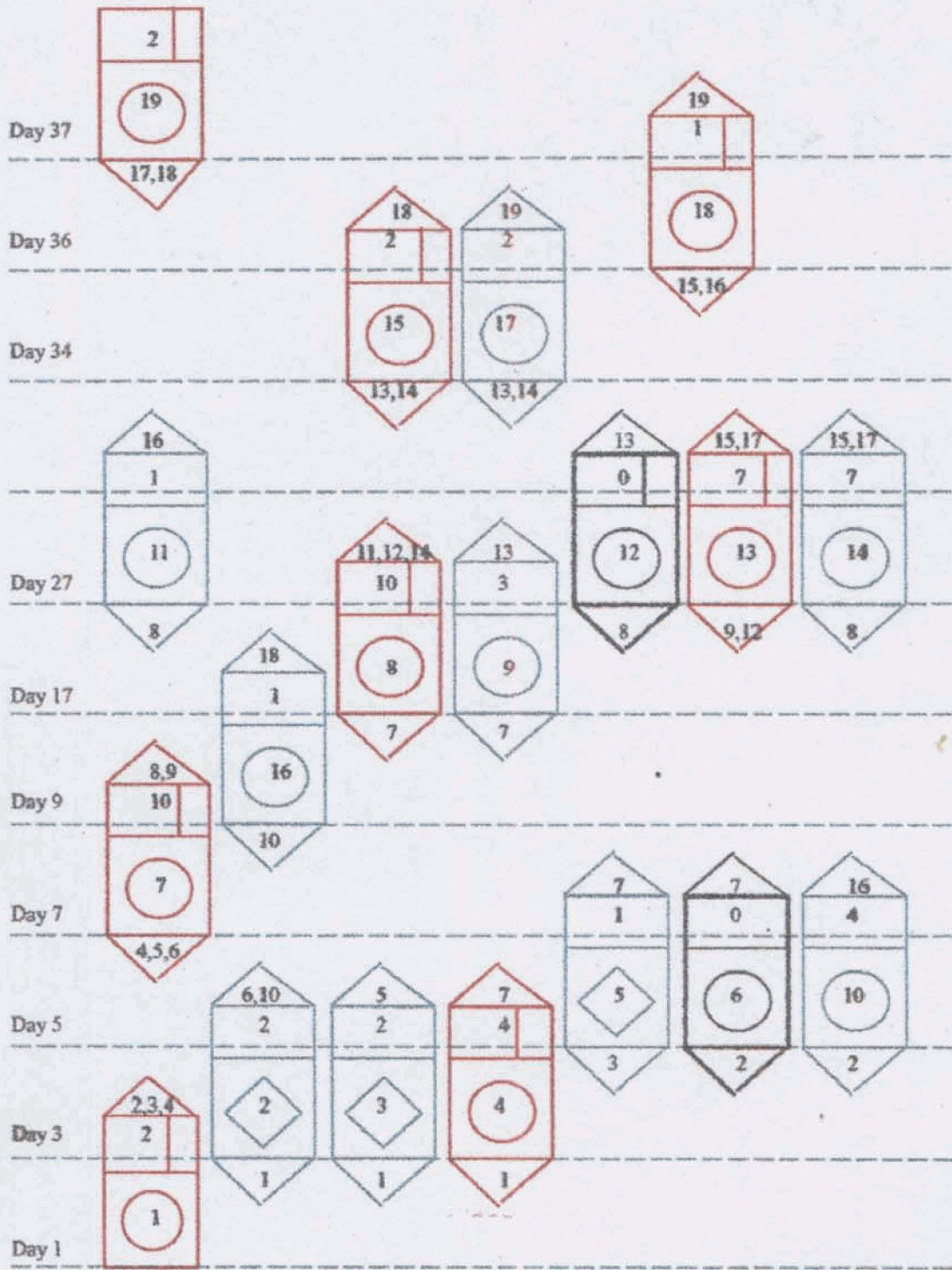


Fig.16.3 Activity-on-arrow network for the construction of a brick compound wall with top fencing for a security area (Table 16.1).

TSC for the same project is shown in Fig 16.4. The critical activities in the *TSC* are shown by red colour in addition to the usual identification of the vertical line in the right half of the upper rectangle. Dummy activities in the critical path are shown by blue colour and other dummy activities are shown by brown colour.



Red colour indicates critical activities. Brown colour indicates other dummies.
 Aqua colour indicates non-critical activities.

Fig.16.4 TSC for the construction of a brick compound wall with top fencing for a security area (Table 16.1).

The critical path of the project is 1 – 4 – 7 – 8 – 12 – 13 – 16 – 19 – 20. The total duration of the project is 39 days. The *TSC* is drawn with colour combinations to demonstrate the use of coloured nodes in the chart. With different colours, the nature of the node can be easily understood. It should be noted that the *TSC* can be drawn even without the use of dummy activities, whereas, in the A-O-A network, the use of dummy is very necessary for the drawing of the network. The advantages of colouring the nodes in the *TSC* can be better utilized in big projects, where the number of activities is more. In Fig 16.3, all the time estimates of the activities in the network are not shown, as also the duration, so as not to complicate the network. As is the practice, the duration, time estimates, early and late finish times etc. can be provided in the network, though it may take more space for drawing the network.

CONCLUSION AND SUGGESTIONS FOR FURTHER WORK

P.R Sreemahadevan Pillai “A comprehensive study of project scheduling techniques and the introduction of new techniques” Thesis. Department of Civil Engineering, Regional Engineering College Calicut , University of Calicut, 2002

CHAPTER 17

CONCLUSION AND SUGGESTIONS FOR FURTHER WORK

17.1 Summary of the thesis

This thesis has analyzed in depth most of the project scheduling techniques. Some of these techniques are in vogue and others have not yet gained the attention of planners and schedulers. Methods like CPM and PERT have become very popular because of their easy maneuverability and software like *Primavera* and *Microsoft project* have gained popularity because of their adaptability to personal computers and user interactive nature. Much of the other techniques are not coming forward to replace these methods in the field, though they have gained acceptance in the academic circles. Methods like artificial neural network applications and genetic algorithm applications to project scheduling techniques are still in the offing.

While analyzing the 15 methods presented in this thesis, the popularly available methods have been treated in such a way that their complexities and deficiencies are brought forward much, since their advantages are known to most of the common users. In the case of not-that-familiar techniques, the very fundamentals also have been brought out to enable it to be digested easier. Methods like DCPM, MCS, LOB, GERT and VERT are very powerful and versatile, still they have not gained much acceptance because of the demand for a high level user. In the case of popular methods, no much examples have been worked out because it is already available in many text books, whereas in the case of new methods effort has been taken to introduce either the working logic or to work out an example itself. Since a GA scheduler was not available with the author, a typical

example for the GA application could not be worked out, rather excerpts from a simple model have been provided.

The software packages *Primavera* and *Microsoft project* have been analyzed in much detail because of the reasons that a comparative study on these packages are not available to the knowledge of the author and moreover these packages are very popular with the planners abroad, though in India their utility is limited. Since artificial neural network and genetic algorithm applications are new subjects to the author, the scope of their application could be brought out only in a limited form. The fuzzy set theory and fuzzy expert system applications have been extensively used in many other civil engineering applications though their use in project scheduling is yet a new subject of study. The number of research papers available in this field are also very less.

The major contributions of the author in this thesis are the following

1. A comprehensive analysis of the existing methods of project scheduling like Gantt chart, CPM, PERT, DCPM, MCS, LOB, Petri nets, GERT and VERT, bringing out their strength, weakness, versatility, theoretical formulations, missing links, possible areas of application etc.
2. An insight into the new techniques of project scheduling like fuzzy set theory, fuzzy expert systems, artificial neural networks and genetic

algorithms with their fundamental formulations and scope for their further applications.

3. A comprehensive study and comparison of the most recent and popular software used in project scheduling with their competitive advantages.
4. Introduction of a new method of project scheduling chart, which combines the advantage of most of the existing charts.
5. Comprehensive and up-to-date information on project scheduling techniques developed so far.

17.2 Suggestions for further work

Following further works are suggested

Development of software for the application of the following in project scheduling (software for the application of the first four in some other fields are already available)

- (i) Fuzzy set
- (ii) Fuzzy expert system
- (iii) Artificial neural networks
- (iv) Genetic algorithms

(v) Time-step chart

- Up gradation of the existing software for inclusion of techniques like GERT, VERT etc.

- Publication of a book containing all the scheduling methods analyzed here.
(Already begun by the author)

REFERENCE

1. Adel M. Salem and Salah E. Elmaghraby, "Optimal linear approximation in Project compression", *IIE transactions*, Vol. 16, No. 4, December 1984, pp. 339-347.
2. Alan B. Pritsker, "GERT networks", *The Production Engineer*, October 1968, pp. 499-506.
3. Ali Jaafari, "Criticism of CPM for project planning analysis", *Journal of construction engineering and management, ASCE*, Vol. 110, No. 2, June 1984, pp. 222-233.
4. Amir Tavakoli and Roger Riachi, "CPM use in ENR top 400 contractors", *Journal of management in engineering, ASCE*, Vol. 6, No. 3, July 1990, pp. 282-295.
5. Ali Touran and Edward P. Wiser, "Monte Carlo technique with correlated random variables", *Journal of construction engineering and management, ASCE*, Vol. 118, No. 2, June 1992, pp. 258-272.
6. Ang. A., " Analysis of activity networks under uncertainty", *Journal of Engg. Mech., ASCE*, Vol. 101, No. 4, 1975, pp. 373-387.
7. Anklesaria. K. P. and Z. Drezner, " A multivariate approach to estimating the completion time for PERT networks", *Journal of operations research society*, Vol. 37, No. 8, 1986, pp. 811-815.

8. Arnold H. Buss and Meir J. Rossenblatt, "Activity delay in stochastic project networks", *Operations Research*, Vol. 45, No. 1, January – February 1997, pp. 126-139.
9. Arora R.K. and R.K. Sachdeva, "Distributed simulation of activity networks", *Computers and operational research*, Vol. 15, No.2, 1988, pp. 179-188.
10. Arora R.K. and R.K. Sachdeva, "Distributed simulation of resource constrained project scheduling", *Computers and operational research*, Vol.16, No.4, 1989, pp. 295-304.
11. Baker B.M., "A network-flow algorithm for project selection", *Journal of the operations research society*, Vol. 35, No.9, September 1984, pp. 847-852
12. Behnam Malakooti and Ying Q. Zhou, "Feed-forward artificial networks for solving discrete multiple criteria decision making problems", *Management Science*, Vol. 40, No. 11, November 1994, pp. 1542-1561.
13. Bell. C.E. and J. Han, "A new heuristic solution method in resource constrained project scheduling", *Naval Research Logistics*, Vol. 38, 1991, pp. 315-331.
14. Berny. J, "A new distribution function for risk analysis", *Journal of the operational research society*, Vol. 40, No. 12, 1989, pp. 1121-1127.

15. Bilal. M. Ayub and Achintya Haldar, "Project scheduling using fuzzy set concepts", *Journal of construction engineering and management, ASCE*, Vol. 110, No. 2, June 1984, pp. 189-203.
16. Bowman. R.A. and Muckstadt. J.A., "Stochastic analysis of cyclic schedules" *Operations research*, Vol. 41, No.5, Sept./Oct. 1993, pp. 947-958.
17. Carlos F. Diaz and Fabian C. Hadipriono, "Nondeterministic networking methods", *Journal of construction engineering and management, ASCE*, Vol. 119, No.1, March 1993, pp. 40-57.
18. Catling. J. R and Westwood. J. B, "A simulation approach to cost estimation in the project contracting industry", *Journal of operational research society*, Vol. 36, No.4, 1985, pp. 289-296.
19. Chang.T.C., and William Ibbs, "Network resource allocation with support of a fuzzy expert system", *Journal of construction engineering and management, ASCE*, Vol. 116, No.2, June 1990, pp. 239-260
20. Chia-Yiu Maa and Michael A. Shanblatt, "A two-phase optimization neural network", *IEEE transactions on neural networks*, Vol. 3, No. 6, November 1992, pp. 1003-1009.

21. Chinnam. R. B and Kolarik. W. J, "Neural network based quality controllers for manufacturing system", *International journal of production research*, Vol. 35, No. 9, 1997, pp. 2601-2620.
22. Colin E. Bell, "Maintaining project networks in automated artificial intelligence planning", *Management science*, Vol.35, No. 10, Oct 1989, pp. 1192 – 1214.
23. Cook. T. M. and Jennings R. H. "Estimating a projects' completion time distribution using intelligent simulation methods", *Journal of Opl. Res. Soc.*, Vol. 30, No.12, 1979, pp. 1103-1108.
24. David Arditi and M. Zeki Albulak, "Line-of-balance scheduling in pavement construction", *Journal of construction engineering, ASCE*, Vol. 112, No. 3, September 1986, pp.411-424.
25. David. I. Cleland & William R. King, "System analysis and project management", *Mc Graw Hill International*, 1983.
26. Deckro. R. F and J.E. Hebert, " Resource constrained project crashing", *OMEGA*, Vol. 17, No. 1, 1989, pp. 69-79.
27. Demeulemeester.E., Dodin. B and Herroelen.W, " A random activity network generator", *Operations Research*, Vol. 41, No.5, Sept./Oct 1993, pp. 972 - 980.

28. Dimitri. G.G., "On the distribution of activity time in PERT ", *Journal of Operations Research Society*, Vol.39, No. 8, 1988, pp. 767 - 771
29. Dimitri. G. G., " Controlled alternative activity networks for project management', *European Journal of Operational Research*, Vol. 37,1988, pp. 336- 346
30. Dimitri. G.G., "A new approach to activity - time distribution in PERT", *Journal of Operations Research Society*, Vol.40, No.4, 1989, pp. 389 - 393
31. Dimokritos A. Panagiotopoulos, Robert W. Newcomb and Sanjeev K. Singh, "Planning with a functional neural-network architecture", *IEEE transactions on neural networks*, Vol. 10, No. 1, January 1999, pp.115-127.
32. Ditlevsen. O., "Narrow reliability bounds for structural systems", *J. Struct. Mech., ASCE*, 7(4), 1979, pp. 453-472.
33. Dodin. B. M. and S. E. Elmaghraby, "Approximating the criticality indices of the activities in PERT networks", *Management science*, Vol.31, No.2, 1985, pp. 207 - 223
34. Dodin.B., "Bounding the project completion time distribution in PERT networks", *Operations Research*, Vol. 33, No. 4, July/Aug. 1985, pp. 862-881

35. Douglas.A.S., "Systems analysis and the computer", *Journal of the operations research society*, Vol. 34, No. 8, 1983, Aug, pp. 779-785
36. Dumond J. and VA. Mabert, "Evaluating project scheduling and due date assignment procedures: an experimental analysis", *Management science*, Vol.34, No. 1, Jan 1988, pp. 101 - 118
37. Edmund. N, Chrzanowski Jr. and David W. Johnston, "Application of linear scheduling", *Journal of construction engineering", ASCE* , Vol. 112, No.4, Dec. 1986, pp. 476-491.
38. Elsayed. E. A. and Nasr. N. Z., "Heuristics for resource-constrained scheduling", *International journal of production research*, Vol. 24, No. 2, 1986, pp. 299-310.
39. Erik Demeulemeester, " Minimizing resource availability costs in time-limited project networks", *Management Science*, Vol. 41, No. 10, October 1995, pp. 1590-1598.
40. Fadi. A. Karaa and Anas.Y. Nasr, "Resource Management in construction", *Journal of construction engineering", ASCE*, Vol. 112, No.3, Sept 1986, pp. 346-357.
41. Francis. M. Martin, "Computer modeling and simulation", *John Wiley & Sons Inc.*, New York, 1968.
42. Fred. J. Maryanski, "Digital computer simulation", *CBS publishers and Distributors*, Delhi - 110032, 1980.
43. Geoffrey Gordon, "System simulation", *Prentice Hall of India Pvt. Ltd.*, New Delhi-1, 1989.

44. Grant. F.H., "A note on efficiency of the antithetic variate method for simulating stochastic networks", *Management science*, Vol. 29, No. 3, 1983, March, pp. 381-384.
45. Habson. R., "Time and cost securing in complex large scale projects", *Project Management*, Vol. 3, August 1985, pp. 178-181.
46. Handa. V. K. and Barcia. R. M., "Linear scheduling using optimal control theory", *Journal of construction engineering, ASCE*, Vol. 112, No. 3, September 1986, pp. 387-393.
47. Heng Li and Peter Love, "Using improved genetic algorithms to facilitate time-cost optimization", *Journal of construction engineering and management, ASCE*, Vol. 123, No.3, September 1997, pp. 233-236.
48. Hira. N. Ahuja and V. Nandakumar, "Simulation model to forecast project completion time", *Journal of construction engineering and management, ASCE*, Vol. III, No.4, Dec. 1985, pp. 325-389.
49. Hoffmann. T.R. and G.D. Scudder, "Priority scheduling with cost consideration", *International Journal of Production Research*, Vol.21, No.6, 1983, pp. 881-889.
50. Hojjat Adeli and Asim Karim, "Scheduling/cost optimization and neural dynamics model for construction", *Journal of construction engineering and management, ASCE*, Vol.123, No.4, December 1997, pp.450-458

51. Hughes. M. W., "Why projects fail, the effect of ignoring the obvious", *Industrial Engineering*, Vol. 18, No. 4, April 1986, pp. 14 -18.
52. Igelmund. O. and Radermacher. F. J., "Pre selective strategies for the optimization of stochastic project networks under resource constraints", *Networks*, Vol. 13, No.1, April 1983, pp. 1-28.
53. Jason Portas and Simaan Abou Rizk, "Neural network model for estimating construction productivity", *Journal of construction engineering and management, ASCE*, Vol. 123, No. 4, December 1997, pp. 399-410.
54. Jerome. D. Wiest and Ferdinand. K. Levy, "A management guide to PERT / CPM", *Prentice Hall of India Pvt. Ltd.*, New Delhi -1, 1985.
55. John Caulfield and Jason M. Kinser, "Finding the shortest path in the shortest time using PCNN's", *IEEE transactions on neural networks*, Vol. 10, No. 3, May 1999, pp. 604-606.
56. Jonathan F. Bard and Jeanne L. Miller, "Probabilistic shortest path problems with budgetary constraints", *Computers and operations research*, Vol. 16, No. 2, 1989, pp. 145-159.

57. Julian Rietman, "Computer simulation applications", *John Wiley & Sons Inc.*, New York, 1971.
58. Keefer. D. L. and Verdini. W. A., "Better estimation of PERT activity time parameters", *Management science*, Vol. 39, No.9, September 1993, pp. 1086-1091.
59. Kent Davis, W. B. Ledbetter and James L. Burati., "Measuring design and construction quality costs", *Journal of construction Engineering & Management, ASCE*, Vol. 115, No.3, Sept. 1989, pp. 385-400.
60. Keys. P., "MS / OR Projects and their interaction", *OMEGA*, Vol. 17, No.2, 1989, pp. 113-122.
61. Kidd.J., "Project Analysis Today- The end users' Disquiet?", *OMEGA*, Vol. 17, No.2, 1989, pp. 103 –111.
62. Kulkarni. V. G., "A compact hash function for paths in PERT networks", *Operations Research Letters*, Vol. 3, Aug. 1984, pp. 137-140.
63. Kulkarni. V. G. and Adalakha. V. G., "Markov and Markov regeneration PERT networks", *Opns. Research*, Vol. 34, No.5, 1986, pp 769-781.
64. Kurtulus. I. S. and S.C. Narula, " Multi Project scheduling, Analysis of project performance", *IIE Transactions*, Vol. 17, No. 1, 1985, pp. 58-66.

65. Leonhard. E. Bernold, "Simulation of non-steady construction processes", *Journal of construction engg. & management, ASCE*, Vol.115, No.2, June 1989, pp. 163-169.
66. Levine S. H. and E. Romanoff, "Economic impact dynamics of complex engineering project scheduling", *IEEE transactions on systems*, Vol. 19, No.2, March/April, 1989, pp. 232-240.
67. Li. K. Y and Wills. R. J., "An interactive scheduling technique for resource constrained project scheduling", *European Journal of operations Research*, Vol. 56, No. 3, Feb. 1992, pp. 370-379.
68. Lootsma. F. A., "Stochastic and fuzzy PERT", *European Journal of Operational Research*, Vol. 43, 1989, pp. 174-183.
69. Mario I. Norbis and Mac Gregor Smith. J, "Two level heuristic for the resource constrained scheduling problem", *International journal of production research*, Vol. 24, No. 5, 1986, pp. 1203-1219.
70. Mario I. Norbis and Macgregor Smith. J, "A multi-objective multilevel heuristic for dynamic resource constrained scheduling problems", *European journal of operational research*, Vol. 33, 1988, pp. 30-41.
71. Mehrez. A. & Regev. V., "Stochastic dominance relations in a project cost problem", *IIE Transactions*, Vol. 17, No. 2, June 1985, pp. 194 –197.

72. Mhatre. S., "Failures in computerized project management - an analysis", *NICMAR journal of construction management*, Volume IX, No.111, Oct. 1994, pp. 213-217.
73. Minghe Sun, Antonie Stam and Ralph E. Steuer, "Solving multiple objective programming problems using feed-forward artificial neural networks: the interactive FFANN procedure", *Management Science*, Vol. 42, No. 6, June 1996, pp. 835-849.
74. Mohanty. R.P. and M.K. Siddiq, "Multiple projects - multiple resources - constrained scheduling: some studies", *International Journal of Production Research*, Vol.27, No.2, 1989, pp. 261 -280.
75. Mohanty. R.P. and M.K. Siddiq, "Multiple projects - multiple resources constrained scheduling: a multi objective analysis", *Engineering costs and production economics*, Vol. 18, 1989, pp. 83-92.
76. Mohring R. H, "Minimizing costs of resource requirements in project network subject to a fixed completion time", *Operations research*, Vol. 32, No. 1, Jan/Feb. 1984, pp. 89 - 120.
77. Monhor. D, "On the application of concentration function to PERT", *Optimization*, Vol. 14, No.2, 1983, pp. 237-244

78. Mustafa K. Mehmet Ali and Faouzi Kamoun, "Neural networks for shortest path computation and routing in computer networks", *IEEE transactions on neural networks*, Vol. 4, No.6, November 1993, pp. 941-953.
79. Nabil A. Kartam and Raymond E. Levitt, "Intelligent planning of construction projects", *Journal of construction engineering and management, ASCE*, Vol. 116, No. 2, April 1990, pp. 155-176
80. Nandi. S. N. and B. K. Dutta, "Trends in project cost control systems", *OMEGA*, Vol. 16, No.6, 1988, pp. 513-520.
81. Narsingh Deo, "System simulation with digital computer", *Prentice Hall of India Pvt. Ltd*, New Delhi -1, 1991.
82. Nudtapon Nudtasomboon and Sabah U. Randhawa, "Resource-constrained project scheduling with renewable and non-renewable resources and time-resource tradeoffs", *Computers and industrial engineering*, Vol. 32, No. 1, 1997, pp. 227-242.
83. Oya Icmeli and S. Selcuk Erengue, "A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows", *Management Science*, Vol. 42, No. 10, October 1996, pp. 1395-1408.

84. Osama Moselhi and Khaled El-Rayes, "Scheduling of repetitive projects with cost optimization", *Journal of construction engineering and management, ASCE*, Vol. 119, No. 4, December 1993, pp. 681-697.
85. Osman Coskunoglu, "Optimal probabilistic compression of PERT networks", *Journal of construction engineering and management, ASCE*, Vol. 110, No. 4, December 1984, pp. 437-446.
86. Pasit Lorterapong and Osama Moselhi, "Project network analysis using fuzzy sets theory", *Journal of construction engineering and management, ASCE*, Vol. 122, No.4, December 1996, pp. 308-318.
87. Patterson. J. H., "A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem", *Management science*, No. 30, Vol. 7, July 1984, pp. 854-867.
88. Paul B. Watta and Mohammed H. Hassoun, "A coupled gradient network approach for static and temporal mixed-integer optimization", *IEEE transactions on neural networks*, Vol. 7, No. 3, May 1996, pp. 578-593.
89. Pierre L' Ecuyer, Nataly Giroux and Peter W. Glynn, "Stochastic optimization by simulation: numerical experiments with the M/M/1 queue in steady-state", *Management Science*, Vol. 40, No. 10, October 1994, pp. 1245-1261.

90. Pinto. M, Pinto J. K. and Prescott. J. E., "Antecedents and consequences of project team cross functional cooperation", *Management science*, Vol. 39, No. 10, Oct. 1993, pp. 1281-1297.
91. Poiaga. L., "Operations research in project management and cost engineering: an outlook for new operational developments", *European Journal of Operational Research*, Vol. 41, 1989, pp. 1-14.
92. Punmia. B. C. and K. K. Khandelwal, "Project planning and control with PERT & CPM", *Laxmi Publications*, New Delhi-2, 1988.
93. Ragsdale. C., "The current state of network simulation in project management theory and practice", *OMEGA*, Vol. 17, No. 1, 1989, pp. 21-25.
94. Rainer Kolisch, Arno Sprecher and Andreas Drexl, "Characterization and generation of a general class of resource-constrained project scheduling problems", *Management Science*, Vol. 41, No. 10, October 1995, pp. 1693-1703.
95. Ransing. R. S. and Lewis. R. W., "A semantically constrained neural network for manufacturing diagnosis", *International journal of production research*, Vol. 35, No. 9, 1997, pp. 2639-2660.

96. Rehab M. Reda, "RPM: Repetitive Project Modeling", *Journal of Construction Engineering and Management, ASCE*, Vol. 116, No.2, June 1990, pp. 316-330.
97. Richard F. Deckro and John E. Herbert, "A multivariate objective programming framework for tradeoffs in project scheduling", *Engineering costs and production economics*, Vol. 18, 1990, pp. 255-264.
98. Richard I. Levin et al., "Quantitative approaches to Management", *Mc Graw Hill Inc*, 1992.
99. Robert A. Russell, "A comparison of heuristics for scheduling projects with cash flows and resource restrictions", *Management Science*, Vol. 12, No. 10, October 1986, pp. 1291-1300.
100. Robert C. Leachman, "Multiple resource leveling in construction systems through variation of activity intensities", *Naval research logistics quarterly*, Vol. 30, 1983, pp. 187-198.
101. Robert. G. Murdick and Joel. E. Ross, "Information systems for modern management", *Prentice Hall India*, New Delhi -1, 1992.

102. Robert I. Carr, "Cost, schedule and time variances and integration", *Journal of construction engineering and management, ASCE*, Vol. 119, No. 2, June 1993, pp. 245-265.
103. Ron R. Wakefield and Glenn A. Sears, "Petri nets for simulation and modeling of construction systems", *Journal of construction engineering and management, ASCE*, Vol. 123, No. 2, June 1997, pp. 105-111.
104. Rosenblatt. M. J. and Roll. Y, "A future value approach to determining project duration", *IIE transactions*, Vol. 17, No. 2, June 1985, pp. 164-167.
105. Sagar V. Kamarthi, Victor E. Sanvido and Soundar R. T. Kumara, "Neuroform – neural network system for vertical formwork selection", *Journal of computing in civil engineering, ASCE*, Vol. 6, No. 2, April 1992, pp. 178-199.
106. Salah E. Elmaghraby and Jerzy Kamburowski, "The analysis of activity networks under generalized precedence relations (GPRs)", *Management Science*, Vol. 18, No. 9, September 1992, pp. 1245-1263.
107. Salem. A. M. and S.E. Elmaghraby, "Optimal linear approximation in project compression", *IIE Transactions*, December 1984.

108. Sculli. D., "A historical note on PERT times", *OMEGA*, Vol.17, No.2, 1989, pp. 195-200.
109. Shih. H. M and C.K.H. Leung, "Management Petri net – a modeling tool for management systems", *International journal of production research*, Vol. 35, No. 6, 1997, pp. 1665-1680.
110. Shih-Lin Hung and Jain. J. C, "MS-CMAC neural network learning model in structural engineering", *Journal of computing in civil engineering*, January 1999, pp. 1-11.
111. Shouhong Wang, "Neural networks in generalizing expert knowledge", *Computers and industrial engineering*, Vol. 32, No. 1, 1997, pp. 67-76.
112. Simaan M. Abou Rizk and Daniel W. Halpin, "Probabilistic simulation studies for repetitive construction process", *Journal of Construction Engineering and Management, ASCE*, Vol. 116, No.4, December 1990, pp. 575-594.
113. Suleyman Tufekci, "A flow-preserving algorithm for the time-cost trade-off problem", *IIE transactions*, Vol. 14, No. 2, June 1982, pp. 109-113.

114. Sung. C. S. and Lim. S. K., "A scheduling procedure for a general class of resource constrained projects", *Computers and industrial engineering*, Vol. 32, No. 1, 1997, pp. 9-17.
115. Syslo. M. M., "On the computational complexity of the minimum dummy activities problem in a PERT network", *Networks*, Vol. 14, No. 1, spring 1984, pp. 37-45.
116. Talbot. F. B. "Resource constrained project scheduling with time-resource trade offs; the non preemptive case", *Management science*, Vol. 28, No. 10, Oct. 1982, pp. 1197-1210.
117. Timothy Ch. U. Kalu, "New approach to construction management", *Journal of construction engineering and management*, ASCE, Vol. 116, No. 3, September 1990, pp. 494-513.
118. Thomas R. Hoffmann and Gary D. Scudder, "Priority scheduling with cost considerations", *International journal of production research*, Vol. 21, No. 6, 1983, pp. 881-889.
119. Weiss. G, "Stochastic bounds on distributions of optimal value functions with applications to PERT, network flows and reliability", *Operations Research*, No.4, July-August 1986, pp. 595-605.

120. Weng-Tat Chan, David K.H. Chua and Govindan Kannan, "Construction resource scheduling with Genetic algorithms", *Journal of construction engineering and management, ASCE*, Vol. 122, No. 2, June 1996, pp. 125-131.

121. William J. Rasdorf and Osama Y. Abudayyeh, "NIAM conceptual data-base design in construction management", *Journal of construction engineering and management, ASCE*, Vol. 6, No. 1, January 1992, pp. 41-62.

122. Wills R. J., "Critical path analysis and resource constrained project scheduling-theory and practice", *European Journal of Operations Research*, Vol. 21, No. 2, Aug. 1985, pp. 149-155.

123. Yinzhen Li, Kenichi Ida, Mitsuo Gen and Reiko Kobuchi, "Neural network approach for multicriteria solid transportation problem", *Computers and industrial engineering*, Vol. 33, Nos. 3-4, 1997, pp. 465-468.

124. Zhou. D. N, Cherkassky. V, Baldwin. T. R and Olson. D. E, "A neural network approach to job-shop scheduling", *IEEE transactions on neural networks*, Vol. 2, No. 1, January 1991, pp. 175-179.

Publications by the author in related topics

1. Sreemahadevan Pillai. P. R., "Project scheduling techniques - a state of the art appraisal", *Ph.D. course seminar, IIT Bombay*, Industrial Management Interdisciplinary programme, May 1989.
2. Sreemahadevan Pillai. P.R., "Managing a project ", *Proceedings of the national seminar on 'Modernization in the Irrigation Sector of Kerala'*, Malampuzha, Palakkad, Kerala, organised by Centre for Water Resources Development and Management, Kunnamangalam, Kozhikode, Kerala, October 10-11, 1990.
3. Sreemahadevan Pillai. P.R., "Who makes a Project Manager", *Proceedings of the national seminar on review and updating of theory and practice in civil engineering*, N.S.S. College of Engg., Palakkad, 29-30 Nov. 1990.
4. Sreemahadevan Pillai. P.R., "History of Artificial Neural Networks", *Proceedings of the AICTE-ISTE summer school on Artificial Neural Networks and Applications*, N.S.S. College of Engg., Palakkad, 2-13 May 1995..
5. Sreemahadevan Pillai. P. R. and Palaniraj. S, "Possible corrections to time estimates in PERT", *National seminar on Project Management*, The Institution of Engineers (India), Palghat Local Centre, December 1995.

6. Sreemahadevan Pillai. P.R., "Material and Labour costing", *Proceedings of the workshop on cost control in construction Projects*, Regional Engineering College, Calicut, 17-19 January, 1996.
7. Sreemahadevan Pillai. P. R., "A concise text on principles of management", *Spade Publishers and Distributors*, Calicut-1, 1997.
8. Sreemahadevan Pillai. P.R., "Project Management - Theory and Practice", *Proceedings of the two-day national seminar on recent developments in civil engineering-theory and practice*, The Institution of Engineers (India), Palakkad, 16-17, August 1997.
9. Sreemahadevan Pillai. P.R., "Managing the Y2k Problem", *Proceedings of the All India Seminar on Information Technology*, The Institution of Engineers (India), Palakkad, 14 Nov., 1999.
10. Jawahar. K. and Sreemahadevan Pillai. P.R, (guide) "Social auditing and alternative construction management for positive community development" *MBA thesis*, *Indira Gandhi National Open University*, December, 2000.
11. Sreemahadevan Pillai. P. R. and Palaniraj.S, "A comparison of probabilistic networking methods used in construction project scheduling", *Under review at NICMAR journal of construction management*.

12. Sreemahadevan Pillai. P. R. and Palaniraj.S, "Artificial neural networks in project scheduling techniques", *Under review at International journal of management and systems.*
13. Sreemahadevan Pillai. P. R. and Palaniraj.S, "Software packages in project scheduling", *Under review at International journal of management and systems.*
14. Sreemahadevan Pillai. P. R. and Palaniraj.S, "Time-step chart for project scheduling" *Under review at Management Science (international journal)*
15. Sreemahadevan Pillai. P. R. and Palaniraj. S, "Project scheduling techniques – a state of the art appraisal", *Proposed to be published by Oxford and IBH publishing company in 2002*