In presenting the dissertation as a partial fulfillment of the requirements for an advanced degree from the Georgia Institute of Technology, I agree that the Library of the Institute shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to copy from, or to publish from, this dissertation may be granted by the professor under whose direction it was written, or, in his absence, by the Dean of the Graduate Division when such copying or publication is solely for scholarly purposes and does not involve potential financial gain. It is understood that any copying from, or publication of, this dissertation which involves potential financial gain will not be allowed without written permission.

7/25/68

A DYNAMIC PROGRAMMING APPROACH TO PLANNING WITH DECISION NETWORKS

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

by

Daniel Sipper

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Industrial and Systems Engineering

Georgia Institute of Technology

September, 1970

A DYNAMIC PROGRAMMING APPROACH TO
PLANNING WITH DECISION NETWORKS

Approved:

Chairman

Date approved by Chairman: OCT. 15, 1970

ACKNOWLEDGMENTS

It is a pleasure to express my sincere appreciation and gratitude to my thesis advisor, Dr. J. Gordon Davis. His advice, suggestions, guidance and assistance were instrumental in the completion of this research. I especially appreciate the friendly and constructive spirit which prevailed during the many hours Dr. Davis spent reviewing this thesis.

Dr. Robert N. Lehrer, Director of the School of Industrial and Systems Engineering, was my advisor during the doctoral study, and also served on the thesis reading committee. His considerable assistance and support made the completion of my program a reality. I shall always remember his moral support during the waning hours of my graduate study.

Dr. Douglas C. Montgomery, member of the thesis advisory committee, carefully read the manuscript and made many helpful suggestions, for which I am very grateful.

Dr. Gerald J. Thuesen and Dr. Kong Chu were members of my thesis advisory committee and were of great assistance in reading the thesis.

Many other members of the faculty and staff of Georgia Tech and fellow graduate students contributed and inspired, directly or indirectly, my graduate work. They are too numerous to mention individually, but to all of them I would like to express my thanks.

I sincerely and gratefully acknowledge the encouragement received from my family. In particular, I am indebted to my wife Shosh for her

patience and sacrifice, and for keeping our family thriving for three difficult years. To her I express my special thanks.

This work is dedicated to the memory of my late father, who taught me how to strive, to seek, to find and never yield.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS. ii

LIST OF TABLES viii

LIST OF ILLUSTRATIONS. ix

SUMMARY. xiii

Chapter

I. INTRODUCTION. 1

- Purpose
- Problem Formulation
  - Decision Network
    - The Case of Certainty
    - Computational Magnitude
    - The Case of Risk with Stochastic Decisions
    - The Case of Risk with Stochastic Outcomes
- Importance of the Problem
- Scope and Limitations
- Organization
- Objectives

II. LITERATURE SURVEY 13

- Introduction
- CPM and PERT
  - CPM (Critical Path Method)
  - PERT (Program Evaluation and Review Technique)
- Network Algebra
- Digital Simulation
- Decision Networks
  - Generalized Networks
  - Decision Trees
  - Deterministic Networks
- Concluding Remarks

III. MATHEMATICAL BACKGROUND 36

- Introduction

Chapter Page

III. MATHEMATICAL BACKGROUND (Continued)

- Definitions
  - Project
  - The Project Network
  - Activities on Arrows (A-O-A)
  - Activities on Nodes (A-O-N)
  - Types of Nodes
    - Activity Node
    - Decision Vertex
    - Decision Node
    - Stochastic Decision Vertex
    - Stochastic Vertex
    - Outcome Node
  - Types of Networks
    - Standard Network
    - Decision Network
    - Stochastic Decisions Network
    - Stochastic Outcomes Network
- Mathematical Basis
  - Reduction of a Decision Network to a Standard Network
  - Precedence Relations
  - Critical Path Computation

IV. THE CASE OF CERTAINTY 51

- Introduction
- Minimum Time Problem
  - DP Model--Simple Formulation
  - DP Model--Extended Formulation
    - Decision Dependent Subnetwork (DDS)
    - Algorithm for Network Decomposition: First Level Labeling
    - DP Model of a DDS
    - Nonserial DP Models
    - Minimum Project Time
    - DP Solution
  - Computational Refinements
    - Dominating Path
    - Common Decision Vertices
  - The Minimum Time Network
- Minimum Cost Problem
  - DP Model
    - Network Decomposition
    - Reduced Cost Network (RCN)
    - Evaluating Cost Returns
  - DDS Minimum Cost

Chapter Page

IV. THE CASE OF CERTAINTY (Continued)

Minimum Cost Problem (Continued)
  Project Minimum Cost
    Computational Refinements
Time Cost Trade-Off
  A Methodology for Finding the Efficient Set
    Efficient Set Algorithm
    Optimality of the Algorithm
    Efficient Set Tableau
  The Efficient Set for a Decision Network
Constraints and Sensitivity

V. THE CASE OF RISK WITH STOCHASTIC DECISIONS. 115

Introduction
Project Time and Cost Extremes
Expected Project Time and Cost
  Expected Project Time
    Optimistic Expected Time
    Project Control Policies
    The Markov Process for the Optimistic Expected Time
    Pessimistic Expected Time
    Example
  Expected Project Cost
    Pessimistic Expected Cost
    Optimistic Expected Cost
    Example
Risk Evaluation
Most Probable Project Network
Concluding Remarks

VI. THE CASE OF RISK WITH STOCHASTIC OUTCOMES 145

Introduction
Minimum Expected Project Time
  Network Decomposition
  Minimum Expected Project Time
    Optimistic and Pessimistic Minimum Expected Time
    Expected Cost of the Minimum Expected Time
    Strategy Tree
Minimum Expected Project Cost
  Network Decomposition
  Evaluation of Minimum Expected Project Cost
    Optimistic and Pessimistic Minimum Expected Cost
    Expected Time of the Minimum Expected Project Cost

Chapter Page

VI. THE CASE OF RISK WITH STOCHASTIC OUTCOMES (Continued)

- Region of Possible Outcomes
  - Region of Possible Outcomes for the Whole Project
  - Region of Possible Outcomes for an Opening Policy
- Range of Outcomes for the Optimal Strategies
- Most Probable Outcome
- Evaluation of Risk in an Optimal Strategy
- Simulation
  - Evaluation of Risk in an Optimal Strategy
    - Simulation Results
  - Probability Associated with an Opening Policy
    - The Simulation Process
    - Simulation Results
- Concluding Remarks

VII. CONCLUSIONS AND RECOMMENDATIONS 174

- Conclusions
- Recommendations

APPENDIX

A. NONSERIAL DP MODELS 180

B. THE CASE OF CERTAINTY: AN EXAMPLE. 187

C. THE CASE OF RISK WITH STOCHASTIC DECISIONS: AN EXAMPLE . . 206

D. THE CASE OF RISK WITH STOCHASTIC OUTCOMES: AN EXAMPLE. . . 216

E. EFFICIENT SET ALGORITHM: AN EXAMPLE. 243

F. NOTATIONS AND ABBREVIATIONS 248

BIBLIOGRAPHY 253

VITA 258

LIST OF TABLES

Table Page

1. Path Enumeration 167

2. DDS Efficient Set Table 203

3. $T_i(X_i,Y_i,\Delta^i)$; $Q_i(X_i,Y_i,\Delta^i)$; $\bar{Q}_i(X_i,Y_i)$ for DDS_1. 219

4. $T_i(X_i,Y_i,\Delta^i)$; $Q_i(X_i,Y_i,\Delta^i)$; $\bar{Q}_i(X_i,Y_i)$ for DDS_2. 221

5. $T_i(X_i,Y_i,\Delta^i)$; $Q_i(X_i,Y_i,\Delta^i)$; $\bar{Q}_i(X_i,Y_i)$ for DDS_3. 222

6. $C_i(X_i,Y_i,\Delta^i)$; $R_i(X_i,Y_i,\Delta^i)$; $\bar{R}_i(X_i,Y_i)$ for DDS_1. 226

7. $C_i(X_i,Y_i,\Delta^i)$; $R_i(X_i,Y_i,\Delta^i)$; $\bar{R}_i(X_i,Y_i)$ for DDS_2. 226

8. $C_i(X_i,Y_i,\Delta^i)$; $R_i(X_i,Y_i,\Delta^i)$; $\bar{R}_i(X_i,Y_i)$ for DDS_3. 227

9. Region of Possible Outcomes for H_i. 232

10. Path Enumeration for the Minimum Expected Time Strategy . . 235

11. Path Enumeration for the Minimum Expected Cost Strategy . . 236

12. Simulation Results: Minimum Cost Criterion. 237

13. Simulation Results: Minimum Time Criterion. 238

14. Simulation Results: $\bar{\bar{U}}^*$ 239

15. Simulation Results: $\bar{\bar{V}}^*$ 240

16. DP Model Matrices 244

17. Efficient Set Tableau 245

18. Admissible Point Test 246

LIST OF ILLUSTRATIONS

Figure Page

1. A Project Network 2
2. A Decision Network. 4
3. A Stochastic Decisions Network. 6
4. A Stochastic Outcomes Network 8
5. Research Done in Decision Networks. 21
6. Eisner's Decision Boxes 22
7. Logical Relations 23
8. Network Algebra 24
9. Node Characteristics and Symbols--GERT. 26
10. w Function. 27
11. Decision Tree 30
12. Types of Nodes--Schematic Illustration. 38
13. Decision Network Reduction. 43
14. A Simplified Decision Network 53
15. A DP Model for the Simplified Network 53
16. Serial DP Model 53
17. A Decision Network: An Example 58
18. Decision Network. 59
19. Decision Network--Schematic Description 59
20. Modified Decision Network 60
21. Decision Network--Schematic Description 61

Figure Page

22. Decision Network with a Common Decision Vertex. 62

23. Decision Network--Schematic Description 62

24. CDDS--An Example. 64

25. DDS_3. 69

26. Decomposed Network. 70

27. Type I Divergence 73

28. DP Model--Type I Divergence 73

29. Cost Decision Network 80

30. Cost Example. 85

31. A Serial DP Model 89

32. Efficient Set Algorithm--Time Version: Flow Chart. 99

33. The Efficient Set 100

34. DP Matrices 101

35. Efficient Set Tableau: Time Version. 107

36. DDS Efficient Set Table 110

37. Constraints 114

38. Region of Possible Outcomes 117

39. A Stochastic Decisions Network. 120

40. Markov Process of a DDS 123

41. Stochastic Decisions Network with
a Common Decision Vertex. 126

42. The Equivalent Markov Processes 128

43. Stochastic Decisions Network: Schematic Illustration . . . 129

44. Elements of a Stochastic Outcomes Network 147

Figure Page

45. Stochastic Outcomes Network with a Common Decision Vertex. 148

46. PRN 154

47. Strategy Tree 155

48. Network Transformation. 163

49. PRN: An Example. 165

50. Probability Zone for Policy H_i. 170

51. A Decision Making Routine 173

52. Type I Divergence 180

53. Type II Divergence. 182

54. Type II Divergence: Equivalent DP Model. 182

55. Type I Feedforward. 184

56. Equivalent DP Model 184

57. Transformed DP Model. 184

58. Transformed DP Model--Cost. 185

59. Type II Feedforward 186

60. Equivalent DP Model 186

61. Second Level Labeling Algorithm--Time 188

62. Second Level Labeling Algorithm--Cost 193

63. Minimum Cost Standard Network 196

64. Minimum Time Standard Network 204

65. Time-Cost Trade-Off 205

66. Region of Possible Outcomes 208

67. DDS_1. 208

68. Most Probable Project Network 215

Figure Page

69. Stochastic Outcomes Network 217

70. DDS_1. 218

71. PRN--Time 223

72. PRN--Cost 229

73. Region of Possible Outcomes 232

74. Region of Possible Outcomes and Probability Zones for Opening Policy H_i: Minimum Time Criterion. 241

75. Region of Possible Outcomes and Probability Zones for Opening Policy H_i: Minimum Cost Criterion. 242

76. The Efficient Set 247

SUMMARY

The general purpose of this research is to develop a methodology for evaluating projects during the planning phase, when there are alternative methods of achieving the stated goals of a given project. The emphasis in this study is on the managerial decision making process involved in approving or rejecting a project.

A network model is used to represent the project, and since this network includes all possible alternatives of performing the project, it is called a decision network.

Three types of decision networks are considered in this research. The first type is a case where all alternatives are known with certainty during the planning phase, and the other two types describe cases where uncertainty is associated with each alternative, described by a probability distribution. Thus, this study handles both deterministic and stochastic decision networks.

No assumptions are made relative to the decision maker. Instead, inputs to the decision making process are developed. The two parameters considered are time and cost.

For the deterministic network, the problem is that of selecting a particular set of alternatives to be performed, each represented by its time and cost values. The input to the decision making process developed for this case takes the shape of a time-cost trade-off relationship, where the interesting sets of alternatives are those that yield a lower cost value for a higher time value.

For the stochastic networks, a variety of inputs to the decision making process is developed, utilizing various criteria of choice. These inputs supplement each other and enable the decision maker to make his evaluation on the basis of a broad information base.

The procedure introduced in this research is based upon discrete dynamic programming concepts for the deterministic network, and stochastic discrete dynamic programming for the stochastic networks. In addition, Monte Carlo simulation is applied to the stochastic networks.

A series of numerical examples is used to supplement the methodology developed in this study, demonstrating that the computation procedure is not complex, and is practical.

Recommendations are made that a project represented by a mixed deterministic and stochastic network be investigated, and that the multi-project case be explored. Also, further investigation of some of the techniques developed in this research is suggested, with the main purpose of applying these techniques for other types of problems.

CHAPTER I

INTRODUCTION

Purpose

The purpose of this research is to develop methodology for evaluating projects during the planning phase, when there are alternative methods of achieving the stated goals of a given project. Such methodology would permit an objective and systematic approach to the problem of project assessment.

Problem Formulation

The area of project management experienced a major change in 1957 with the introduction of new techniques, commonly known as CPM (critical path method) and PERT (program evaluation and review technique). These techniques are characterized by their use of a project model consisting of an acyclic network representing dependencies among the activities to be performed in order to achieve a given goal. The network includes additional information, typically the duration and cost of each activity. Thus, a project network may look like that of Fig. 1, where the nodes represent activities, and arrows indicate precedence relationships among the activities. This type of network is referred to as a "standard network."

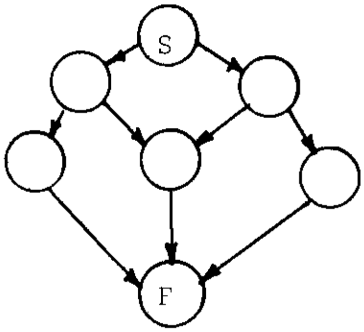


Figure 1. A Project Network

In network based project management there exists distinct separation between the planning phase, the scheduling phase and the control phase. The planning phase is usually identified with the construction of the project network, during which time specific decisions are made on the method of performing the various activities. Thus, if there is a number of competing methods of performing some of the activities, an elimination process takes place *before* the project network is constructed. The scheduling phase is concerned with establishing commencement and completion times for each activity. The control phase provides the basic for adjusting the schedule throughout the life of the project.

Sometimes during the planning-scheduling phases, management is interested in finding out the time and cost of the project. These are the two most crucial elements of the project evaluation process, the outcome of which determines whether the project is going to be undertaken.

Evaluation of the project time and cost is relatively simple for the CPM type network, where the elimination of alternatives is completed before the project network is constructed. However, this elimination process does not guarantee that the most desirable network representing the project will be selected. When a few alternatives exist for performing the project, it is not hard to visualize that management would like to know the available options. For the purpose of this research these options will be in terms of project time and cost for the various alternatives. The a-priori elimination process results in presenting to management just one of these options, which may or may not be the most desirable. Thus, the decision making process of management is basically reduced to accepting or rejecting this one option out of the many, most of which are not even known.

A more desirable approach would be to perform a posterior elimination of alternatives, after all options are known. This can be achieved by introducing explicitly all alternatives into the project network, resulting in a decision network.

Decision Network

Suppose that, during the planning phase, there are different alternatives for performing some of the activities, with each alternative having a different cost, a different time duration, and different dependencies. Also suppose that *no* elimination process takes place *before* the project network is constructed. The result is a different type of project model--one which *includes* all the possible alternatives. Fig. 2 is an example of such a network. The triangular nodes are referred to as "decision vertices" and imply that a decision must be made to select *at most* one of the possible alternatives, called decision nodes, emanating from the decision vertex. This type of project network is called "Decision Network."

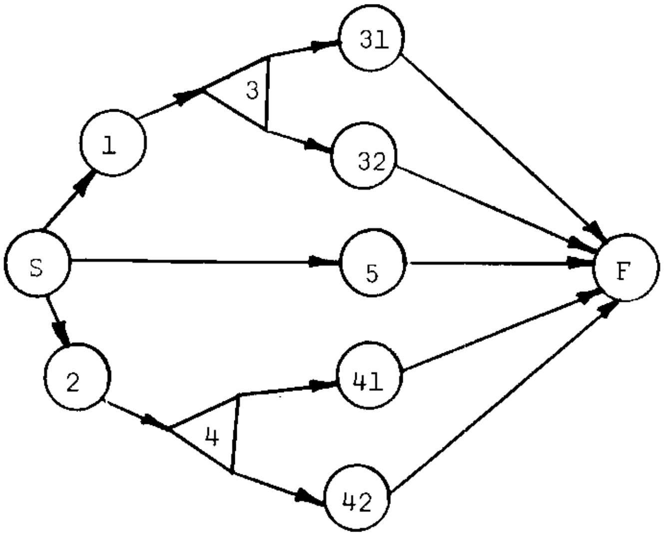


Figure 2. A Decision Network

The Case of Certainty. Consider the decision network of a project, as presented in Fig. 2. In this example, the scheduling phase cannot start until at most one decision node is selected at each decision vertex. This selection is done during the planning phase. By eliminating all decision nodes that were not chosen, the decision network can be reduced to a standard network.

The problem then is that of selecting a particular set of alternatives to be performed. Since each activity is quantitatively described by its duration and cost, then the set of alternatives selected should be that set yielding the most desirable time--cost combination for the whole project. This does not necessarily mean that the combination yielding the minimum project time or minimum project cost is the most desirable one. However, the only combinations of interest are those that for a higher project time value have a lower cost value. Thus, a time-cost trade-off is required to enable management to make the selection that meets its desires or constraints. It is definitely a planning problem.

This problem is referred to as "The Case of Certainty" because all possible alternatives are known during the planning phase, it is possible to eliminate all undesired alternatives, and the outcome of selecting each alternative is known with certainty. Situations like this arise in conventional projects or in development projects.

The mathematical formulation of this problem is as follows:

Let:

T_i - the ith possible project time value

T_i^m - the ith possible project time value obtained by selecting the mth subset of decision nodes

$C_{T_i^m}$ - project cost associated with T_i^m

$C_i = \underset{m}{\text{Min}} \left(C_{T_i^m}\right).$

Then, the time-cost trade-off is given by all points satisfying the following inequalities.

if $T_i > T_{i-1}$ and T_i's are arranged in ascending order of T_i values.

then $C_i < C_e \quad \forall \quad e = 1 \ldots i-1$

and W_i is the policy associated with this point, i.e. the set of decision nodes that, if selected, will yield the above (T,C).

Computational Magnitude. If there are 15 decision vertices each having two decision nodes, there are 2^{15} = 32,768 decision patterns. If there are 3 decision nodes per decision vertex, there are 3^{15} = 9,034,497 decision patterns.

Certain types of projects, especially R & D projects, are characterized by the uncertainty of the outcomes of performing some of the activities. The problem formulation given above would not fit these cases. Therefore a different approach has to be used.

The Case of Risk With Stochastic Decisions. This problem formulation considers the case where all possible alternatives are known during the planning phase; however, preliminary selection is impossible because there is a probability associated with each alternative. A situation like this is represented by the decision network of Fig. 3.

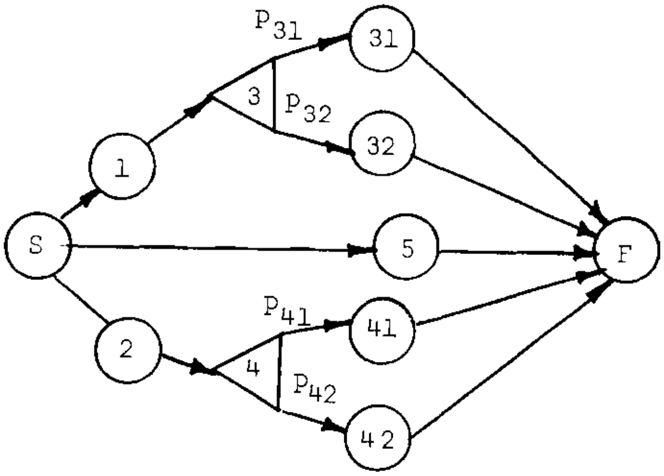


Figure 3. A Stochastic Decisions Network

A network of this type is referred to as a "Stochastic Decisions

Network." The probabilities associated with, say, decision vertex 3 represent the fact that during the planning phase there is not sufficient information to decide whether "31" or "32" should be performed, although as the project unfolds and vertex 3 is reached the decision maker is able to eliminate either one and proceed with the other one. Thus, the probabilities in this case represent the uncertainty during the planning phase. Note that it is not certain at the outset which of these alternatives will be chosen. This is representative for example of a development project where, say, node 1 can represent a certain state of knowledge that has to be acquired before "31" or "32" can be selected.

In contrast to the case of certainty, where selection among alternatives and network reduction were possible during the planning phase, no such approach is possible here.

Due to the nature of this problem, time-cost trade-off has no meaning here. However, the decision maker is still in need of some information in order to decide whether or not to proceed with the project. Thus, the problem here is to develop decision making tools for evaluating projects with known alternatives and uncertain future.

The Case of Risk With Stochastic Outcomes. This problem formulation considers a different case than the previous one. Again, all possible alternatives are known during the planning phase, however, each alternative, if selected, is followed by a finite number of stochastic outcomes. An example of this type of project is represented in Fig. 4. The decision network associated with this case is called "Stochastic Outcomes Network."

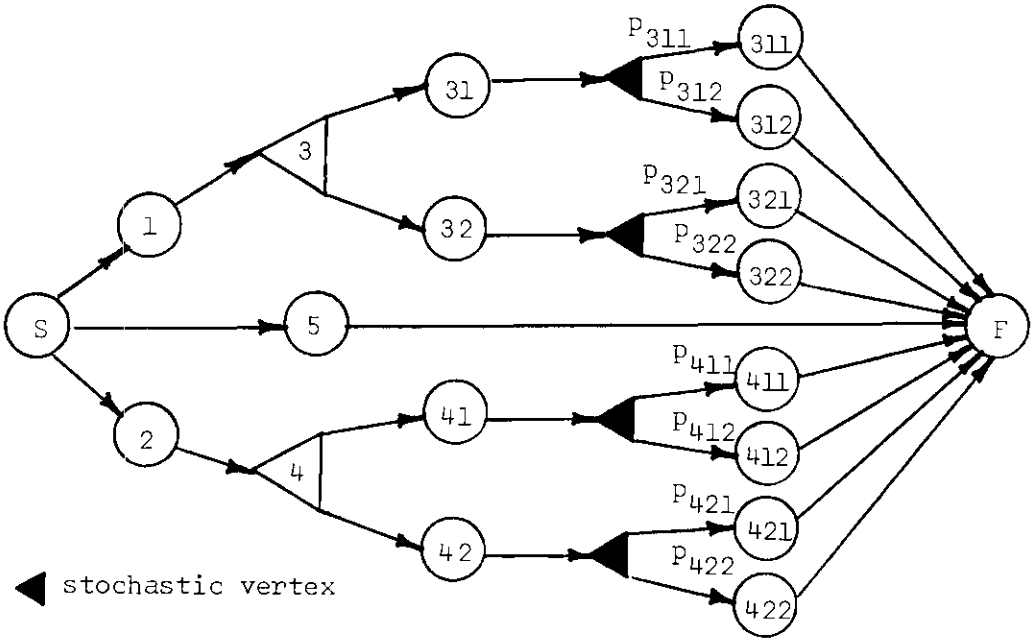


Figure 4. A Stochastic Outcomes Network

The decision maker, either during the planning phase or during the execution of the project, is able to control the alternative selected, but he is unable to control the stochastic outcome associated with the selection of a certain alternative. The stochastic outcomes associated with each alternative are represented by a probability distribution. Each outcome has associated with it a duration and cost values, in the same manner as a regular node. Thus, this case differs from the previous one in the amount of control that can be exerted during the planning and execution of the project. This situation is common to R & D projects or conventional projects, where certain events are uncontrollable (for example, in a construction project the weather would be uncontrollable).

Due to the nature of this case, only partial elimination of alternatives is possible during the planning phase, once a criterion for such elimination is established. Again, a decision has to be made during the planning phase as to whether to proceed with the project or not. The time-cost trade-off approach is not applicable here because of the uncertainty of the future. Therefore, the problem here is to develop inputs to the decision making process to enable the evaluation of a stochastic outcomes project during the planning phase.

Importance of the Problem

Network based project management techniques are widely used today in all phases of industry and business. The vast amount of literature available, and the fact that there are over 50 computer programs available in this area are an indication of the diversified use of these techniques.

One of the shortcomings of the existing techniques is in handling cases where a selection among alternatives has to be made, or whenever a decision has to be made in the face of an uncertain future.

Whenever a new venture is to be undertaken, management is seeking information relative to at least two factors: time and money. The proper combination of the two which is acceptable to management may vary from case to case, but, unless there is such an acceptable combination, the venture will never take place.

Practically thousands of decisions of this type are made every day. When the venture at hand is a project with a few alternatives available for some of its elements, and when the future is either certain or uncertain, management does not have adequate information in order to make the proper decision. The few available techniques, which are applicable to only some of the previously mentioned cases, concentrate on only one criterion of choice for decision making. However, not less important is presenting to management information based upon various decision making criteria and let management decide what the optimal option is.

Scope and Limitations

The research reported herein considers problems associated with the planning phase of project management, emphasizing the managerial decision making process involved in approving or rejecting a project. No assumptions are made relative to the decision maker. Instead, inputs to the decision making process are developed.

The present study is confined to the planning phase only and does not consider the scheduling and control phases..

Although this research is restricted to the analysis of project networks, it is recognized that some of the techniques to be developed might have some other applications, as suggested in Chapter VII.

The research effort presented herein is restricted to the single project case, and no resource constraints or nonsimultaneity constraints are considered. Time and cost are the only parameters examined here, and no other factors affecting the project evaluation process are introduced. The time value of money is not considered explicitly.

It is assumed that activity duration and cost are known with certainty and are single valued. Also, it is assumed that all alternative ways of performing a certain activity and all possible stochastic outcomes following a certain activity (where applicable), as well as the probability distributions for the stochastic cases, are known during the planning phase.

The research is of a general nature and applies to all projects fitting the general model presented herein.

Organization

Chapter II gives a general review of project management and related literature and a detailed review of the literature relevant to the present study.

Chapter III discusses some mathematical concepts of project management and decision networks.

Chapter IV treats the case of certainty. A dynamic programming model of a decision network is introduced, and a solution procedure for finding time-cost trade-off is developed.

Chapter V discusses the case of risk with stochastic decisions. Various analytical techniques are discussed. Among others suggested, the problem is formulated as a Markov Process with rewards.

Chapter VI analyzes the case of risk with stochastic outcomes. Again, various approaches are discussed, where dynamic programming and stochastic dynamic programming are the dominant techniques used. This chapter also examines Monte Carlo simulation as a possible approach to this type of problem.

Chapter VII concludes the research with summary of the results and recommendations for further research. The appendices include examples demonstrating the solution procedures developed.

Objectives

The primary objective of this research is to develop methods by which problems attendant to planning with decision networks can be resolved without examining every possible outcome. The secondary objective is the extension and application of Operations Research techniques to a class of real world problems.

CHAPTER II

LITERATURE SURVEY

Introduction

The literature associated with network based project management is very extensive. The purpose of the survey presented herein is to give an overview of representative literature in this area, followed by a more detailed discussion of the literature closely related to the topic of this research. Some additional items which have a bearing on the problem area under consideration are included too. Thus, this survey is divided into four sections as follows:

CPM and PERT
Network Algebra
Digital Simulation
Decision Networks

For a more extensive literature survey, the interested reader is referred to Krishnamoorthy (39).

CPM and PERT

Initially, most of the research attention that has been directed towards network based project management dealt with the "single" project type--i.e. only one large complex work program is involved.

It was for this type of project that the now widely used PERT (Program Evaluation & Review Technique) and CPM (Critical Path Method) procedures were developed. This was the origin of the network approach to project management, and it can be attributed to two separate projects: one undertaken by industry, the other by the U. S. Government.

Both groups advocated the use of a network depicting explicitly the relationships among various activities. This was a significant change from the then existing technique of using bar-charts.

PERT (45) and CPM (33,34,35) were developed independently and at about the same time in 1958. PERT was originally designed for the Navy's Polaris research and development program, whereas CPM was designed for a construction project at DuPont.

CPM (Critical Path Method)

This method was initiated by Kelly and Walker (33,34,35). It was completely different from the previously known project planning techniques in the sense that the functions of planning and scheduling were separated. In this method, activity durations are considered to be deterministic for a certain level of resource utilization. This level can be varied by varying the amount of money spent for direct cost factors, and accordingly there is a change in the activity duration. In (33) Kelly and Walker introduced the functional relationship between project cost and time, by defining, for each activity, limits for time and cost called "normal" and "crash." Kelly (35) developed this further to a parametric linear programming formulation to obtain the project cost curve. Fulkerson (22) developed a similar analysis. Both Kelly and Fulkerson assumed that a project's time-cost relationship is a continuous, convex function, and that this function can accurately be represented by a piece-wise linear approximation.

Some original simplifications and modification of Fulkerson's algorithm may be found in an article by Roper (57), who has also borrowed ideas from Kelley. Roper's algorithm produces sub-project cost curves in addition to the project cost curve.

A somewhat similar approach was taken by Alpert and Orkland (1), and refined by Moder and Phillips (49, p.109). Their procedure considers only discrete time-cost points. This method does not require the assumption of a convex cost function. However, the method used will not give all possible minimum-cost project time reduction; consequently, it is not necessarily an optimal procedure.

Meyer and Shaffer (48) used integer linear programming to study project cost functions. However, with present algorithms they state that projects of 50 or more activities cannot be handled.

Some extensions of CPM include the work of Gessford (24), who found that "medium and large construction firms may find it economically and administratively, advantageous to add cost constraints to their existing CPM/Time systems," and the article by Kleinschmidt, Moore and Tamashanas (38), who introduced cash flow into CPM and applied it to "make" or "buy" decisions.

The mathematical basis for CPM was established by Kelley (35) and later extended by Levy, Thompson and Wiest (40).

PERT (Program Evaluation and Review Technique)

PERT was formally defined by Malcolm, Roseboom, Clark and Fazar (45). PERT was originally designed to be time oriented--it paid little explicit attention to factors of cost and resource availability. The basic difference between PERT and CPM is that in CPM activity durations are deterministic, whereas in PERT activity durations are subject to a probability distribution. An integral part of this probability distribution is the system of three time estimates--normal, optimistic and pessimistic. The paper by Malcolm *et al.* (45) assumes a Beta distribution for activity duration. It suggests that the probability of completing a project by a given date can be computed by calculating the critical path using an expected activity duration as deterministic quantity and then invoking the central limit theorem.

PERT assumptions were discussed by Murray (50) and MacCrimmon and Ryavec (41,42). They have performed rigorous analyses of the PERT assumptions and have suggested methods which may lead to better time estimates and probability statements. Clark (8) developed an iterative procedure to get the expected value and variances of a network. He uses different assumptions than the original PERT assumptions, the main difference being assuming that the elements of the network are normal random variables. Moder and Phillips (49, p.229-239) provide an illustrative application of this procedure.

In a different article, Clark (9) makes an attempt to validate the probability statements of PERT. Grubbs (26) has pointed out the subjective nature of the PERT estimation problem and the restrictions on the Beta distribution.

While Macrimmon and Ryavec were working on a comprehensive analysis of PERT assumptions, Van Slyke (59,60) was exploring the use of Monte Carlo methods to yield solutions to the PERT problem. He observed that the Monte Carlo estimate of the mean project length is unbiased. Another outcome of his research is "criticality index" for each activity expressed in terms of the probability that the activity will be on the critical path.

An attempt to remove the biases in PERT assumptions was made by Hartley and Wortham (27). The novel feature of this article is the attempt to synthesize various contributions in this direction and evolve a statistical theory for the derivation of unbiased distribution of the project completion times, with a provision for sensitivity analysis relating to assumptions.

A research into the behavioral aspect of time estimating was performed by King and Wilson (37) and King, Vittebrongel and Hazel (36). The second paper is more or less a continuation of the first one. These two papers fulfill a long felt need to initiate research on the estimating behavior of individuals in relation to PERT assumptions. The conclusion of these two papers is that there is no significant change in the accuracy of estimating the remaining portion of an activity as the portion of activity remaining becomes smaller.

Some extensions of PERT include PERT/cost (15) and PERT/Reliability (46). PERT/cost adds the considerations of resource costs to the schedule produced by PERT/Time; however, it does not provide probability information relative to cost. There is no attempt to use cost data in such a way as to optimize total project costs. PERT/Reliability is an extension of PERT into reliability management.

For the computer aspects of PERT, the article by Phillips (53) gives a wealth of information concerning various CPM and PERT computer programs.

To conclude this discussion of PERT, the following quote from Caruthers and Battersby (3) is of interest: "In spite of the initial success of PERT, its distinguishing feature of a statistical distribution of activity time, is seldom used."

Network Algebra

It is not the purpose here to cite all work done in network algebra, but only a few references having some relationship to the research presented herein.

Charnes and Cooper (4) applied the subdual algorithm of linear programming to critical path scheduling. The project graph is converted to a network by imposing a flow on it. Then, a pair of linear programming problems is synthesized so that pertinent applications of the theory of subdual algorithms can be applied. Another illustration of this approach is given in Moder and Phillips (49, p.135-139).

Application of chance constrained programming methods to examining some statistical properties of PERT networks is reported by Charnes and Thompson (5). The main focus of this paper is on the statistical distributions of the project completion times.

The idea of decomposing a project network into subnetworks is discussed by Parikh and Jewell (52). Their paper considers a "critical path" networks, and presents a method to decompose a project network into subnetworks, schedule the subnetworks and then put the project network back together. First, time-only networks are handled. Then, the method is extended to cost-time networks by a method which is a generalization of Fulkerson's (21, p.151-169) project cost curve method and "out of kilter" algorithm.

Martin (47) presents a method of computing the density function of the passage time from source to sink of an acyclic network. An interesting technique in this paper is an algorithm reducing a series-parallel network to a single arc whose density function is that of the time through the original network.

The problem of finding a mini-maximal path in a disjunctive PERT network was solved by Balas (2). The procedure is iterative and consists of a two-stage solution: In stage 1 a 0-1 integer programming problem is solved to select a subset of the disjunctive pairs of arcs between all pairs of nodes. In stage 2 the critical path is found, and a simple test shows whether it is minimaximal or not. If it is not, each critical path of stage 2 defines a new constraint for stage 1.

Digital Simulation

The application of digital simulation to project management has been discussed previously in connection with PERT assumptions (ref. 59, 60). A few more references that are of interest are cited here.

Trilling (58) describes a job shop simulation of orders that are networks. This work describes a coding procedure based on binary numbers while defining the networks represented by the routings or line-ups of shop orders. Several decision rules are tested.

The application of GPSS to Project management is reported by Hicks and Jain (28). They considered a number of examples of complex precedence relationships employing GPSS/360 program. Their conclusion is that GPSS/360 can be employed to develop project management information not readily attainable employing standard project management programs.

An interesting application of Monte Carlo simulation to investment risk analysis is reported in the work of Hess and Quigley (30). They worked out an example where the distribution of a certain profitability criterion was obtained using Monte Carlo, for the case of a few variables with given distributions, where analytical techniques fail because of complexity. Clark (10) made a similar analysis for the case of two investments, where the cash flows have a probability distribution. Using Monte Carlo simulation he obtained the probability distribution of the rate of return of each investment.

Hespos and Strassman (29) applied simulation to the case of stochastic decision trees. This is discussed in the following section.

Decision Networks

A somewhat more detailed discussion is going to be presented here, as this area is the most relevant to the present study.

Fig. 5 summarizes the flow of ideas in this area. Thus, in 1962 Eisner (17) introduced decision boxes into PERT network, while Magee (43,44) published his decision tree analysis in 1964. Elmaghraby (19) added feedback loops to the network, while Hespos (29) followed Magee with stochastic decision trees. Pritsker *et al.* (16,54,55,56, 61) developed GERT, and Elmaghraby (18) further developed his previous work. Chillcot and Thursfeld (6,7) illustrated applications of Eisner's

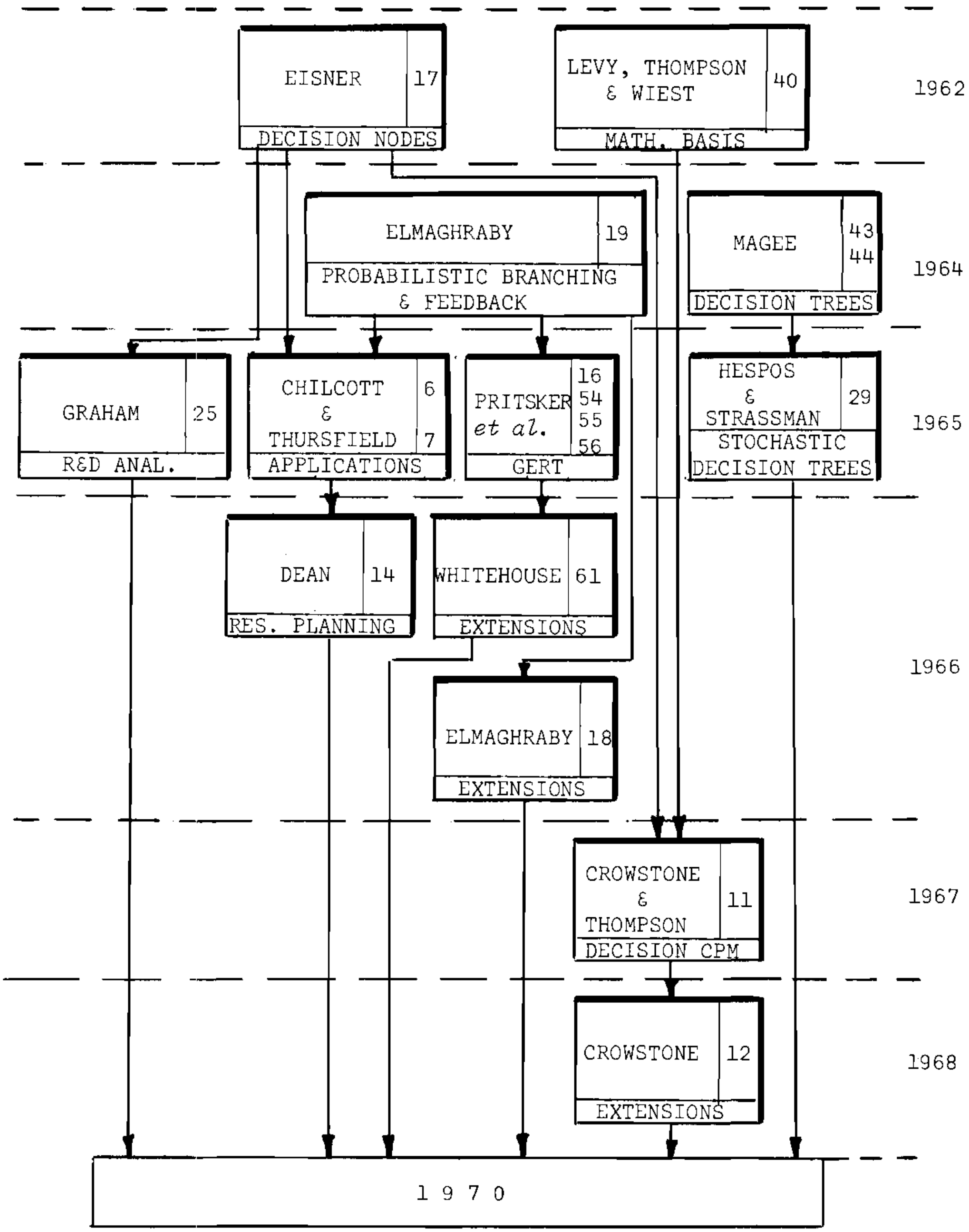


Figure 5. Research Done in Decision Networks

ideas to research management and project evaluation. Dean (14) investigated the application of stochastic networks to research planning. Graham (25) extended the ideas of Eisner to analyze R and D expenditures, and finally, Crowstone and Thompson (11,12) introduced "Decision CPM."

Generalized Networks

Eisner (17) describes his method as a generalization of the PERT network for R and D that allows alternative procedures for accomplishing research tasks to be considered. Thus, he proposes a network with decision vertices (termed decision boxes--DB) to represent such situations:

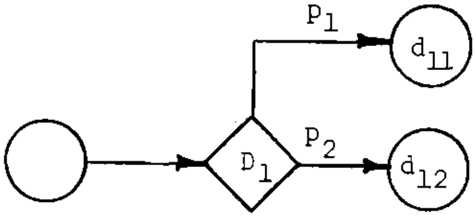


Fig. 6. Eisner's Decision Boxes

If decision vertex D_1 is reached, then d_{11} occurs with probability p_1 and d_{12} with probability p_2. *Note that it is not certain at the outset which of these alternatives will be chosen.* Eisner solved a simple example by *enumerating* the outcomes and evaluating their respective probabilities. To evaluate the favorable outcome, he introduced the notion of "entropy." Time values are introduced only after the entropy calculations.

Essentially Eisner constructs a decision tree with time values. The proposed solution by enumeration cannot be regarded as very useful.

Also, using the "entropy" as a means of evaluating the outcomes is questionable. At the time, this work had some value as being one of the first works in this direction.

Elmaghraby's important papers (18,19) account for a *significant departure from the well-known directed acyclic network* having either deterministic or probabilistic activity durations. Based on the ideas of Eisner (17) regarding the probabilistic branching of activities from a given event, Elmaghraby evolves an algebra for generalized activity networks. He introduces a series of logical relations to network formulation. Each event has probability of occurrence, and some parameter-time, attached to it. In (19) only deterministic values for this parameter were discussed, whereas in (18) the formulation was extended to the case of probabilistic times. Elmagrahby defines logical relations that may exist between events, and a graphical symbol is defined for each relation, so that the relations can be expressed graphically. The relations and their symbols are summarized in Fig. 7 and Fig. 8.

Graphic Symbol	Type of Receiver	Type of Source
	And	Must-follow
	Inclusive-or	Must-follow
	Exclusive-or	Must-follow
	And	May-follow
	Inclusive-or	May-follow
	Exclusive-or	May-follow

Figure 7. Logical Relations (19)

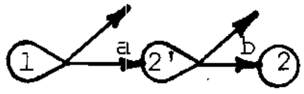
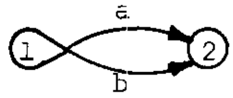
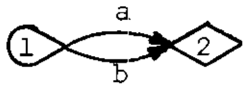
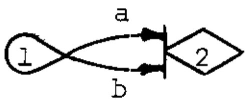
	Original Network	Name	Equivalent Arc
a.		Series	$p_e = p_a \cdot p_b$ $t_e = t_a + t_b$
b.		Parallel "And"	$p_e = P(a \cap b)$ $t_e = max(t_a, t_b)$
c.		Parallel "Inclusive-or"	$p_e = P(a \cup b)$ $\bar{t}_e = [t_a p_a + t_b p_b + \{\min(t_a;t_b) - t_a - t_b\} P(a \cap b)]/p_e.$
d.		Parallel "Exclusive-or"	$p_e = P(a \cup b) - P(a \cap b)$ $\bar{t}_e = [t_a p_a + t_b p_b + M\ P(a \cap b)]/p_e$
e.		Self Loop	$p_e = p_b/(1-p_e)$ $\bar{t}_e = t_a + \{t_b + t_e p_e/(1-p_e)\}$

Figure 8. Network Algebra (18)

Elmaghraby suggests a *complete enumeration* of paths, and shows algebraically that for each such path a time and probability of occurrence may be determined. He concludes by combining path time and probability information for an overall expected value for project completion. Note that Elmaghraby combines time and probability information, whereas Eisner did not.

The applicability of this method to PERT-CPM network is rather limited. These networks are composed mainly of "AND" nodes, and as Elmaghraby himself notes:

> . . . an AND node can be replaced with two Exclusive-or nodes. . . . Therefore, it seems that the price paid for the use of an already established theory is the enlargement of the original logic of the network. Whether such trade-off is advantageous or not can be answered only from empirical experience.

The work of Elmaghraby received further momentum as a result of four recent papers by Pritsker (56), Pritsker and Happ (55), Drezner and Pritsker (16) and Whitehouse (61). A new term was given to the exploration of stochastic activity network: "GERT" - for Graphical Evaluation and Review Technique.

Whitehouse (61, p.1) describes GERT as a procedure which combines the disciplines of flow graph theory, moment generating functions and PERT to obtain solution to stochastic problems. In (55), a stochastic network is defined as having the following properties:

a) A branch has associated with it a probability that the activity represented by the network will be performed.

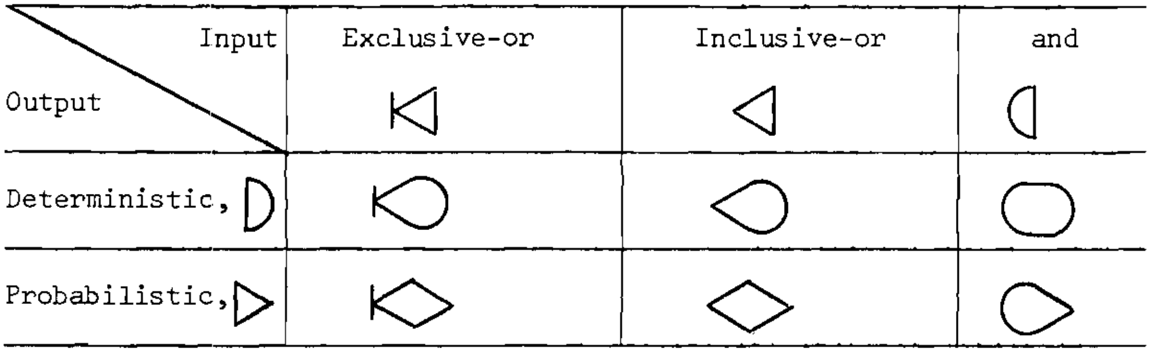
b) Other parameters describe the activities which the branches represent. These parameters may be additive, such as time, or multiplicative such as reliability.

c) A realization of a network is a particular set of branches and nodes which describe the network for one experiment.

Note that property (b) above creates difficulty for CPM-PERT type networks, as time is not purely additive.

GERT derives both the probability that a node will be realized, and the conditional moment generating function (M.G.F.) of the elapsed time required to traverse between *any two nodes*.

With the help of Signal Flow Graphs, Pritsker and Happ (55) have been able to outline an algebra for the solution of stochastic networks. They formalized node-symbols as in Fig. 9.



Exclusive-or - The realization of any branch leading into the node causes the node to be realized; however, one and only one of the 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 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 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.

Probabilistic - At most one branch emanating from the node is taken if the node is realized.

Figure 9. Node Characteristics and Symbols--GERT (55).

Pritsker and Happ consider the principles of network reduction for *exclusive or nodes only*. Essentially they employ a moment generating function (M.G.F) $M_t(s)$, continuous or discrete, associated with activity duration (t) which is further transformed by the conditional probability p oriented to the branch to form a W function where:

$$W(s) = pM_t(s)$$

The outcome of this two-stage transformation results in a system of *linear* independent equations which is amenable to solution by flowgraph techniques. The two key features of network reduction at the transform level are: (1) The M.G.F. of the sum of time values is the product of the M.G.F. of individual time values; (2) the M.G.F. of a mixture of two distributions is the sum of M.G.F. of individual distributions, each one being weighted by their conditional probabilities. At the two-stage transformation level this is reduced to simple addition of the corresponding functions as in Fig. 10.

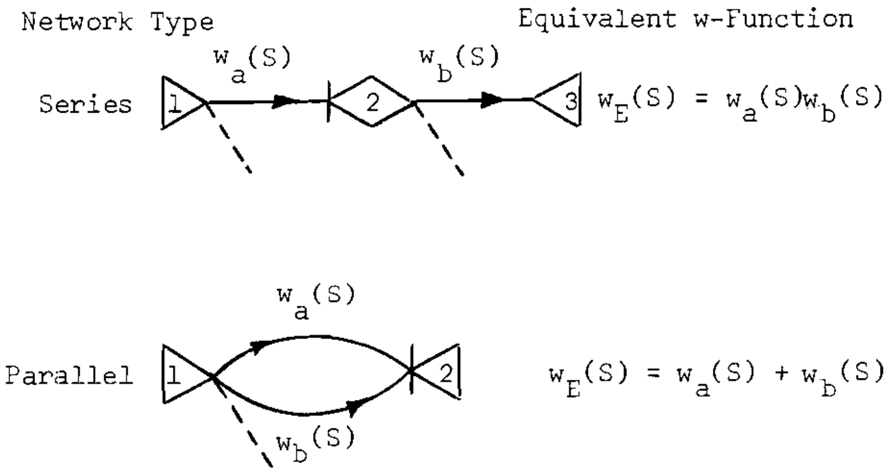


Figure 10. w Function (55)

Whitehouse (61) has investigated different approaches for obtaining the distribution function from an M.G.F. so as to ascribe confidence limits to system performance.

In (16) the application of GERT to a space vehicle countdown is reported. The authors ended up using simulation for system performance.

An attempt to solve a network with "AND" nodes was made, and the reported result is: ". . . the programming and storage problems associated with computations appear formidable. The area, however, is one for future research."

The applicability of GERT to CPM-PERT networks is doubtful. Pritsker (56, p.101) notes that "the analysis of AND nodes is similar to the analysis of PERT-type networks. To date there is no exact solution to the analysis of PERT networks that is computationally tractible."

The same observation was reiterated later by Wolfe (62) who wrote his Ph.D. dissertation under Pritsker. He states that "Although GERT appears to hold promise in aiding the analysis of stochastic networks; at present no general method for analyzing PERT and CPM type networks has been incorporated in this procedure."

The reason for the difficulty encountered with CPM-PERT networks is that an "AND" node is regarded as nonlinear, and therefore GERT cannot transform it in a simple fashion to a linear independent equation and apply flow graph techniques to it.

The works of Chilcott and Thursfield (6,7) illustrate applications of Eisner's ideas in research management and project evaluation. Graham (25) applies Eisner's decision box ideas to analyze R and D expenditure. Solving a simple example, he enumerates all paths and takes the expected cost of each path. A discounted cash flow analysis is also introduced. No time element in the sense of CPM-PERT is considered.

Dean (14) concentrates his efforts on the analysis of stochastic networks for research planning. After discussing the three planning techniques of CPM, CPM-minimum cost, and PERT Time, he concludes that these techniques are not applicable to research projects, because the sequence of events and activities is known with certainty. A research project is viewed as a process of acquiring knowledge sequentially. This leads to the development of the stochastic network concept--stochastic because the nodes of the network are not known with certainty in advance. The stochastic network represents the researcher's planned acquisition of knowledge. The change in this plan is represented by a configurational change in the research network.

Four models are presented for solving four decision problems in research planning:

a) Sequencing of research tasks
b) Selection of technical alternatives
c) Funding of components concepts
d) Cost allocation across systems.

The solution of the first model yields that high risk tasks should be performed first if the costs are the same, and the cheaper tasks should be performed first if the risks are the same. Models (b) through (d) are solved sequentially in that the results of each decision problem are used in the next decision problem. Discrete Dynamic Programming is used to solve each one of these models--for a specific example. The solution obtained is cost vs. maximum probability of success for various alternatives. Dean states that solutions may be obtained manually if the number of alternatives is less than 20, and otherwise may be programmed for an electronic computer. Note that none of the models considers the time element directly.

Decision Trees

The two papers by Magee (43,44) are an extension of decision tree analysis developed in various texts of statistical decision theory. The idea is that a choice among alternatives can be made, and this choice of a specific alternative is followed by certain stochastic events, with known probabilities, where the decision maker has no control over the outcome. Each combination of decisions and stochastic events has a different outcome.

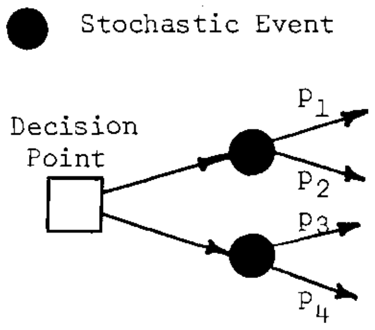


Figure 11. Decision Tree

One can see that this case is different from the decision box method presented by Eisner. Magee used this method for analysis of capital investments, by using discounted cash flows and expected values.

Hespos and Strassmann (29) introduce the concept of "Stochastic decision trees," with the following features:

a) All quantities and factors, including stochastic events, can be represented by continuous, empirical probability distributions.

b) The information about the results from any or all possible combinations of decisions made at sequential points in time can be obtained in a probabilistic form.

c) The probability distribution of possible results from any particular combination of decisions can be analyzed using the concepts of utility and risk.

This method calls for eliminating the stochastic node and replacing it with a probability distribution. Using simulation, a single branch is selected at each node for which a probability distribution exists. The result of the simulation is a probability distribution of the variable of interest. The authors report the results of solving a stochastic decision tree problem with GPSS simulation.

Deterministic Networks

Crowstone and Thompson (11), based upon the idea of Eisner, introduced the activity alternatives into CPM network. A new term was coined for this analysis: Decision CPM. Thus, if there is a number of competing methods of performing some of the activities, each method having a different cost, a different time duration and different technological dependencies, these possibilities are included in the project network. Among all these alternative, the ones minimizing total project cost are selected. The authors recommend the repeated use of the same method during the execution phase of the project.

Using activity-on-node representation of a network, and following Levy (40), they define a decision project graph G as a graph with nodes representing activities and directed line segments connecting two nodes s_{ij} (in activity set i) and s_{mn} (in activity set m), if *and only if* s_{ij} *is an immediate predecessor of* s_{mn}, i.e., $s_{ij} << s_{mn}$. Additional interdependencies of contingent relations between activities are defined too (Crowstone (12) later calls these interdependencies "other constraints." Their nature is not disclosed, but this might be, for example, the non-simultaniety constraint discussed by Davis (13)). A decision project is defined as a set J (a set of activity-sets S_i) together with the specified interdependencies and the precedence relationships. In addition, all alternatives for a given activity have identical predecessor successor relations. Exactly one of the activities of each decision set must be performed--and in contrast to Eisner, this activity is performed with *certainty*.

The outcome of decisions based on project cost analysis would be to eliminate a cluster of activities from the decision project network to obtain the final project network used in regular CPM analysis. Two methods are suggested to achieve this pruning: 0-1 integer programming, and heuristics with partial enumeration. Crowstone and Thompson illustrate with a few examples the several ways in which the 0-1 variable can be used to represent interdependencies among decisions.

This work has two main shortcomings: the fact that all alternatives of a given decision set have identical predecessor-successor relations makes the problem very unrealistic. The exponential growth of constraint sets as the number of activities are increased poses severe practical problems for a 0-1 integer programming solution. The authors recognized this by stating:

> In very large problems the integer programming solution technique becomes impractical because of the resulting large number of constraints and variables. For that reason we have developed heuristic solution techniques for solving the problem.

The heuristic method is a modification of the one given by Moder and Phillips (49, pp.109-122).

Crowstone's paper (12) is an extension of the previous work. He removed the above requirement of identical predecessor-successor relations, and added a method that is supposed to reduce the size of the integer programming formulation and also suggested a branch and bound solution.

The reduction method amounts to enumerating 2^h paths through the network, where h is the total number of decision alternatives. Some of these paths are not feasible, because of the "other constraints." The author developed a reduction algorithm, however, it is not certain whether the time saved in the integer programming solution is not consumed by this reduction algorithm.

The branch and bound technique developed solves, at a particular node of the tree, an integer programming problem of the form:

$$\text{Min } c_j = \sum_{i=1}^{h} \sum_{j=1}^{k(i)} c_{ij} d_{ij}$$

subject to

$$\text{Acceptance} \quad d_{ij} = 1$$

$$\text{Exclusion} \quad d_{ij} = 0$$

interdependency

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

plus "other interdependency constraints."

c_{ij} is the cost associated with decision node ij. This is the cost of node (ij) *plus* the cost of regular nodes associated with decision ij. No method is given, here or in the regular 0-1 solution, indicating how this cost c_{ij} should be evaluated.

Crowstone (12) states (p.40) that "In order to complete the project, one of the jobs from each job set must be completed . . ."

This is a limiting assumption. There might be cases where a decision node will not be considered at all, and therefore *at most* one activity (job) from each decision set (job set) must be selected.

The rest of this paper discusses resource constrained decision networks, and shows some applications of Decision CPM.

Concluding Remarks

The following conclusions may be drawn from the literature survey presented herein:

1. Although some research has been done in decision networks, the problem of time-cost trade off, for different sets of decisions has not been investigated. For the regular CPM analysis, this problem has been treated by introducing "crash time," and "normal time" for each activity. (See, for example: (1,22,33,48,49,57).) The works of Crowstone and Thompson are a first step in this direction for decision networks.

2. The attempts made so far for the case of certainty centered on minimizing cost only, utilizing integer programming, branch and bound or heuristic programming.

3. The stochastic cases of decision networks received some attention in the literature. All these approaches use expected value as the sole criterion of choice, only one parameter at a time is considered (cost or time), and a project decision network is rarely treated explicitly.

CHAPTER III

MATHEMATICAL BACKGROUND

Introduction

The purpose of this chapter is to present some of the definitions, concepts and theorems relevant to this research.

Definitions

Project

A project is a collection of well-defined activities which, when completed, mark the end of the project. These activities are partially ordered; i.e., certain subsets must be performed in a given sequence.

The Project Network

Each project can be modeled by a project network (or graph). This necessitates the following definition. A *directed network* or directed *linear graph* G is a set $M = \{m_i | i=1...n\}$ together with A which is a subset of M×M, where $M{\times}M = \{m_i, m_j | m_i \epsilon M, m_j \epsilon M\}$. The elements of M are called nodes, and the elements of A are called arrows (or arcs). In the research that follows, a directed network G will be denoted by G(M,A), and the term directed will be suppressed.

There are two different methods of modeling a project--activities on arrows and activities on nodes.

Activities on Arrows (A-O-A)

In this method, arrows represent activities, and nodes represent events. All immediate predecessors of a given activity connect to a node at the tail of the activity arrow, and all immediate successor activities emanate from the node at the head of the activity arrow. Thus, a node marks the event of completion of all activities leading into the node. The precedence relationships among activities is shown by the manner in which the activities are connected through events. In order to portray accurately all predecessor relationships, "dummy activity" must often be added to the project graph.

Activities on Nodes (A-O-N)

In this method, nodes represent activities and arrows indicate the precedence relationships among the activities. For convenience, all nodes without predecessors are connected to a node marked "Start." All nodes with no successors are connected to a node marked "Finish." *Throughout this research, the A-O-N method will be used.* Additional parameters which will be included in the network are the duration of each activity and its cost. The duration and cost of "Start" and "Finish" are taken to be zero.

Types of Nodes

Throughout this research, six different types of nodes will be encountered. Definitions of each type follow (refer to Fig. 12).

Activity Node. This node describes one and only one activity. This is the basic node for an A-O-N network, and is represented by a circle. Each activity node is denoted by m_i, and the set of all activity nodes is:

$$M = \{m_i\}$$

M actually contains more than one m, but the range of i was omitted for notational simplicity. The same approach has been used for the rest of the nodes' notations.

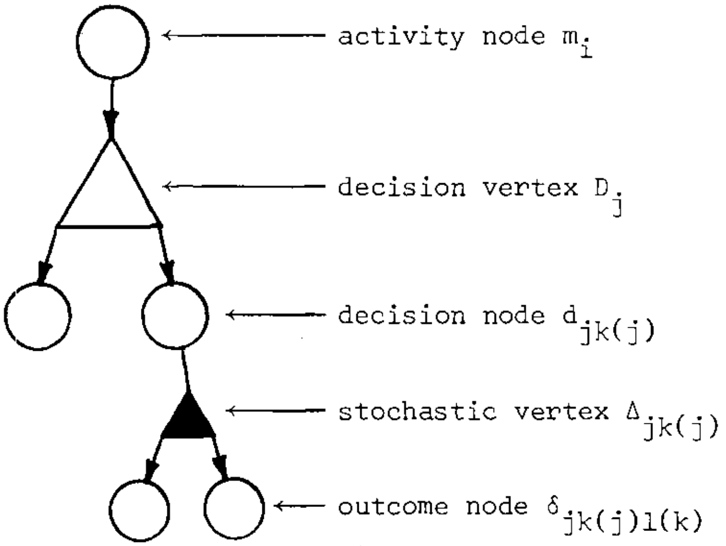


Figure 12. Types of Nodes--Schematic Illustration

Decision Vertex. From this node emanates a decision set with a finite number of elements, out of which *at most* one element must be selected. This node is illustrated by a triangle, and is denoted by D_j. Each activity node may or may not have a decision vertex following it; also, each activity node may have more than one decision vertex. Note that the subscript j on the decision vertex is not directly related to the subscript i on the activity node, for example, at m_{15} one could have D_2.

The set of all decision vertices is:

$$D = \{D_j\}$$

Decision Node. This node describes a decision alternative which is an element of the decision set D_j. It is illustrated by a circle and denoted by $d_{jk(j)}$. In the subscript on d, k(j) means that k depends on j also.

$$D_j = \{d_{jk(j)}\}$$

Stochastic Decision Vertex. This is essentially a decision vertex with a probability function distributed over its elements. The notation and graphical symbol are the same as that of the decision vertex D_j. Hence, it is not illustrated in Fig. 12.

Stochastic Vertex. From a stochastic vertex emanates a finite set of mutually exclusive stochastic elements with a probability function distributed over its elements. If the stochastic vertex is encountered, *exactly* one of its elements is realized. Of course, if this vertex is not encountered, none of its elements is realized. This node is illustrated by a small shaded triangle and is denoted by $\Delta_{jk(j)}$, since, if this node exists, it always follows a decision node $d_{jk(j)}$. Also:

$$\Delta = \{\Delta_{jk(j)}\}$$

Outcome Node. This node describes an outcome which is an element of the stochastic vertex. A probability is associated with each outcome node. This node is illustrated by a circle and is denoted by $\delta_{jk(j)l(k)}$.

In the subscript on an outcome node, l(k) means that l depends on k. Also note that

$$\Delta_{jk(j)} = \{\delta_{jk(j)l(k)}\}$$

The hierarchy among these nodes is as follows:

$$M = \{m_i\}$$

$$D \supset \{d_{jk(j)}\} \supset \{\Delta_{jk(j)}\} \supset \{\delta_{jk(j)l(k)}\}$$

where $\supset$ is the usual set notation for contains.

The duration and cost of a decision vertex, stochastic decision vertex and stochastic vertex is always zero. Note that the decision maker can control, at some stage of the project life, the alternative selected at a decision vertex (deterministic or stochastic); however, he can never control the outcome associated with a stochastic vertex.

Types of Networks

Four major types of networks will be encountered throughout this research (two less important types will be introduced later). The following are definitions of each type.

Standard Network. This is the acyclic directed network previously defined where all nodes are activity nodes m_i. The network has one "Start" node (labeled "S") and one "Finish" node (labeled "F"). The duration and cost of "S" and "F" are zero. Thus, this network is the ordinary project network of the CPM variety (see Fig. 1).

Decision Network. A decision Network $G(J,A)$ is a directed acyclic network as defined before, where the set of nodes J is given by:

$$J = M \cup D = \{j_i\}$$

and

$$M \cap D = \emptyset$$

Essentially, this is a deterministic decision network (see Fig. 2).

Stochastic Decisions Network. This is the decision network $G(J,A)$ with a probability distribution associated with the elements of each D_j (see Fig. 3).

Stochastic Outcomes Network. A Stochastic Outcomes Network $G(\Lambda,A)$ is a directed acyclic network where the set of nodes Λ is given by:

$$\Lambda = J \cup \Delta = M \cup D \cup \Delta = \{\lambda_i\}$$

where

$$M \cap D \cap \Delta = \emptyset$$

$$M \cap D = \emptyset \quad M \cap \Delta = \emptyset \quad D \cap \Delta = \emptyset$$

and there is a probability distribution associated with each $\Delta_{jk(j)}$ (see Fig. 4).

Mathematical Basis

The concepts of this section are developed mainly for the decision network G(J;A). A modification of these concepts will be used in Chapters V and VI for the stochastic decisions network and stochastic outcomes network.

Reduction of a Decision Network to a Standard Network

Given a decision network G(J;A), then a certain set of activities must be performed in order to complete the project. These activities are as follows:

a) *At most* one $d_{jk(j)}$ for each D_j, call this $d^*_{jk(j)}$.

b) Choose all or part of $M = \{m_i\}$, denote this subset of M by $M^* = \{m^*_i\}$.

Then, it is possible to reduce the decision network to a standard network $G(J^*,A^*)$.

where $G(J^*,A^*)$ is the standard network obtained by decision network reduction.

Hence, $J^* \subset J$; $A^* \subset A$

The reduction of G(J,A) to $G(J^*,A^*)$ is done using the following procedure:

a) All nodes $d^*_{jk(j)}$ become elements of the standard network.

b) Any activity node with at least one of its immediate predecessors being an element of the standard network, is also an element of the standard network.

c) All decision vertices are eliminated as follows: If at least one of the immediate predecessors of D_j is an element of the standard network, then it is connected with an arrow to $d^*_{jk(j)}$. If no predecessor of D_j is an element of $G(J^*,A^*)$, then D_j is eliminated along with all its $d_{jk(j)}$.

d) Any node other than D_j that is eliminated from the network, then all its incoming and outgoing arrows are eliminated. If this leaves some adjacent node without incoming arrows, this node is eliminated too and the process continues until no further nodes can be eliminated. Note that "S" and "F" are always elements of the standard network.

Following is an example of the above procedure.

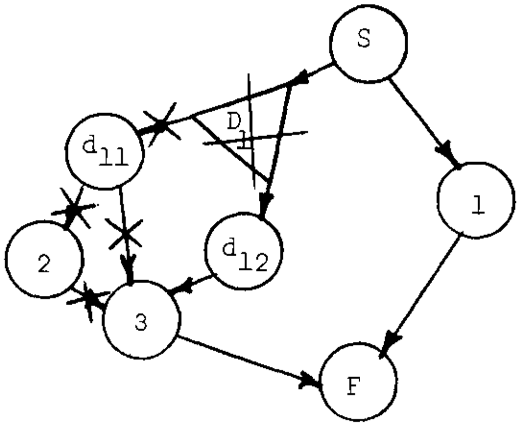


Figure 13. Decision Network Reduction

Referring to Fig. 13, one has:

$$J = \{S,1,2,3,D_1,F\}$$

and

$$D_1 = \{d_{11}, d_{12}\}$$

Now if

$$d^*_{1k(1)} = d_{12}$$

then d_{11} is eliminated since there can be at most one $d_{jk(j)}$ for each D_j.

Hence

$$M^* = \{S, 1, 3, F\}$$

and the standard network is:

$$J^* = \{S, 1, 3, d_{12}, F\}$$

Since d_{11} was eliminated, then, following rule (d) the three arrows entering into and emanating from d_{11} are also eliminated, and correspondingly are marked with an "X." As a consequence, activity node 2 and the arrow emanating from it are also eliminated. Note that because of rule (b), activity node 3 remains an element of the standard network.

Throughout this research, no interdependencies are considered among $d_{jk(j)}$ for different j, as of the type mentioned by Crowstone (12).

Precedence Relations

The following discussion considers precedence relation in a standard network; however, the concepts and notations hold true for all other types of networks, with some minor differences, as indicated.

Let "<<" denote a precedence relation between two nodes; for example, if for some m_i, $m_j \epsilon M$, either m_i is an immediate predecessor of m_j, or, equivalently, m_j is an immediate successor of m_i and this is denoted by

$$m_i << m_j.$$

The set

$$(IP)_k = \{m_i | m_i << m_k\}$$

is the immediate predecessor set of some m_k. Similarly:

$$(IS)_k = \{m_i | m_k << m_i\}$$

is the immediate successor set of some m_k.

The network G{M;A} is a planar graph with nodes representing activities, and a directed line segment exists from some node m_i to some node m_k if and only if $m_i << m_k$. Note that $S \epsilon M$ is the only activity without predecessor, and $F \epsilon M$ is the only activity without successor.

A path in G is a set of nodes $m_i, m_j \ldots m_r$ for which the immediate predecessor relation holds as follows:

$$m_i << m_j << m_k << \cdots << m_r$$

A cycle in G is a closed path of the form:

$$m_i << m_j << m_k \cdots << m_r << m_i$$

A graph is acyctic if and only if it has no cycles.

$m_i < m_k$ means that m_i must precede m_k, but must not immediately precede m_k. This is possible if and only if there is a set of nodes $\{n_j\}$ such that:

$$m_i << n_1 << n_2 \cdots << n_r << m_k$$

i.e., $m_i < m_k$ if and only if there is a *path* from m_i to m_k in the network G.

All the above concepts and definitions hold true if M is replaced with J, J^* or A. Some differences, however, do exist between the standard network and the other networks.

For a standard network, $m_i << m_j$ means that m_i must be completed before the commencement of m_j. For all types of decision networks, this is still true for the activity nodes; however, $m_i << D_j$ means that $m_i << d_{jk(j)}$ for all k(j) where j is fixed.

Let:

$m_i \not\ll m_j$ mean that m_i is not the immediate predecessor of m_j.

Then, the following precedence relations hold for the various decision networks.

$$D_j \ll d_{jk(j)}$$

$$d_{jk(j)} \ll \Delta_{jk(j)}$$

$$\Delta_{jk(j)} \ll \delta_{jk(j)l(k)}$$

$$D_j \not\ll m_i$$

$$m_i \not\ll \Delta_{jk(j)}$$

$$m_i \not\ll \delta_{jk(j)l(k)}$$

The above precedence relations always exist for the various decision networks. Other relations may or may not exist.

Finally the predecessor matrix B for a project with n nodes is the n×n matrix with components b_{ij} defined as follows:

$$b_{ij} = \begin{cases} 1 & \text{if } m_i \ll m_k \\ 0 & \text{otherwise} \end{cases}$$

This is essentially a "From To" matrix. Note that B^T is the successor matrix.

Critical Path Computation

For the purpose of this research, only a short form of the critical path computation is required. Once the decision network has been reduced to a standard network, normal CPM analysis can be applied whenever applicable, and it is possible to evaluate early start, late start, slack, etc. These terms will not be defined here, and the interested reader should consult Moder (49).

The following discussion describes concepts used in standard networks. The same concepts will be used in the formulation of the three problems discussed in this research.

A critical activity is an activity which contributes directly to the overall project time.

Obviously, every project has at least one critical activity. This leads to the following theorem:

Theorem 1. a) There is at least one path, called "critical path," from "S" to "F" such that every activity on the path is critical. b) Every critical activity lies on such a path. c) The sum of the activity durations on every critical path is T, which is at least as large as the sum of the activity durations on every other path from "S" to "F." [For proof, see Levy (40)].

Let t_i be the duration of activity m_i (this is the time required to complete activity m_i). Let Z_k represent the kth path from "S" to "F," and let $L(Z_k)$ represent the length of this path. Thus, if Z_1 = $S << m_2 << m_3 \cdots << m_{n-1} << F$, where $m_1 = S$, $m_n = F$, then:

$$L(Z_1) = \sum_{i=1}^{n} t_i$$

Then, the length T of the critical path is

$$T = \max_{k} L(Z_k)$$

The critical path can be determined after using a forward pass to evaluate ES (early start), and EC (early completion). The following algorithm, due in part to Davis (13), will be used:

1) Assign ES of "S" as 0.

2) Proceed along any path from "S." To each activity assign ES and EC as follows:

$$ES(m_k) + t_k = EC(m_k),$$

where $ES(m_k)$ is the early start of activity m_k, etc.

$$ES(m_k) = \max_{\text{all } m_i \epsilon (IP)_k} EC(m_i)$$

3) If an activity m_i is encountered such that one or more members of $(IP)_k$ have not been assigned an ES and EC, temporarily defer further consideration of this path.

4) Find any other activity whose predecessors have all been assigned an EC but which have not been assigned an ES.

5) Proceed along any path starting with such an activity until an impasse is reached as cited above, or until "F" is assigned an EC.

6) The forward pass is complete when "F" is assigned an EC.

7) Start with "F" and proceed backwards through the network.

8) Whenever a merge event is encountered, the critical path(s)

follows the activity(s) for which

$$EC(m_i) = ES(m_j) \quad \text{for } m_i << m_j$$

The concept of the critical path can be extended to the case of two nodes, m_i, m_j, provided there is at least one path from m_i to m_j. If there is such a path, the critical path will give the time required to reach m_j from m_i.

CHAPTER IV

THE CASE OF CERTAINTY

Introduction

This chapter develops the methodology for the case of certainty, as defined previously. The approach used evaluates first the two extreme points satisfying the conditions of this problem (referred to later as o_1^* and o_N^*), and then evaluates all points in between, together yielding the time-cost trade-off desired. This is done by solving three separate problems, as follows:

a) Minimum time problem

b) Minimum cost problem

c) Time-cost trade-off problem

The problems are discussed in this order. The basic approach used is a dynamic programming formulation.

Minimum Time Problem

The minimum time problem is defined as follows: Given the decision network G(J,A), select, for each decision vertex (decision set) D_j, at most one decision node $d_{jk(j)}^* \epsilon D_j$ so that the completion time of the project will be minimized. Find the minimum cost associated with the minimum time.

The approach taken in solving this problem is to transform the decision network into a dynamic programming[1] model. To structure a DP model of the decision network, one defines the following elements and conventions:

i) The decision variables are grouped in stages, and the stages considered sequentially.

ii) The only information about previous stages relevant to selecting optimal values for the current decision variables is summarized by a so-called state variable (or input variable).

iii) The current decision, given the present state of the system, has an influence on the state at the next stage, and it is represented by a "return."

In the following section a DP model of a simplified decision network is introduced, along with a short review of DP concepts, and this is later extended to the general case of a decision network.

DP Model--Simple Formulation

A simplified decision network as shown in Fig. 14 can be described by the standard serial DP model. In this case, the model will have three stages, as shown in Fig. 15 (the stages are numbered backwards).

The decision variables associated with each stage are composed of the proper decision sets of the network. (For a more complicated problem, it is not immediately obvious how to form the decision variable of a particular stage of the DP model. An algorithm to achieve this is introduced later.)

---

[1] Throughout this research, dynamic programming will be denoted as DP.

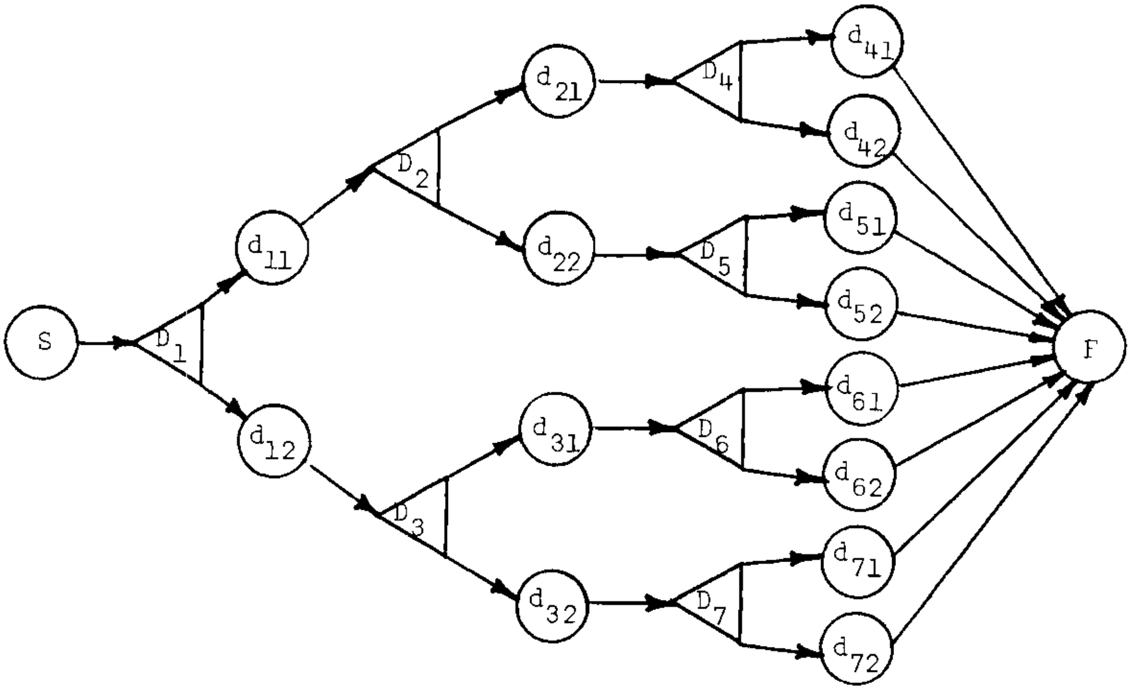


Figure 14. A Simplified Decision Network

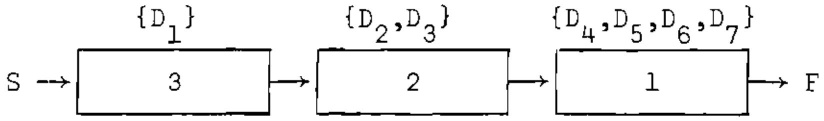


Figure 15. A DP Model for the Simplified Network

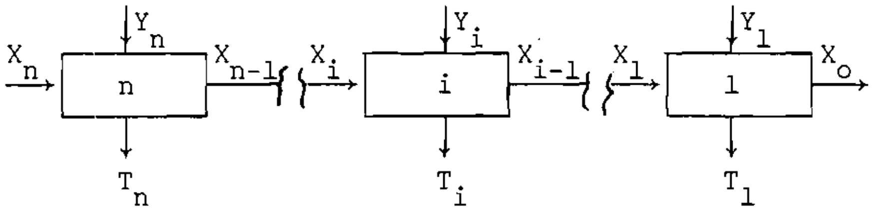


Figure 16. Serial DP Model

The next step in developing a DP model is defining the stage transformation of the state variable. Referring to Fig. 16, for the standard serial DP model one has (stages are numbered backwards)

$$X_{i-1} = g_i(X_i,Y_i)$$

where X_i is the *state variable* of stage i, defined as:

$$X_i = \{x_{ij}\}$$

and

$$Y_i = \{y_{ij}\}$$

is the decision variable of stage i. g_i is the forward state transformation at stage i. This transformation, for a decision network, has the convenient form of:

$$X_{i-1} = g_i(Y_i) = Y_i$$

i.e., the decision variable of stage i is the state variable of stage (i-1). Both can be n-dimensional. Also, one always has

$$X_n = \{S\}$$

and $$X_o = \{F\}$$

where both X_n and X_o are element state variables.

To complete the DP model formulation, the returns associated with each stage have to be defined. Obviously enough, if the objective is to minimize the total project time, the return matrix of stage i would be composed of the durations of the decision nodes associated with stage i.

Let: $T_i(X_i,Y_i)$ be the return matrix for stage i--time, and let $t_{jk(j)}$ be the time return of decision node $d_{jk(j)}$ (this is the duration in this case). Then, if: $y_{il} = d_{jk(j)}$, the lth column of $T_i(X_i,Y_i)$, for *all* X_i, would be equal to $t_{jk(j)}$, whenever the transformation $g_i(X_i,Y_i)$ exists. Thus, for stage 2 of Fig. 15, this would be:

$$T_i(X_i,Y_i) = \begin{array}{c} \\ d_{11} \\ d_{12} \end{array} \begin{array}{c} \begin{array}{cccc} d_{21} & d_{22} & d_{31} & d_{32} \end{array} \\ \left[\begin{array}{cccc} t_{21} & t_{22} & x & x \\ x & x & t_{31} & t_{32} \end{array}\right] \end{array}$$

Note that each row of $T_i(X_i,Y_i)$ has entries only for columns associated with the same D_j. This statement is true also for the more general case introduced later.

All possible project times for this case will be obtained by evaluating the length of all paths from S to F.

Let: U^m be the mth set of decision nodes $d_{jk(j)}$ associated with the mth path. Then, the length of the mth path is:

$$L(Z_m) = \sum_j t^m_{jk(j)}$$

where $t^m_{jk(j)}$ are the time values associated with the mth path. To minimize the total project time, one has to evaluate:

$$L(Z^*) = \underset{m}{\text{Min}}[L(Z_m)] = \underset{m}{\text{Min}}(\sum_j t^m_{jk(j)})$$

The optimal policy, indicating the decision node (if any) to be selected for each D_j is given as:

$$U^* = \{d^*_{jk(j)}\}$$

This procedure is equivalent to minimizing the sum of stage returns of the DP model. A formulation of this type is generally known as the "Stage Coach" problem.

Using Bellman's principle of optimality, it is possible to write the recursive equations for solving this case, assuming that a backward solution approach is used.

$$f_{i(t)}(X_i) = \min_{Y_i} Q_i(X_i,Y_i) \qquad i=1\ldots n$$

$$Q_1(X_1,Y_1) = T_1(X_1,Y_1)$$

$$Q_i(X_i,Y_i) = T_i(X_i,Y_i) + f_{(i-1)(t)}(X_{i-1})$$

where $f_{i(t)}(X_i)$ denotes the minimum time at stage i as a function of the input variable, and $Q_i(X_i,Y_i)$ denote the i-stage time matrix.

Note that:

$$f_{(i-1)(t)}(X_{i-1}) = f_{(i-1)(t)}(Y_i)$$

DP Model--Extended Formulation

The general case of a project decision network is more complicated than the simple example of the previous section, where the equivalence with a DP model was almost obvious. Thus, for the decision network of Fig. 17 it is not immediately clear what the equivalent DP model is, with its input variable, decision variable and returns for each stage.[2] The model for the general case is not necessarily composed of serial stages, and as a consequence evaluating the optimal policy is somewhat different. This section develops the methodology for the general case, which includes four steps, as follows:

1) Network Decomposition into Decision Dependent Subnetworks, denoted DDS_i, i=1...m;

2) Formulation of a DP model for each DDS_i;

3) Determination of the minimum project time; and

4) Development of the DP solution.

Decision Dependent Subnetwork (DDS). Prior to rigorously defining the notion of DDS, a few examples are discussed to introduce the concept on an intuitive basis.

Suppose that the decision network of Fig. 18 is given.

One can immediately observe that there are two independent subnetworks, one associated with the decision vertex D_1, the other

---

[2] Throughout this research, whenever no ambiguity arises, D_j will be denoted by j, and $d_{jk(j)}$ by jk(j).

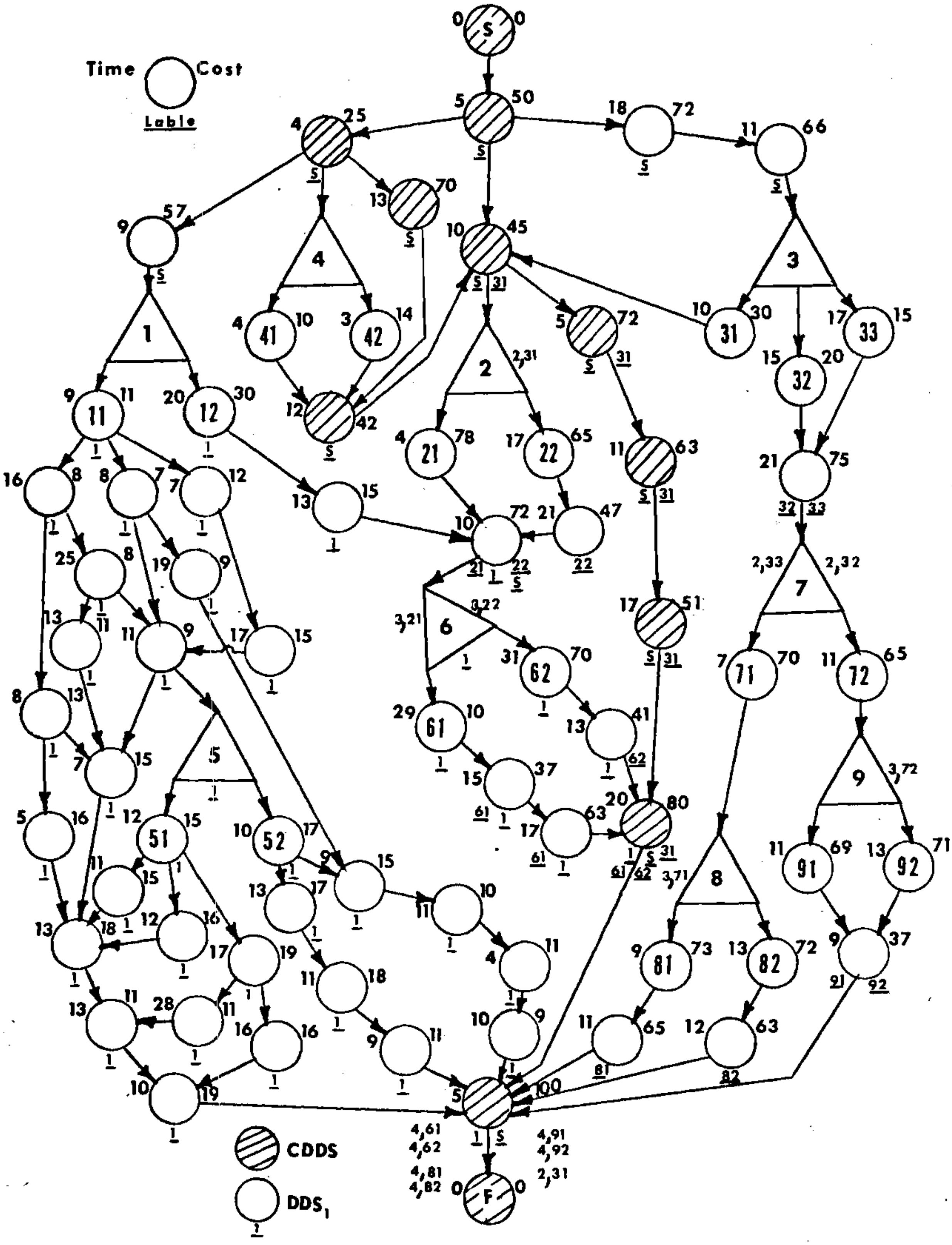


Figure 17. A Decision Network: An Example

associated with the decision vertex D_2. Thus, schematically this network could be illustrated as shown in Fig. 19. (For a clearer illustration, F is shown separately, but it is actually an element of both subnets 1 and 2.)

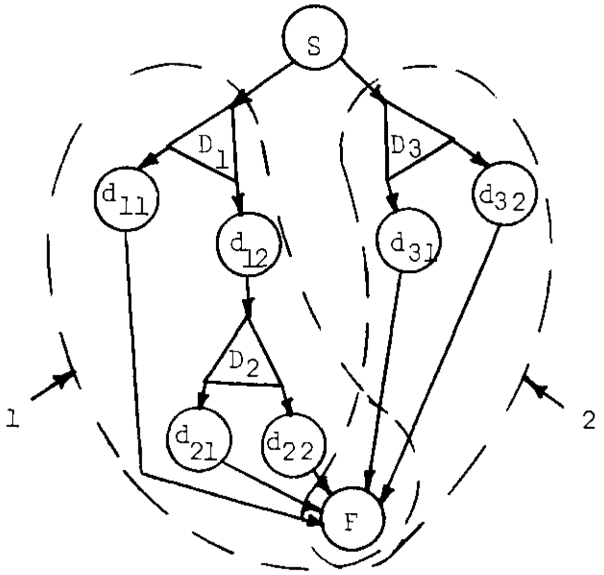


Figure 18. Decision Network

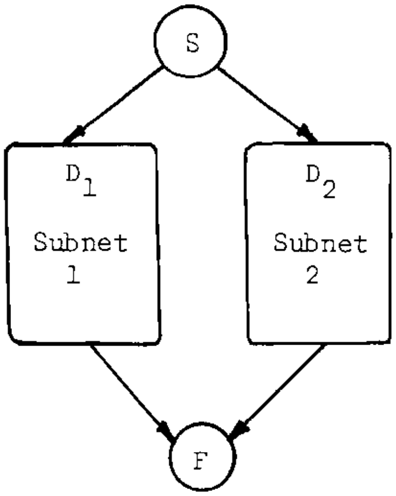


Figure 19. Decision Network--Schematic Description

Now, suppose the above decision network is modified as shown in Fig. 20.

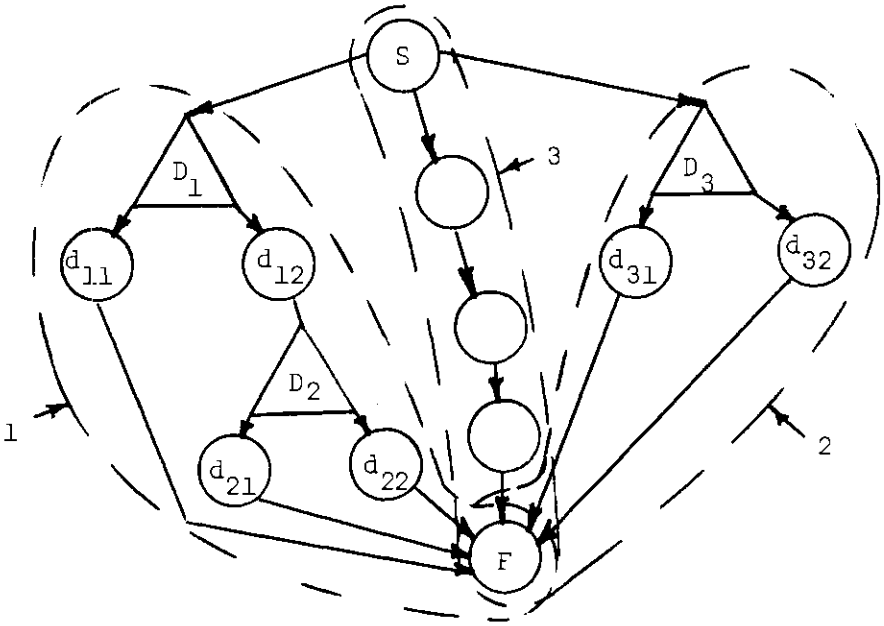


Figure 20. Modified Decision Network

Clearly, in this case there are three independent subnetworks; that is, two subnetworks as previously described and a third subnetwork not dependent on any decision vertex. To be consistent with the previous case, where every subnetwork was associated with a decision vertex, the first activity node S is regarded as a *conjunctive decision node*, where a conjunctive decision node is a node that all activities emanating from it are to be performed.[3]

Thus, subnetwork 3 can be regarded as dependent upon the

---

[3] This definition is similar to that of Eisner (17).

conjunctive decision node "S," and the decision network of Fig. 20 can be reduced to the one shown in Fig. 21.

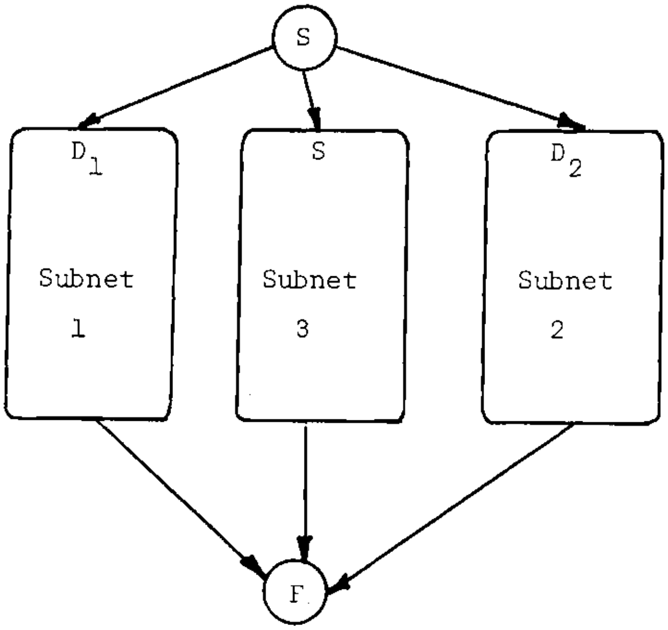


Figure 21. Decision Network--Schematic Description

Finally, assume the decision network of Fig. 18 is modified as shown in Fig. 22.

It remains possible to identify two subnetworks, 1 and 2; however, they are no longer independent, as decision vertex D_2 belongs to both subnetworks. The network of Fig. 22 is illustrated schematically in Fig. 23.

It is now possible to define the concept of a decision dependent subnetwork (DDS).

Definition. The decision dependent subnetwork $G\{J^{(s)};A^{(s)}\}$ of a network $G\{J;A\}$ is a connected network such that $J^{(s)} \subseteq J$, $A^{(s)} \subseteq A$, and the

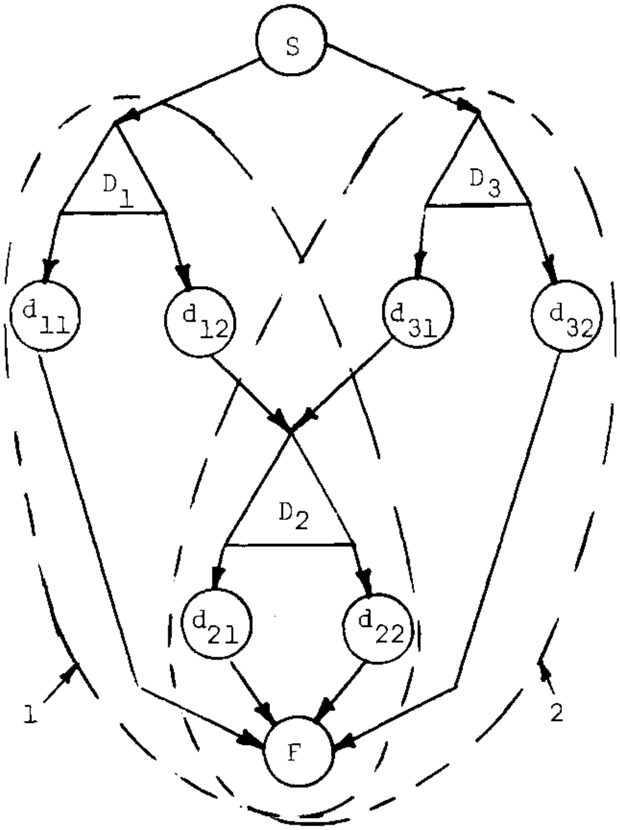


Figure 22. Decision Network with a Common Decision Vertex

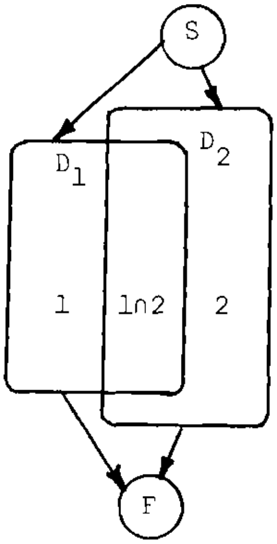


Figure 23. Decision Network--Schematic Description

following conditions hold:

(1) Exactly one decision vertex D_j is preceded by the conjunctive decision node "S," and $S \notin J^{(s)}$.

(2) For exactly one node $j_k^{(s)} \epsilon J^{(s)}$, the set $(IS)_k = \emptyset$ (i.e., this is the node "F").

(3) For every node $j_k^{(s)} \epsilon J^{(s)}$, DDS includes *all* elements of the set $(IS)_k$ (the immediate successors) but not necessarily all elements of the set $(IP)_k$ (the immediate predecessor).

Thus, referring to Fig. 22, one has:

$$\{d_{11},d_{12},d_{21},d_{22},F\} \in J_1^{(s)} \quad \text{for } DDS_1$$

$$\{d_{31},d_{32},d_{21},d_{22},F\} \in J_2^{(s)} \quad \text{for } DDS_2$$

Note that nodes $\{d_{21},d_{22},F\}$ are elements of both DDS_1 and DDS_2.

Note also that $\{d_{21},d_{22}\} \in DDS_1$ because $\{d_{21},d_{22}\} \in (IS)_{12}$, but, $d_{31} \notin DDS_1$, because $d_{31} \notin (IS)_{12}$, although $d_{31} \in (IP)_{21}$ and $d_{31} \in (IP)_{22}$.

Clearly, since $G\{J,A\}$ is a directed, acyclic network, so is $G\{J^{(s)},A^{(s)}\}$.

The previous definition has to be supplemented to include the case of subnetwork 3 of Fig. 20, i.e. the case where there is *at least* one path such that:

$$S = m_1 << m_2 << \cdots << m_n = F$$

where all nodes are activity nodes.

Definition. The conjunctive decision dependent subnetwork (CDDS) $G\{J^{(c)},A^{(c)}\}$ of a network $G\{J;A\}$ is a connected network such that $J^{(c)} \subseteq J$, $A^{(c)} \subseteq A$ and the following conditions hold:

(1) The conjunctive decision node "S" is the initial node of this network, and $S \in CDDS$.

(2) For every node $(j_k = m_k) \in J$ define:

$$(IS)_k^{(c)} = \left\langle \begin{Bmatrix} m_i/m_k << m_i, \text{ and} \\ \\ m_i << m_{i+1} << \cdots << F \end{Bmatrix} \right\rangle \forall k$$

Then, if $(IS)_k^{(c)} \neq \emptyset$, and $m_k \in CDDS$, then $(IS)_k^{(c)} \in CDDS$.

(3) $(IS)_k^{(c)} = \emptyset$ if $k = F$.

Conditions (2) and (3) above need some more elaboration. Condition 2 guarantees that all nodes which are elements of CDDS are going to be elements of some path from S to F that does not include any D_j. This implies condition 3. Thus, in the following example:

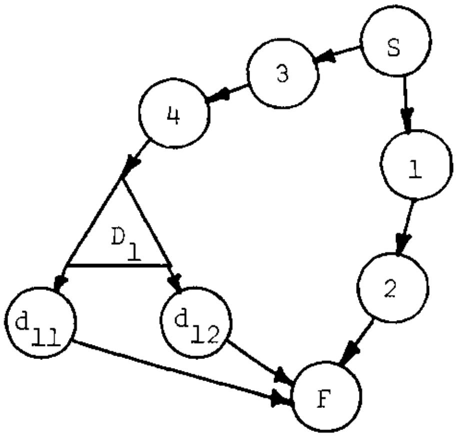


Figure 24. CDDS--An Example

$(IS)_4^{(c)} = \emptyset$ and therefore, node 4 should be eliminated. As a consequence, $(IS)_3^{(c)} = \emptyset$; in other words, nodes 3 and 4 are not elements of some path from S to F. Therefore

$$\{S,1,2,F\} \in CDDS$$

Clearly, every decision network has *at most* one CDDS associated with it (which may include more than one path from "S" to "F"). Also, the standard network is CDDS.

It remains to be shown now how a decision network can be decomposed into its DDS and CDDS. The following labeling algorithm was developed for this purpose.

Algorithm for Network Decomposition: First Level Labeling. The algorithm initially identifies the first decision vertices that can be reached from S, and then finds all possible paths from each decision vertex to "F." Each such vertex is the origin of a DDS, and elements of all paths emanating from it are elements of this specific DDS. A similar approach is used for CDDS. This is done as follows:

Step 1. First S receives the label "S." S is now labeled.

Step 2. Check the set $(IS)_S$--all immediate successors of S. If this set contains activity nodes m_i, they are labeled "S." If it contains decision vertices D_j, they are labeled i=1,2,...m consecutively. At the end of this step, all $(IS)_S$ is labeled.

Step 3. Repeat step 2 for each node m_k labeled "S" and its proper $(IS)_k$. If more decision vertices are encountered, they are labeled i=m+1,m+2,... consecutively.

Step 4. Labeling with "S" stops when:

a) no more "S" labeling is possible from an "S"-labeled node.

Or:

b) if (a) holds and "F" has been labeled "S."

Step 5. Go to the decision vertex D_j labeled i=1. Starting with this D_j, label all $j_k \ni D_j < j_k$ or $D_j << j_k$ with "1" (even if some of them carry a different label already). This process continues until:

a) no more labeling is possible, *and*

b) "F" is labeled "1."

Step 6. Repeat step 5 for the D_j labeled i=2,3,...

Step 7. From the set of nodes labeled "S" for every node m_k eliminate all nodes $m_i \notin (IS)_k^{(c)}$.

Thus, steps 1, 2, 3, 4, 7 generate the CDDS, if it exists; all nodes labeled "S" are elements of CDDS.

Steps 5 and 6 generate the various DDS; all nodes with the same label belong to the same DDS. Note that, as was mentioned before, some nodes may belong to more than one DDS.

The algorithm has been applied to the decision network of Fig. 17, yielding DDS_1, DDS_2, DDS_3, DDS_4 and CDDS. In Fig. 17, only DDS_1 and CDDS are shown (the rest are discussed in the example of Appendix B). The decision vertices associated with each DDS are:

$$\{D_1,D_5,D_6\} \in DDS_1$$
$$\{D_2,D_6\} \in DDS_2$$
$$\{D_3,D_2,D_6,D_7,D_8,D_9\} \in DDS_3$$
$$\{D_4,D_2,D_6\} \in DDS_4$$

DP Model of a DDS. Once the decision network has been decomposed into its DDS, each DDS is transformed into a DP model. As was indicated before, this requires the definition of stages, decision variables, state variables and returns. This is achieved through the following algorithm, called the *Second Level Labeling Algorithm.* This procedure has to be applied to each DDS separately. For the sake of clarity of presentation, it is shown for DDS_3 of the decision network of Fig. 17.

Step 1. Start with the first decision vertex $D_j = \{d_{jk(j)}\}$, i.e. $D_3 = \{d_{31}, d_{32}, d_{33}\}$. D_j is always the decision variable of the first stage of the DP model (stage n in a backward solution).

Step 2. Evaluate ES for this decision vertex.

Step 3. Select one of the above $d_{jk(j)}$, say d_{31}. Label jk(j) (i.e., 31) all $j_i \ni d_{jk(j)} < j_i$ or $d_{jk(j)} << j$ until:

a) another decision vertex or "F" is reached, or both, *and*

b) no more $m_i \ni d_{jk(j)} < m_i$ or $d_{jk(j)} << m_i$ can be labeled.

Step 4. If one or more decision vertices (or F) are reached, label them

2, jk(j) (i.e., 2,31)

where:

2 indicates that this decision vertex (decision set) is element of the second stage (stage n-1) of the DP model, and jk(j) indicates that $d_{jk(j)}$ (i.e., d_{31}) is an element of the state variable of this stage.

Step 5. Repeat step 4 for all $d_{jk(j)}$ of the decision vertex selected (i.e., for d_{32}, d_{33}).

Step 6. Repeat steps 4, 5 for *all* decision vertices labeled 2, jk(j) (i.e., 2, 31, 2, 32 and 2, 33), labeling the new decision vertices encountered 3, jk(j).

Step 7. Continue until all decision vertices are labeled.

Step 8. Using the critical path procedure, evaluate the length of time required by the longest path--denoted CP--from each $d_{jk(j)}$ associated with the ith stage to all $d_{jk(j)}$ associated with the (i+1) stage, provided there is at least one path between the two. This is done by considering for each $d_{jk(j)}$ only m_i that have the label jk(j). Note that some activity nodes may have more than one label, and should be considered accordingly. When evaluating CP for decision nodes of the first decision vertex, ES for each decision node is the one obtained in step 2. For all other cases, ES=0.

The last step of this algorithm evaluates $t_{jk(j)}$--the time return associated with decision node $d_{jk(j)}$. Thus, this algorithm yields the required elements of a DP model. (Note that during the application of this algorithm the stages are numbered forward. Once the stages are established, they are renumbered backward.)

This algorithm has been applied to DDS_3 of the decision network of Fig. 17, producing the reduction of DDS_3 to the decision network shown in Fig. 25.

Note the addition of another dummy decision vertex added after decision node 31. This has to do with the nonserial DP model introduced in the following section. The specific case of Fig. 25 is handled in Appendix B.

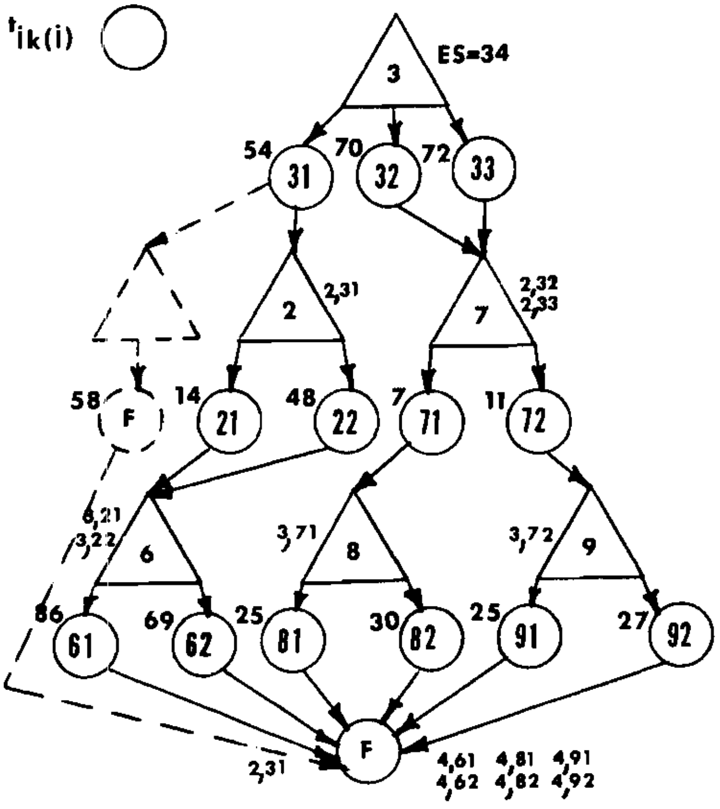


Figure 25. DDS_3

Nonserial DP Models. The DP model is not necessarily a serial model. Due to the method of constructing the DDS, and since each DDS is acyclic, only two other major types of DP models are possible, namely:

a) the diverging stages DP model, and

b) feed-forward loop DP model.

Two varieties of each one of these models are discussed in Appendix A, and they are: type I divergence, type II divergence, type I feedforward and type II feedforward. Later, type I divergence is introduced to show an example of the computational refinements required for nonserial stages.

An indication that a nonserial DP model exists is given whenever a decision vertex D_j has two or more labels of the same DDS, relating this D_j to a different stage, or whenever two different D_j have two or more labels, out of which at least two are exactly the same. This is amplified more in Appendix A.

Minimum Project Time. Consider the decision network of Fig. 17. The decomposed network can be schematically described as shown in Fig. 26.

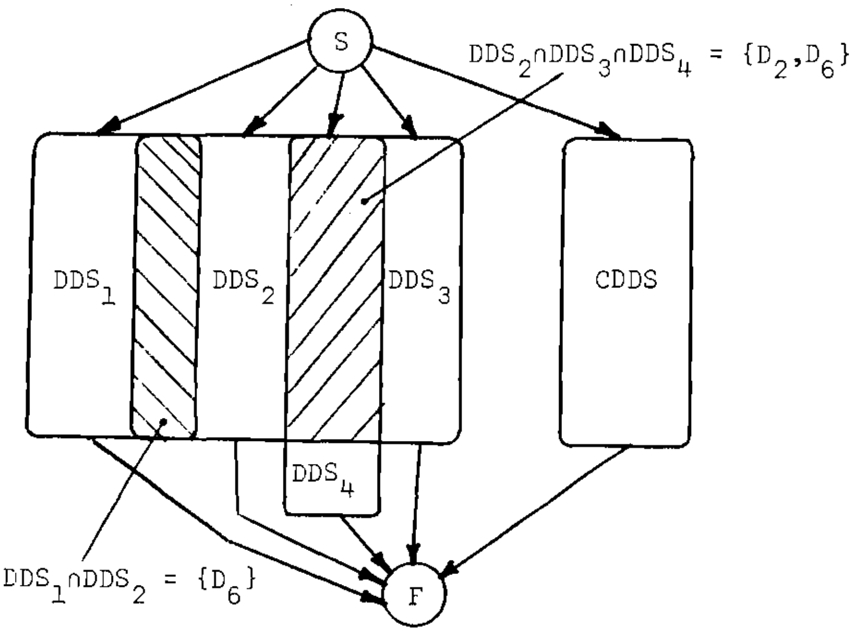


Figure 26. Decomposed Network

For convenience, "S" and "F" are shown separately, as before.

Let T^i denote the minimum time to get from S to F, associated with DDS_i. For CDDS, T^i is equal to the critical path from S to F, and is denoted by T^o as follows:

$$T^{o} = \underset{k}{\mathrm{Max}}\ L(Z_k)$$

For each DDS_i, T^i is the minimum critical path from "S" to "F," i.e.:

$$T^i = \underset{j}{\mathrm{Min}}(CP)^i_j = \underset{j}{\mathrm{Min}}[\underset{k}{\mathrm{Max}}\ L(Z_k)]^i_j$$

where $(CP)^i_j$ is the critical path of the jth standard network of DDS_i.

The optimal policy for DDS_i yielding T^i is denoted by

$$U^*_i = \{d^*_{jk(j)}\}_i$$

Inspecting Fig. 26, it is seen that all T^i are associated with paths from S to F. Therefore, the project minimum time is given by:

$$T^* = \underset{i}{\mathrm{Max}}(T^i) = \underset{i}{\mathrm{Max}}[\underset{j}{\mathrm{Min}}(CP)^i_j] = \underset{i}{\mathrm{Max}}\{\underset{j}{\mathrm{Min}}[\underset{k}{\mathrm{Max}}\ L(Z_k)]^i_j\}$$

And the optimal policy for the whole project is:

$$U^*_{\text{max. slack}} = \{U^*_i\} = \{d^*_{jk(j)}\}$$

The subscript "max.slack" indicates that this set of decision nodes will yield the maximum slack, but not necessarily the minimum cost of the project. Since T^* is a value of a specific DDS, it may be possible to select different set of decision nodes for some other

DDS, yielding a lower cost, without affecting the total project time. This is done later as part of the time-cost trade-off procedure.

Thus, once T^i is obtained for each DDS_i, it is simple to evaluate the project minimum time. To obtain T^i, the DP model of DDS_i has to be utilized, as described in the following section.

DP Solution. If the DP model describing the DDS is a serial stage model, the solution procedure described for the simple formulation is applicable here, with $t_{jk(j)}$ being the time return obtained in step 8 of the second level labeling algorithm. T^i is obtained from:

$$T^i = f^i_{n(t)}(X_n)$$

where $f^i_{n(t)}(X_n)$ is the minimum n stage time return obtained by solving the DP model of DDS_i. Since $X_n = \{S\}$, then:

$$T^i = f^i_{n(t)}(S)$$

To show the computational modifications required for a nonserial model, the case of type I divergence is described here, and other types of nonserial DP models are discussed in Appendix A.

Type I divergence is defined as follows: Suppose that there are three decision vertices D_1, D_2, D_3 such that

$$D_1 < D_2$$

$$D_1 < D_3$$

and

$$D_1 = \{d_{1k(1)}\}$$

Then, type I divergence (pure divergence) is the case when:

$$d_{1k(1)} < D_2 \; \forall \; k(1)$$

and

$$d_{1k(1)} < D_3 \; \forall \; k(1)$$

For example, consider the following decision network.

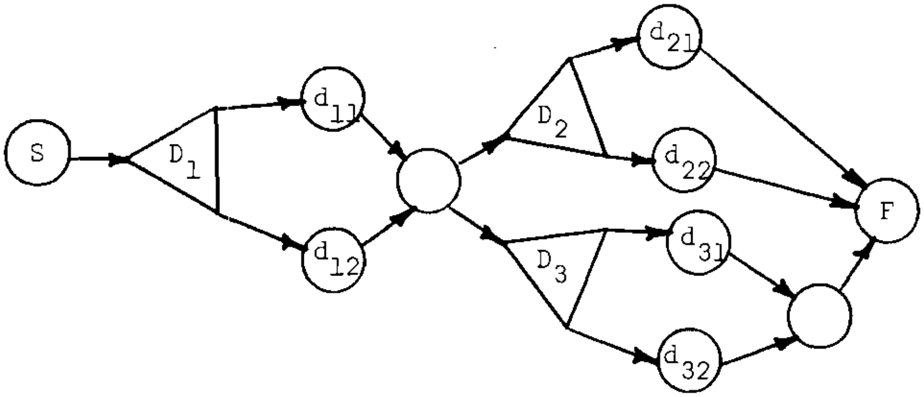


Figure 27. Type I Divergence

This is type I divergence, with one diverging branch, and the equivalent DP model is as shown in Fig. 28.

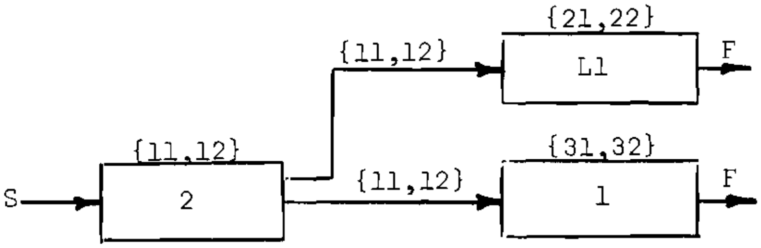


Figure 28. DP Model--Type I Divergence

Following the general procedure for a diverging branch system given in Nemhauser (51), let stage k be the divergence stage, let L1 represent the last stage of branch 1 (for a backwards approach and one branch type I divergence), and let:

$f_{(L1)(t)}(X_{k-1})$ - minimum L stage time return of branch 1.

$f_{(k+L1)(t)}(X_k)$ - minimum total time return at the diverging stage k.

$f_{(n+L1)(t)}(X_n)$ - minimum (n+L1) stage time return.

Since time values are not additive at a diverging stage, one gets:

$$f_{(k+L1)(t)}(X_k) = \underset{Y_k}{\text{Min}}\ Q_k(X_k,Y_k)$$

where:

$$Q_k(X_k,Y_k) = T_k(X_k,Y_k) + \text{Max}[f_{(L1)(t)}(X_{k-1}),f_{(k-1)(t)}(X_{k-1})]$$

and:

$$Y_k = X_{k-1}$$

The total minimum time for this DDS is:

$$T = f_{(n+L1)(t)}(X_n) = \underset{Y_n}{\text{Min}}[T_n(X_n,Y_n) + f_{(n-1+L1)(t)}(X_{n-1})]$$

Similar modifications are required for the other types of divergence. Diverging stages DDS can be optimized with no more effort than would be required for a serial stage model with equivalent number of stages.

Computational Refinements

Dominating Path. In some cases there might be a path bypassing a decision vertex, and longer than any path going through the decision vertex. An example of such a path is the one bypassing decision vertex 4 of Fig. 17. In cases like this, the decision node selected at this vertex will not affect the time value, and therefore the one yielding the smaller cost should be selected.

Common Decision Vertices. It was observed before (Fig. 22) that in some cases there is dependency between various DDS in the sense that $D_j \in DDS_i$ for some values of i; i.e., the same decision vertex can belong to more than one DDS. The question now is whether it is possible that while solving for DDS_i decision node d_{j1} will be selected, whereas solving for DDS_m decision node d_{js} will be selected, where:

$$\{d_{j1}, d_{js}\} \in D_j$$

The following theorem resolves this problem.

Theorem 2. If a certain decision vertex D_j is both $D_j \in DDS_i$ and $D_j \in DDS_m$, then either:

a) $d^*_{jk(j)} \in U^*_i$ and $d^*_{jk(j)} \in U^*_m$

or b) $d^*_{jk(j)} \in U^*_i$ and $d_{jk(j)} \notin U^*_m \forall k(j)$

or c) $d_{jk(j)} \notin U^*_i \forall k(j)$ and $d^*_{jk(j)} \in U^*_m$

or d) $d_{jk(j)} \notin U^*_i \forall k(j)$ and $d_{jk(j)} \notin U^*_m \forall k(j)$

Proof. Two cases have to be considered, as follows:

1) D_j is the first common decision vertex of DDS_i and DDS_m.

2) D_j is any other decision vertex succeeding the first one.

Case 1. Due to the method by which the DDS was constructed, all nodes succeeding D_j are elements of both DDS_i and DDS_m, i.e., let:

$$J_k = \{j_i | D_j < j_i \quad \text{or } D_j << j_i\}$$

then: $J_k \in DDS_i$ and $J_k \in DDS_m$.

Thus, because of the backwards solution procedure for the DP model, if D_j is an element of the decision variable of stage r of DDS_i, the same is true for DDS_m, i.e.

$$D_j \in Y_r \quad \text{for} \quad DDS_i$$

$$D_j \in Y_r \quad \text{for} \quad DDS_m$$

Let X_{r_i} be the input states for stage r of DDS_i, X_{r_m} the input states for stage r of DDS_m. Although $X_{r_i} \neq X_{r_m}$, the $d_{jk(j)}$ columns of $Q_r(X_{r_i}, Y_r)$ of DDS_i are equal to the $d_{jk(j)}$ columns of $Q_r(X_{r_m}, Y_r)$, because of the state transformation:

$$Y_r = X_{r-1}$$

Now, the r-stage minimum time return is:

$$f^{i}_{r(t)}(X_{r_i}) = \underset{Y_r}{\text{Min}}\; Q^{i}_{r}(X_{r_i},Y_r) = \text{Min}[\underset{Y'_r}{\text{Min}}(X_{r_i},Y'_r),\; \underset{D_j}{\text{Min}}(X_{r_i},D_j)]$$

where:

$$Y'_r \cap D_j = \emptyset \qquad Y'_r \cup D_j = Y_r$$

Also:

$$f^{m}_{r(t)}(X_{r_m}) = \underset{Y_r}{\text{Min}}\; Q^{m}_{r}(X_{r_m},Y_r) = \text{Min}[\underset{Y'_r}{\text{Min}}(X_{r_m},Y'_r),\; \underset{D_j}{\text{Min}}(X_{r_m},D_j)]$$

Therefore, the decision alternative selected for each input state is:

	$\underline{DDS_i}$		$\underline{DDS_m}$
(1)	$y^{*}_{rg} \neq d_{jk(j)} \;\forall k(j)$	and	$y^{*}_{rg} \neq d_{jk(j)} \;\forall k(j)$
or (2)	$y^{*}_{rg} = d^{*}_{jk(j)}$ with $k(j)^{*}$	and	$y^{*}_{rg} \neq d_{jk(j)} \;\forall k(j)$
or (3)	$y^{*}_{rg} \neq d_{jk(j)} \;\forall k(j)$	and	$y^{*}_{rg} = d^{*}_{jk(j)}$ with $k(j)^{*}$
or (4)	$y^{*}_{rg} = d^{*}_{jk(j)}$ with $k(j)^{*}$	and	$y^{*}_{rg} = d^{*}_{jk(j)}$ with $k(j)^{*}$

If $d_{jk(j)} \neq y^{*}_{rg}$ it can never become an element of U^{*}_{i} or U^{*}_{m}. If $d_{jk(j)} = y^{*}_{rg}$ it may or may not become an element of U^{*}_{i} or U^{*}_{m}. Since the only possible outcomes at stage r are the ones mentioned above, it is clear that only (a), (b), (c), or (d) of the theorem can happen.

Case 2. The proof of this case is essentially the same as that of the previous one. This proves the theorem.

The Minimum Time Network

Once $U^*_{max.slack}$ has been established, the decision network can be reduced to a standard network by following the procedure described in Chapter III. All nodes of the standard network are regarded now as activity nodes, and the cost associated with this network is obtained by adding the cost of each node. As mentioned before, this is *not* the minimum cost possible for the minimum time solution. This minimum cost is evaluated later.

This concludes the discussion of the minimum time problem. The procedure developed here has been applied to the decision network of Fig. 17 in Appendix B.

Minimum Cost Problem

The minimum cost problem is defined as follows: Given the decision network G(J,A), select for each decision vertex (decision set) D_j at most one decision node $d^*_{jk(j)} \epsilon D_j$, so that the total project cost will be minimized. Find the time associated with the minimum cost.

The approach taken for solving this problem is basically the same one used for the minimum time problem, i.e. transforming the decision network into a DP model. Some modifications of the previous procedure have to be introduced to accommodate the fact that one deals here with cost returns vs. time returns before.

Specifically, the solution procedure for this case involves three steps, as follows:

1) DP model for each DDS

2) DDS Minimum Cost

3) Project Minimum Cost and Time.

DP Model

Constructing the DP Model for this case is done in three steps: decomposing the network, establishing the reduced cost network, and evaluating the cost returns.

Network Decomposition. Decomposing the decision network into DDS is done using the same procedure outlined for the minimum time problem. Thus, if the minimum time problem is solved first, the network decomposition is already available.

Reduced Cost Network (RCN). Before establishing the notion of the reduced cost network and discussing its use, the notion of the permanent nodes has to be introduced.

Definition. The set N of permanent nodes is the set of all nodes $m_i \in M$ (i.e., $N \subseteq M$) such that:

$$N \subseteq J^* \ \forall \ d_{jk(j)}$$

Thus, permanent nodes are all nodes that their inclusion in the standard network is not affected by the selection of V^*--where V^* is the policy of the minimum cost solution.

The set N can be easily identified during the second level labeling as follows:

1) All nodes labeled "S" are permanent nodes.

2) During the second level labeling, if a node m is such that $D_j < m_i$, and if this node is labeled with *all* node numbers of $d_{jk(j)} \in D_j$, then this node is permanent if $m_k << D_j$ is permanent.

The concept of reduced cost network can be introduced now.

Definition. Reduced Cost Network (RCN) is the network obtained after reducing to zero the cost of each permanent node.

The concept of RCN is essential for the minimum cost solution, as can be seen from the following example:

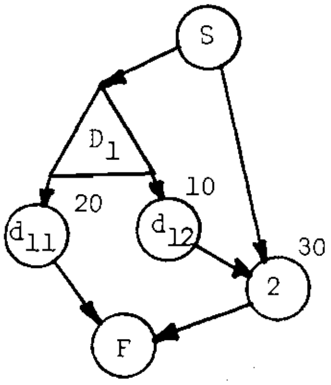


Figure 29. Cost Decision Network

Activity node 2 is obviously a permanent node. Now, to evaluate the minimum cost network, one evaluates the cost associated with each decision node. This is:

$$c_{11} = 20$$

$$c_{12} = 10 + 30 = 40$$

Thus, for minimum cost one would select d_{11} and eliminate d_{12}. But the corresponding cost of the standard network is 20 + 30 = 50, and this is *not* the minimum, which is 40 and can be obtained by selecting d_{12}.

Obviously, solving with RCN (i.e., reducing the cost of node 2 to 0) will eliminate this problem. The reason is that the cost associated with a certain $d_{jk(j)}$ should be the incremental cost over the "fixed" cost which is not affected by the selection of a particular decision node.[4]

Evaluating Cost Returns. The stage returns can be evaluated by a modified version of the second level labeling algorithm of the minimum time problem (p. 67). The modified algorithm is:

Steps 1-7. As before, with the *elimination* of step 2 and where the DDS is composed of elements of RCN.

Step 8. If an activity node has two or more labels jk(j) associated with decision nodes of different decision vertices, say D_j and D_m, the label associated with the $d_{jk(j)}$ connected with a smaller stage number is dominating, provided $D_j < D_m$ or $D_m < D_j$. The rest of the labels should be eliminated. If the labels are associated with $d_{jk(j)}$ of the same D_j, the same stage number, or if $D_j \nprec D_m$ or $D_m \nprec D_j$, they all remain.[5]

The cost return associated with each $d_{jk(j)}$ is then:

---

[4] The problem does not exist for the minimum time problem, because time values are not strictly additive. Thus, if in Fig. 29 the values represent time, one would select d_{11} yielding a total time of 30 vs. 40 if d_{12} is selected.

[5] The symbol $\nprec$ means "does not precede."

$$c_{jk(j)} = \sum_{\substack{\text{all nodes}\\ \text{labeled}\\ jk(j)}} c_i$$

where c_i is the cost of activity node m_i, and $c_{jk(j)}$ also includes the initial cost of decision node $d_{jk(j)}$.

The idea behind step 8 above is the same as that of the RCN. If a certain cost has been committed at an early stage, later on only the incremental addition should be considered. The use of RCN eliminates the need of adding a dummy decision node as in Fig. 25, if a backwards solution approach is used.

Application of this algorithm yields the required elements of a DP model.

DDS Minimum Cost

The solution procedure for the cost problem is exactly the same as that for the time problem. The DP model is equivalent, and the only difference is that instead of time returns one deals with cost returns. Let, for DDS_m:

$C_i(X_i,Y_i)$ be the cost matrix for stage i.

$R_i(X_i,Y_i)$ be the i stage cost return matrix.

$f_{i(c)}(X_i)$ be the minimum i stage cost return.

C^m be the minimum cost of DDS_m.

Then:

$$C^m = f^m_{n(c)}(X_n) = \underset{Y_n}{\text{Min}}\ R_n(X_n,Y_n)$$

where:

$$R_n(X_n,Y_n) = C_n(X_n,Y_n) + f_{(n-1)(c)}(X_{n-1})$$

and $f^m_{n(c)}$ is the total minimum cost return of DDS_m.

In contrast to time values, cost values are always additive. Therefore, the solution of nonserial stage models follows the standard procedure for these cases (see Nemhauser (51)). Thus, the procedure for type I divergence described previously would be in this case:

$$f_{(k+L1)(c)}(X_k) = \underset{Y_k}{\text{Min}}\ R_k(X_k,Y_k)$$

where:

$$R_k(X_k,Y_k) = C_k(X_k,Y_k) + f_{(k-1)(c)}(X_{k-1}) + f_{(L1)(c)}(X_{k-1})$$

and:

$$C^m = f^m_{(n+L1)(c)} = \underset{Y_n}{\text{Min}}[C_n(X_n,Y_n) + f_{(n-1+L1)(c)}(X_{n-1})]$$

Other types of nonserial models are discussed in Appendix A.

Project Minimum Cost

Once the minimum cost solution of each DDS has been obtained, the project minimum cost can be evaluated. This is done as follows:

Let:

C^i - minimum cost of DDS_i.

V_i^* - optimum policy for DDS_i yielding C^i.

Then, the optimal policy for the project minimum cost is:

$$V^* = \{V_i^*\} = \{d^*_{jk(j)}\}$$

Note that the property discussed in Theorem 2 is valid also here.

The optimal policy V^* enables the reduction of the decision network to the standard network $G(J^*,A^*)$ for the minimum cost solution. Then, the project minimum cost is given by:

$$C^* = \sum_{m_i \epsilon J^*} C_{i_*} + C_{PN}$$

where C_{PN} is the total cost of the permanent nodes.

The value of T_{C^*}--the time associated with the minimum cost solution--is found by evaluating the critical path of the standard network $G(J^*,A^*)$.

Computational Refinements. Some refinements of the minimum cost procedure are required for DDS's having common decision vertices. The problem is similar to the one discussed while introducing the RCN, and can be best described through the example of Fig. 30.

Considering the costs represented by the upper case numbers and solving for the minimum cost for each DDS, the outcome is $V^* = \{12,22,31\}$

with a total minimum cost of 240. However, the minimum cost of this network is 210, with $V^* = \{12,31,21\}$.

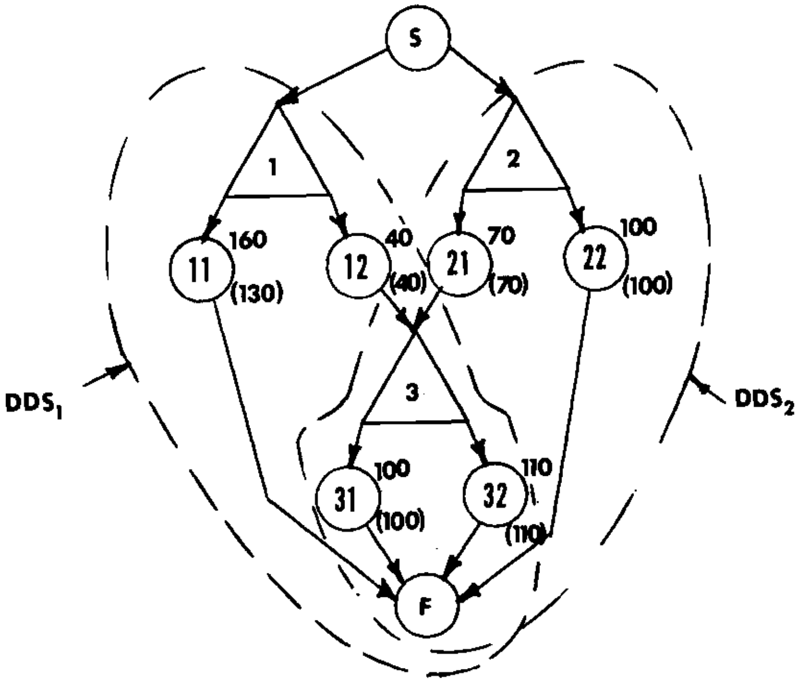


Figure 30. Cost Example

The other difficulty is represented by the lower case numbers (in parenthesis). Solving for this case yields $V^* = \{11,22\}$ with a minimum cost of 230, whereas the true minimum cost is 210 with $V^* = \{12,21,31\}$. This leads to the following modification, which is, in a sense, an incremental analysis. There are three cases to be considered, as follows:

Case 1. Let D_m represent the *first* common decision vertex for some DDS, i.e., $D_m \in DDS_i$, i=1,2,... Then, if the outcome of the minimum cost solution is such that all[6] $d_{jk(j)} << D_m$ of different DDS_i are included in V^*, no further action is required.

Referring to Fig. 30, this means that if $\{12,21\} \epsilon V^*$ by solving separately for each DDS, then $\{11,22\}$ cannot yield a lower value for the cost.

Case 2. Let D_m be as before, and suppose that at least one $d_{jk(j)} << D_m$ is not an element of V^*. Let this $d_{jk(j)}$ be associated with DDS_i, i.e. $(d_{jk(j)})_i$. Let D_m be associated with the decision variable of stage r of the DP model of DDS_i, i.e.,

$$D_m \epsilon Y_r$$

or

$$\{d_{mk(m)}\} \epsilon Y_r$$

As mentioned in Theorem 2, $D_m \epsilon Y_r$ for all other DDS's considered.

Since one $d_{mk(m)}$, namely $d^*_{mk(m)}$ was already committed for some other DDS, it is now a problem of checking only the incremental addition to the total cost contributed by the minimum cost path starting with $(d_{jk(j)})_i$ and leading to the first decision vertex of $(DDS)_i$. This is done as follows:

First, $Q_r(X_{r_i},Y_r)$ is modified to give:

$$Q_r(X_{r_i},Y_r)' = Q_r(X_{r_i},Y_r) - Q_r(X_{r_i},d^*_{mk(m)})$$

---

[6] $d_{jk(j)} << D_m$ means here the decision nodes that precede D_m, so that a path from $d_{jk(j)}$ to D_m includes only activity nodes m_i.

Note that $Q_r(X_{r_i}, d^*_{mk(m)})$ has only one value for each input state. Once this is done, the DP solution proceeds as usual from stage (r+1) to n. This may yield a different V^*_i than was obtained initially. If this happens, V^* is adjusted accordingly.

Case 3. Let D_m be defined as before, and suppose none of the decision nodes immediately preceding D_m (in the sense defined before) is initially selected for V^*. Then, for similar reasons as in case 2, in stage r the following r-stage return matrix is formed.

$$Q_r(X_r,Y_r)' = Q_r(X_r,Y_r) - \underset{D_m}{\text{Min}}(X_r,D_m)$$

The DP routine is reapplied now to stages (r+1) through n for each DDS. If a new policy V^* is obtained, its cost is evaluated, and if it is less than the cost obtained before, this is the minimum cost policy.

Applying this rule to the decision network of Fig. 30 amounts to having:

$$c_{31} = 0 \qquad c_{32} = 10$$

and thus d_{31} is selected for stage 1. This will yield a selection of d_{12} for DDS_1 $(c_{12}<c_{11})$, and d_{21} for DDS_2 $(c_{21}<c_{22})$, yielding $V^* = \{12,21,31\}$ and C = 210 as before.

Note that the three cases discussed here do not create any difficulty for the minimum time problem, because time values are not strictly additive.

This concludes the discussion of the minimum cost problem. The procedure developed here has been applied to the decision network of Fig. 17 in Appendix B.

Time Cost Trade-Off

The two procedures described previously yield the minimum project time and the minimum project cost with its associated time. The purpose of this section is to develop a procedure for finding the "Efficient Set," which is the collection of all admissible points, defined as follows. Let:

$0 = \{o_j\}$ = set of all possible outcomes, i.e.

$$0 = \{o_j | o_j = (T,C) \text{ where } T,C \in \text{Reals and } T \geq 0,\ C \geq 0\}.$$

o_j^* - an admissible point.

$0^* = \{o_j^*\}$ = the efficient set.

$$0^* = \left\{ \begin{array}{l} o_j^* | o_j^* = (T_i,C_i) \text{ where } T_i,\ C_i \in \text{Reals},\ T_i \geq 0,\ C_i \geq 0, \\ \qquad T_i \text{ are arranged in ascending order of } T_i \text{ values}, \\ \qquad \text{and if } T_i > T_{i-1},\ C_i < C_e \ \forall\ e=1\ldots i-1. \end{array} \right\}$$

where

T_i - the ith possible project time value.

$C_i = \underset{m}{\text{Min}}\{C_{T_i^m}\}$, and

T_i^m = the ith possible project time value obtained by selecting the mth subset of decision nodes. Note that for fixed i, T_i^m is constant for all m.

$C_{T_i^m}$ - project cost associated with T_i^m.

Obviously, $0^{*} \subset 0$ and it is the lower boundary of 0.

The point $(T_{C*}, C*)$ obtained before is an admissible point, as well as the point $(T*, C_{T*})$, where only T* has been evaluated so far. These two points define a closed interval[7] in E^2 so that any point outside this interval is of no interest.

In order to obtain 0^{*} for the problem at hand, first the general methodology for finding the efficient set is developed, and then it is applied to decision networks.

A Methodology for Finding the Efficient Set

Consider a serial dynamic programming model, where the decision variable of one stage is the state variable of the following one, and each input state and decision alternative has two discrete returns associated with it, say time and cost. This model can be described as illustrated in Fig. 31 (assuming the backward solution approach is to be used):

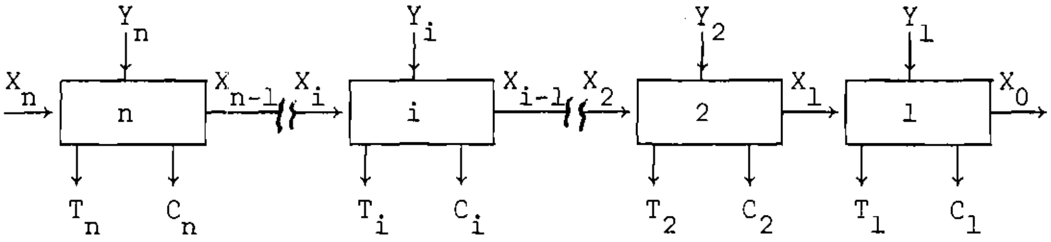


Figure 31. A Serial DP Model

---

[7] A closed interval in E^2 is defined as the set of points $\{(x_1, x_2) \in E^2 | a_i \leq X_i \leq b_i, \forall i=1,2\}$.

The recursive backward solution procedure described before treats the above model as two separate problems--one with time returns and one with cost returns, yielding

$$T^* = f_{n(t)}(X_n) \text{ with } U^* \text{ - the minimum time policy}$$

$$C^* = f_{n(c)}(X_n) \text{ with } V^* \text{ - the minimum cost policy.}$$

Note that for both problems $g_i(X_i,Y_i)$ is the same, i.e.:

$$X_{i-1} = g_i(X_i,Y_i) = Y_i$$

and X_n is an element set.

The value of C_{T*} can be easily obtained in this case by summing all values of $C_i(X_i,Y_i)$ associated with U^*. Similarly, the value of T_{C*} is obtained by summing all values of $T_i(X_i,Y_i)$ associated with V^*. Thus, the two extreme points of the efficient set O^* are obtained by solving two separate DP problems yielding:

$$o_1^* = (T^*,C_{T*})$$

$$o_N^* = (T_{C*},C*)$$

if the "Efficient Set" has N points.

One way to find all other $o_j^* \in O^*$ is by complete enumeration of all possible outcomes, i.e., all $o_j \in O$. This method gets very fast out of hand. For the example shown in Appendix E, this would mean evaluating 4096 elements of O, whereas O^* has only 8 elements.

The method introduced here reduces substantially the computational effort required. The method can be regarded as a *sensitivity analysis* or *incremental analysis* of a DP problem of the type shown above. It can be applied in two ways: starting at o_1^* and working towards o_N^*, or vice versa. This procedure is introduced first as an algorithm, followed by a discussion of optimality.

Efficient Set Algorithm. The main idea behind this algorithm is the utilization of the information obtained in the minimum cost or minimum time solutions to generate a set of "promising" points, where the efficient set is contained in this set. This is done by utilizing the matrices of the DP solution to perform incremental analysis and find what the best ways to move away from the optimum are, so as to remain on the lower boundary O^*. Thus, the algorithm can be applied in two ways: starting with o_1^* progressing to o_N^* utilizing the DP matrices of the minimum time solution, or starting with o_N^* progressing to o_1^* utilizing the DP matrices of the minimum cost solution. Following is a description of the first approach, to be referred to as the "Efficient Set Algorithm--Time Version." The nature of the changes required for the cost version will be introduced later. The "Time Version" is as follows:

Step 1. Solve the minimum cost problem, obtaining (T_{C^*},C^*), and V^*.

Step 2. Solve the minimum time problem, obtaining (T^*, C_{T^*}) and U^*. As a consequence of this procedure, for each stage i, the i stage return matrix is available, having the following elements:

$$Q_i(X_i, Y_i) = (q_{ijk})$$

Step 3. Construct, for each stage i, the i stage *cost matrix of the minimum time solution* as follows:

Let Y_i^* represent the optimum set of decision alternatives selected at stage i *given each* input state of X_i, for the minimum time solution. Denote this by:

$$Y_i^* = \{y_{ik}^* | x_{ij}\}$$

i.e.: the element y_{ik}^* selected given input x_{ij}. Thus, for example, if

$$X_i = \{x_{ij}\} \quad j=1...4$$

$$Y_i = \{y_{ik}\} \quad k=1...3$$

The decision set Y_i^* may look like: $Y_i^* = \{(y_{i3}|x_{i1}), (y_{i1}|x_{i2}), (y_{i2}|x_{i3}), (y_{i3}|x_{i4})\}$.

Now, $\Theta_i(X_i, Y_i)$ - the i stage cost matrix of the minimum time solution is defined as follows:

$$\Theta_1(X_1,Y_1) = C_1(X_1,Y_1) = (\theta_{1jk}) = (c_{1jk})$$

$$\Theta_i(X_i,Y_i) = C_i(X_i,Y_i) + F_{i-1}(X_{i-1}) \quad i=2\ldots n$$

where:

$$F_{i-1}(X_{i-1}) = \Theta_{i-1}(X_{i-1},Y^*_{i-1})$$

Since

$$X_{i-1} = Y_i$$

Then:

$$\Theta_i(X_i,Y_i) = C_i(X_i,Y_i) + F_{i-1}(Y_i) = (\theta_{ijk}) \quad i=2\ldots n$$

Step 4. Construct $\Delta Q_i(X_i,Y_i)$--the *time increment matrix* for stage i, as follows:

$$\Delta Q_i(X_i,Y_i) = (q_{ijk}-\min_k q_{ijk}) = (q_{ijk}-q_{ijk'}) = (\Delta q_{ijk}) \quad i=1\ldots n-1$$

Note that $\Delta q_{ijk} \geq 0 \;\forall\; i,j,k$.

Step 5. Construct $\Delta\Theta_i(X_i,Y_i)$--the *cost increment matrix* of the minimum time solution, for stage i, as follows:

$$\Delta\Theta_i(X_i,Y_i) = (\theta_{ijk}-\theta_{ijk'}) = (\Delta\theta_{ijk}) \quad i=1\ldots n-1$$

where k' is the one obtained for $\Delta Q_i(X_i,Y_i)$.

Note that $\Delta\theta_{ijk} \gtrless o \forall i,j,k$.

Step 6. Eliminate in $Q_n(X_n,Y_n)$, $\Theta_n(X_n,Y_n)$ all entries which correspond to $y_{n\bar{j}} \in Y_n$, where $y_{n\bar{j}}$ are all elements for which

$$Q_n(X_n,Y_n) > T_{C*}$$

(Recall that $X_n = \{x_{n1}\}$--i.e. an element set, and:

$$T^* = \underset{Y_n}{\mathrm{Min}}[Q_n(X_n,Y_n)])$$

Eliminate in $\Delta Q_{(n-1)}(X_{n-1},Y_{n-1})$, $\Delta\Theta_{n-1}(X_{n-1},Y_{n-1})$ all rows corresponding to $\{x_{(n-1)\bar{j}}\}$.

(Recall that $X_{n-1} = Y_n$.)

Step 7. Construct:

$\Omega_i^m(X_i,Y_i)$ - the *time change matrix* for *stage i* when the procedure starts at stage m.

$\Phi_i^m(X_i,Y_i)$ - *the cost change matrix for* stage i, when the procedure starts at stage m. Initially m = 1.

These matrices are defined as follows:

$$\Omega_m^m(X_m,Y_m) = \Delta Q_m(X_m,Y_m) \qquad i=m$$

$$\Omega_i^m(X_i,Y_i) = \Delta Q_i^m(X_i,Y_i) + \Omega_{i-1}^m(X_{i-1},Y_{i-1})^* \qquad m<i\leq n-1$$

$\Omega_i^m(X_i,Y_i) = (\omega_{ijk_r}^m)$ where the subscript r on k indicates that $\omega_{ijk_r}^m$ may have more than one element in each location (jk) at stage i.

$\Omega_i^m(X_i,Y_i)^* = (\omega_{ijk_r^*}^m)$

$$\Phi_m^m(X_m,Y_m) = \Delta\Theta_m(X_m,Y_m) \qquad i=m$$

$$\Phi_i^m(X_i,Y_i) = \Delta\Theta_i(X_i,Y_i) + \Phi_{i-1}^m(X_{i-1},Y_{i-1})^* \qquad m\le i\le n-1$$

$$\Phi_i^m(X_i,Y_i) = (\phi_{ijk_r}^m)$$

$$\Phi_i^m(X_i,Y_i)^* = (\phi_{ijk_r^*}^m)$$

The starred matrices are formed using the following procedure.

(a) Any element that is starred or eliminated for $\Omega_i^m(X_i,Y_i)$, the corresponding element in $\Phi_i^m(X_i,Y_i)$ is starred or eliminated too, respectively, and vice versa.

(b) Eliminate all elements for which:

(1) $\omega_{ijk_r}^m > (T_{C^*}-T^*)$

(2) $\phi_{ijk_r}^m > 0$

(3) $\left.\begin{matrix}\Delta\theta_{(n-1)jk}\\ \Delta q_{(n-1)jk}\end{matrix}\right\}$ were eliminated in Step 6.

$$(4)\quad \left.\begin{array}{ll} \text{If,} & \omega^{m}_{ijk_r} \geq \omega^{a}_{ijk^*_r} \\ & \\ \text{and:} & \phi^{m}_{ijk_r} \geq \phi^{a}_{ijk^*_r} \end{array}\right\} \begin{array}{l} \forall\ a = 1 \ldots m-1 \\ \forall\ k^*_r \end{array}$$

(c) The first starred element for each row is: (provided there is at least one element which has not been eliminated or is not equal to zero)

$$\omega_{ijk^*_r} = \underset{k_r}{\mathrm{Min}}\ \omega_{ijk_r}$$

in case of more than one value for $\omega_{ijk^*_r}$, the one with the smallest corresponding $\phi_{ijk^*_r}$ is starred, and the rest are eliminated.

(d) Additional starred elements are all elements for which:

$$\left.\begin{array}{ll} \text{if:} & \omega^{m}_{ijk_r} \geq \omega^{m}_{ijl_s} \\ & \\ \text{then:} & \phi^{m}_{ijk_r} < \phi^{m}_{ijl_s} \end{array}\right\} \forall\ l_s, \quad \begin{array}{l} \text{where } l_s \text{ includes} \\ \text{also all values} \\ \text{(except one) of } k_r. \end{array}$$

The above procedure has to be performed in the sequence described. If, as a consequence of (b), all elements of $\Omega^m_i(X_i,Y_i)$ and $\Phi^m_i(X_i,Y_i)$ are either eliminated or zero, then the next step is step 9.

Step 8. Evaluate the set of "promising points" (T_p,C_p) as follows:

$$\left.\begin{aligned} T_{pj_r} &= q_{n1j} + \omega^{m}_{(n-1)jk^*_r} \\ C_{pj_r} &= \theta_{n1j} + \phi^{m}_{(n-1)jk^*_r} \end{aligned}\right\} \forall\ j,\ k^*_r$$

The subscript j indicates that the specific values of T_p or C_p are obtained from the value in the jth column of $Q_n(X_n,Y_n)$ or $\Theta_n(X_n,Y_n)$. The subscript r on j indicates that there might be more than one value in column j.

Note that $Q_n(X_n,Y_n)$ and $\Theta_n(X_n,Y_n)$ are also elements of (T_p,C_p).

Step 9. Repeat steps 7, 8 for m = 2,3,...n-1.

Step 10. Initial Elimination. Given the set of promising points (T_p,C_p), then:

(a) Eliminate all pairs for which:

$$T_{pj_r} > T_{C^*}$$

or:

$$C_{pj_r} > C_{T^*}$$

(b) Eliminate all entries (T_{pm_r},C_{pm_r}) for which:

$$\left.\begin{aligned} &\text{If:} & T_{pm_r} &= T_{pj_r} \\ &\text{then:} & C_{pm_r} &> C_{pj_r} \end{aligned}\right\} \forall\ m_r,\ j_r$$

Step 11. Admissible Point Test. Arrange (T_{pj_r}, C_{pj_r}) in ascending order of T_{pj_r}. Call these pairs now (T_e, C_e), where the index e increases with increasing values of T_e, i.e.

$$T_1 = T^*$$

$$T_M = T_{C*} \quad \text{if} \quad 1 \le e \le M$$

The efficient set is composed of all points $o_j^* = (T_e^*, C_e^*)$ satisfying the following conditions simultaneously:

$$T_e > T_{e-1}$$

$$C_e < C_i \quad \forall \quad 1 \le i \le e-1$$

The policy W_e associated with each o_j^* is found by tracing back y_{ik^*} according to $\omega^m_{ijk^*_r}$ or $\phi^m_{ijk^*_r}$ in the matrices $\Omega^m_i(X_i, Y_i)^*$ or $\Phi^m_i(X_i, Y_i)^*$.

The above algorithm is recapped in the flow chart of Fig. 32.

Optimality of the Algorithm. To show the optimality of this procedure, first the general concept is discussed in more detail, and then each step is discussed separately. While doing so, it will be helpful to refer to the following graphical description of the efficient set.

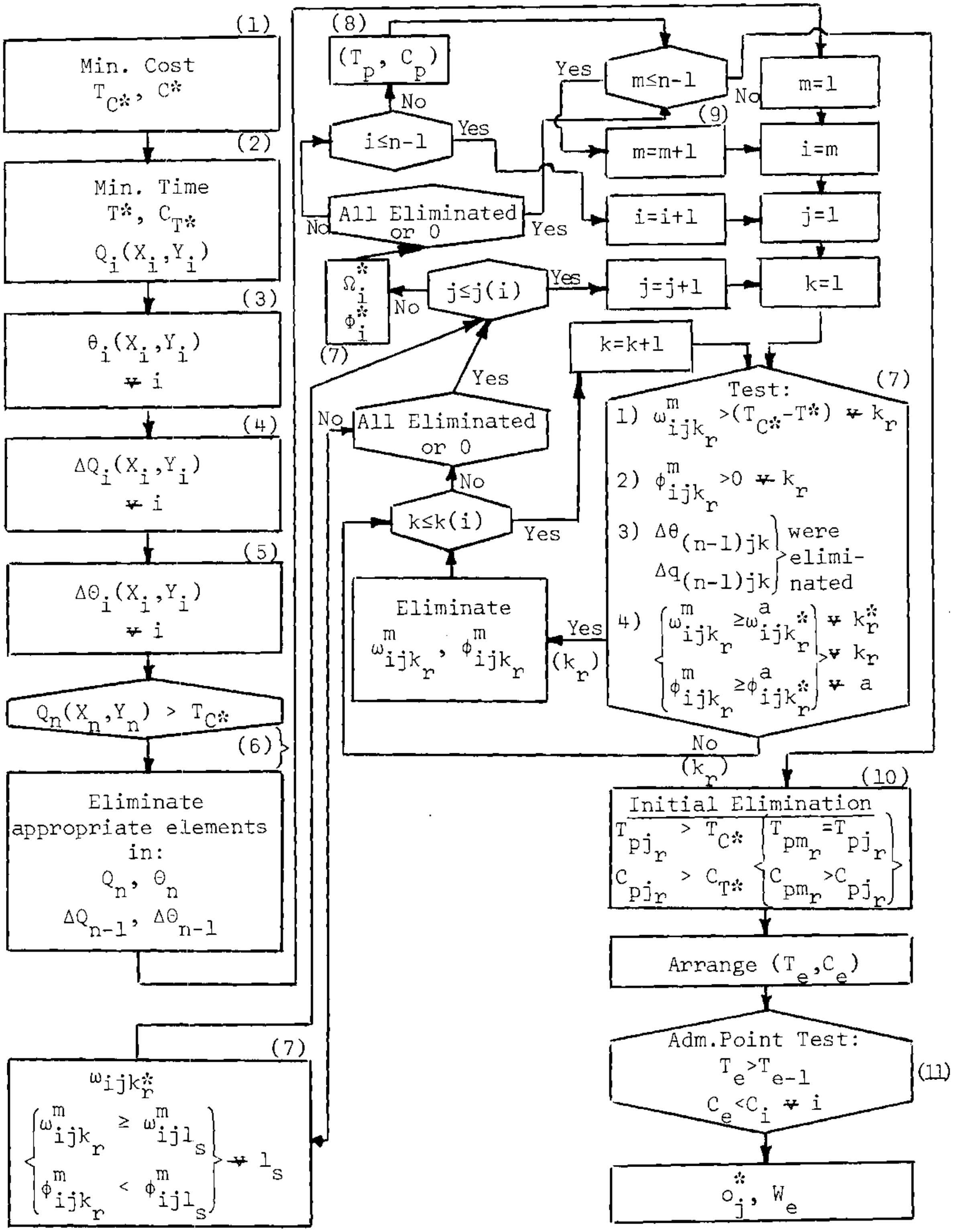


Figure 32. Efficient Set Algorithm--Time Version: Flow Chart

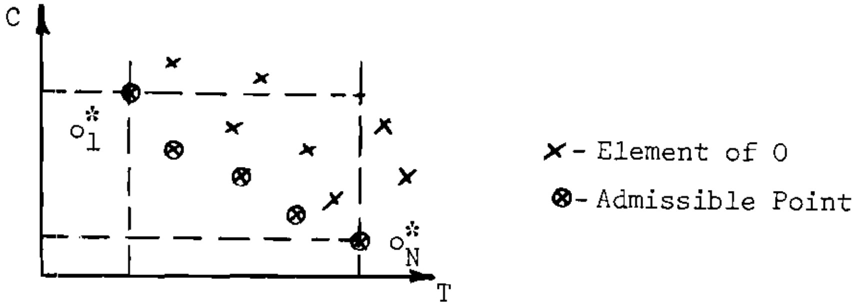


Figure 33. The Efficient Set

Obviously, the minimum time solution and minimum cost solution yield o_1^* and o_N^*, respectively. As mentioned before, any point falling outside the closed interval in E^2 is definitely of no interest, whereas points within E^2 may or may not be elements of the efficient set.

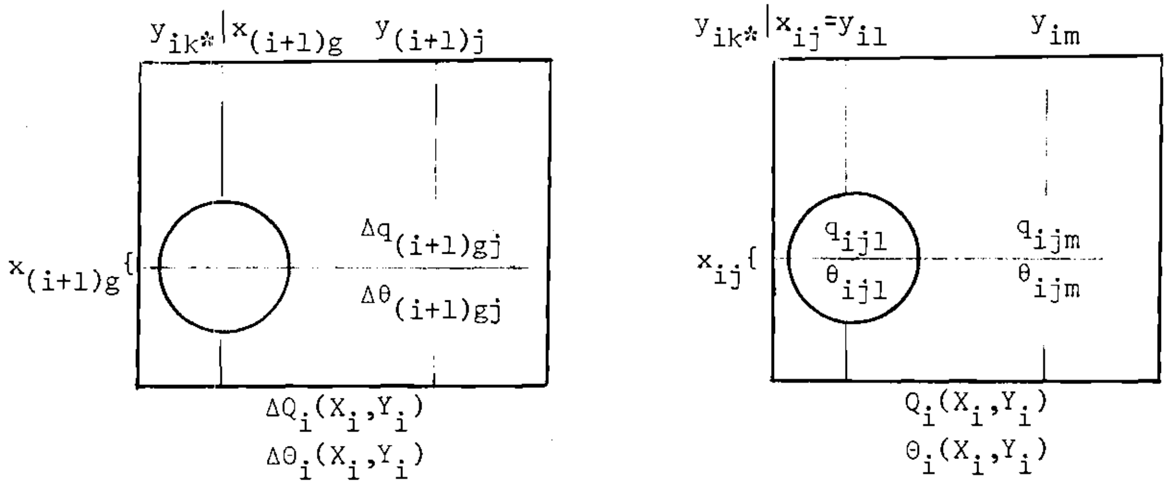
Due to the nature of the DP solution, the values of $Q_n(X_n,Y_n) = Q_n(x_{n1},Y_n)$ represent the minimum possible time value associated with each y_{nk} and were obtained by selecting the minimum value for each x_{ij} and all y_{ik} up to i=n. Furthermore, the values of $Q_i(X_i,Y_i)$ obtained for each stage i represent the *minimum* i-stage time for each x_{ij} and y_{ik}.

Due to the way $\Theta_i(X_i,Y_i)$ is constructed, its values represent the i-stage cost for each x_{ij} and y_{ik} if the i-stage minimum time policy is followed. Note that $\Theta_i(X_i,Y_i)$ is not the i-stage *minimum cost* matrix, which is given by $R_i(X_i,Y_i)$. Accordingly, $\Theta_n(X_n,Y_n)$ yields the costs of the minimum possible times associated with each y_{nk}.

Consider now stage i. Let:

$$y_{i1} = y_{ik^*} | x_{ij}$$

i.e., the optimal decision associated with input state x_{ij}, and let y_{im} be some other decision alternative. The elements of $Q_i(X_i,Y_i)$ and $\Theta_i(X_i,Y_i)$ associated with the above would be as illustrated in Fig. 34(a).



(b) (a)

Figure 34. DP Matrices

Suppose that, instead of making the optimal decision y_{i1}, at stage i, the decision y_{im} is selected. If all $y_{ik}^*|x_{ij}$ for $(i+1) \le i \le n$ do not change, then one of two things may happen:

1) $Q_n(X_n,Y_n)$, $\Theta_n(X_n,Y_n)$ remain unchanged.

2) One or more pairs of values of $Q_n(X_n,Y_n)$, $\Theta_n(X_n,Y_n)$ will change.

If (1) happens, then moving away from the optimum decision at stage i will not affect the final result. The change associated with (2) will be:

The change in time:

$$\Delta q_{ijm} = q_{ijm} - q_{ijl} \geq 0$$

since $q_{ijl} \leq q_{ijm}$; and the change in the associated cost:

$$\Delta\theta_{ijm} = \theta_{ijm} - \theta_{ijl} \overset{>}{<} 0$$

The time values will never decrease, since the values of $Q_n(X_n,Y_n)$ are the minimum possible and $\Delta q \geq 0$, whereas the cost values may increase, decrease or remain the same, depending on the value of $\Delta\theta$. Of course, if *just the above change was made,* and $\Delta\theta > 0$, none of the new values obtained for (T_p,C_p) could be an admissible point.

Refer again to stage i. Suppose that instead of moving away from the optimal y_{il} to y_{im}, one moves to y_{is}, for which

$$\Delta q_{ijs} = q_{ijs} - q_{ijl} < \Delta q_{ijm}$$

$$\Delta\theta_{ijs} = \theta_{ijs} - \theta_{ijl} < \Delta\theta_{ijm} < 0$$

Obviously, under these conditions, moving away to y_{im} can never yield an admissible point, since moving to y_{is} gives a higher decrease in cost for a smaller increase in time.

If, on the other hand,

$$\Delta q_{ijs} < \Delta q_{ijm}$$

$$\Delta\theta_{ijs} > \Delta\theta_{ijm} \text{ and } \Delta\theta_{ijs}, \Delta\theta_{ijm} < 0$$

Then both moves *may* yield admissible points. The above procedure is referred to as the "promising change procedure."

If the procedure starts at stage i and $\Delta\theta_{ij\cdot} \geq 0$, then moving away from the optimal decision y_{i1} will never yield an admissible point.

Now suppose that a move away from the optimum to y_{im} is made at stage i, and a move away from the optimum to $y_{(i+1)j}$ (which is equal to x_{ij}) for $x_{(i+1)g}$ is made at stage (i+1), and from there on all $y_{ik*}|x_{ij}$ for $(i+2) \leq i \leq n$ do not change. (See Figure 34(b)). As before, if there is a change in one or more (T,C) pairs in stage n it will be

time change: $\Delta q_{ijm} + \Delta q_{(i+1)gj} > 0$

cost change: $\Delta\theta_{ijm} + \Delta\theta_{(i+1)gj}$

Again, only if the total change is negative, this may be an admissible point(s). Note that if $\Delta\theta_{ijm} > 0$, the combined change will never yield an admissible point, if the procedure starts at stage i, as considering the change of stage (i+1) alone will give a higher cost decrease for a smaller time increase. However, $\Delta\theta_{ijm} < 0$, $\Delta\theta_{(i+1)gj} > 0$ may yield an admissible point, if the combined cost change is negative. If both $\Delta\theta$'s are negative, then the combined change, as much as each change considered separately, may yield an admissible point.

By recursively repeating the above argument, the restriction of following the optimal decisions from a certain stage on can be removed, and the total change in the time and cost values of stage n will be the sum of changes at each stage, where the changes of stage i and (i+1) are summed according to the transition:

$$Y_{i+1} = X_i$$

Introducing all possible changes would amount to enumerating all possible outcomes. By carrying forward only the "promising" changes at each stage according to the "promising change procedure," only a small fraction of all possible outcomes has to be evaluated, and the efficient set can be easily identified. The stage is set now to explain the algorithm step by step.

Steps 1, 2, 3. These steps should be obvious from the previous discussion.

Steps 4, 5. $\Delta Q_i(X_i,Y_i)$ and $\Delta \Theta_i(X_i,Y_i)$ represent the amount of possible change in one or more (T,C) pairs at stage n if moving away from optimality was to be made at stage i only. Note that each row will contain at least one zero in both cases.

Step 6. $Q_n(X_n,Y_n)$ represents the minimum possible value that can be obtained. If some of them are already greater than T_{C*}, they cannot be reduced by moving away from optimality in some stage i and therefore should be eliminated.

Step 7. The matrix Ω_i^m represents the accumulated time change at stage i for each x_{ij} and y_{ik}, if the "promising change procedure" has been applied to stage m through i-1, where m<i; likewise, the matrix Φ_i^m represents the same for the cost changes.

The procedure initially starts at stage 1 (m=1) yielding Ω_i^1, Φ_i^1. Then it is reapplied starting at stage 2(m=2), etc. By doing so, all possible accumulated changes are considered, allowing for selecting only the promising ones at each step. Due to the tests made at this step, the process converges rapidly, and from m=2 and up, only a few values have to be carried to stage n.

The reasoning behind the procedure for finding the starred elements (pg. 95) is as follows:

(a) Obvious

(b) (1) $(T_{C*}-T^*)$ is the maximum allowable time change in stage n. Therefore, if $\omega_{ijk_r}^m > (T_{C*}-T^*)$ it will not yield an admissible point.

(2) Should be clear from the initial discussion.

(3) Obvious.

(4) This test compares the accumulated changes at stage i, obtained when the procedure starts at stage m, with the previously chosen entries (starred elements) for the *same* x_{ij}, when the procedure started at stages 1...m-1. By the same reasoning as introduced before, if a previously chosen entry eliminates the possibility that the current one will become an admissible point, the current entry should be eliminated.

Obviously, if all elements of $\Omega_i^m(X_i,Y_i)$ are either eliminated or equal to zero (and therefore $\Phi_i^m(X_i,Y_i)$ is the same), the process terminates for this m.

(c) No point can dominate this one in the sense that a smaller time increase will yield a higher cost decrease. Therefore, it is always starred.

(d) Follows from the general discussion presented on page 100.

Step 8. $(\omega^m_{(n-1)jk_r^*}, \phi^m_{(n-1)jk_r^*})$ is the accumulated promising change, and is added to the basis from which the change is made--the initial values of $Q_n(X_n,Y_n)$, $\Theta_n(X_n,Y_n)$, obtained from the minimum time solution. These values by themselves are promising points.

Step 10. Obvious.

Step 11. The reasoning behind the test for C_e is that, as time increases, the cost cannot be higher than the cost associated with the previously selected (T_e^*,C_e^*), which is not necessarily the point immediately preceding the one under consideration.

The Efficient Set Algorithm-Cost Version is constructed in a similar manner to the Time Version. Instead of $Q_i(X_i,Y_i)$, the matrix $R_i(X_i,Y_i)$ is used, and the i-stage time matrix of the minimum cost solution is formed instead of $\Theta_i(X_i,Y_i)$.

Efficient Set Tableau. To reduce the computational effort associated with obtaining the efficient set, the tableau shown in Fig. 35 was developed. The tableau is used as follows: section 1 contains the matrices $\Delta\Theta(X_i,Y_i)$ and $\Delta Q_i(X_i,Y_i)$ that can be easily obtained by using the results of the minimum time solution. The matrices are imbedded in

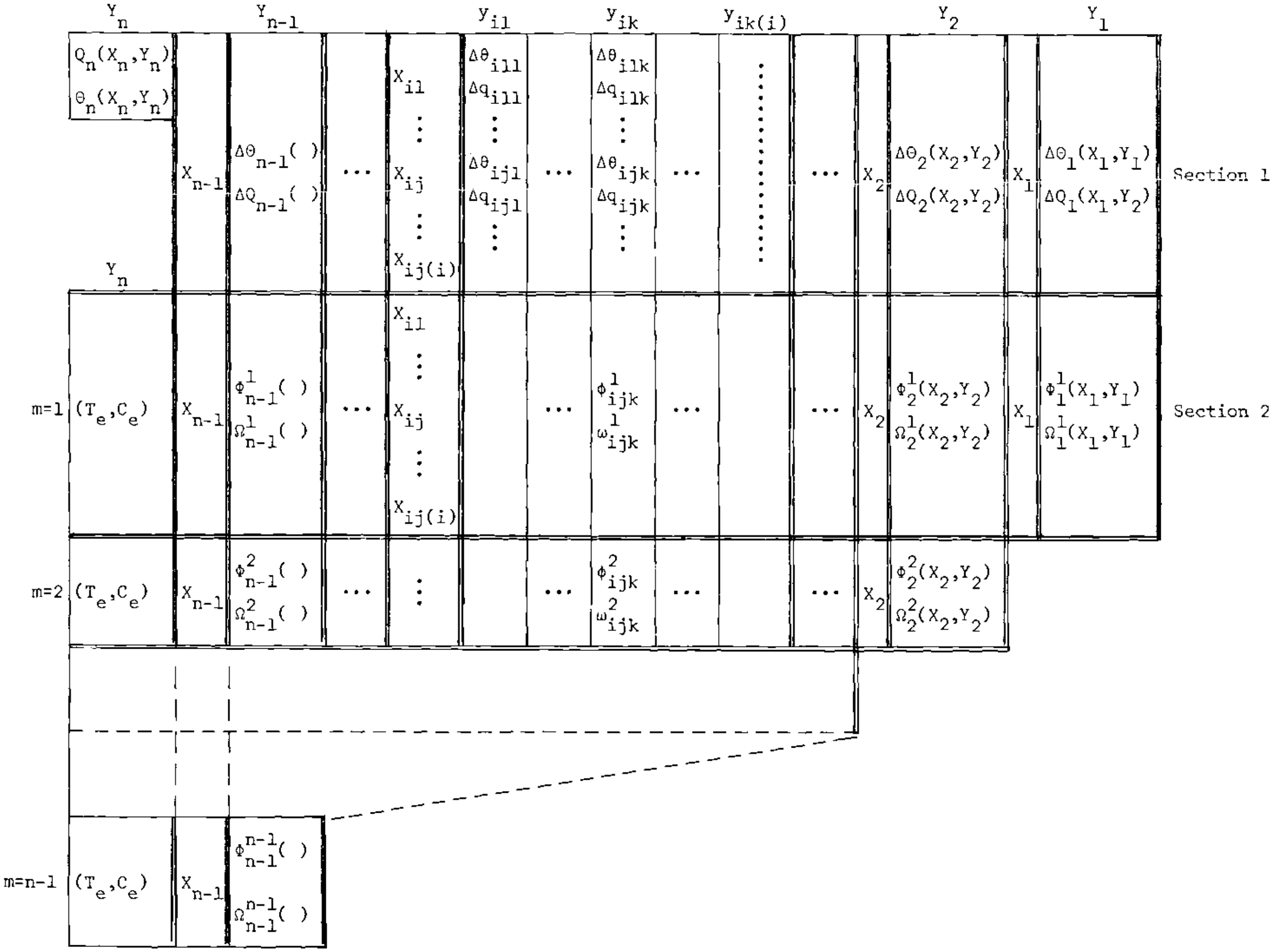


Figure 35. Efficient Set Tableau: Time Version

each other, so that each (x_{ij},y_{ik}) contains two values: the upper one is $\Delta\theta_{ijk}$, the lower one is Δq_{ijk}. Sections 2 and up are used to generate $\Theta_i^m(X_i,Y_i)$ and $\Omega_i^m(X_i,Y_i)$ for m=1...n-1. Again, each (x_{ij},y_{ik}) contains the values $(\phi_{ijk_r}^m,\omega_{ijk_r}^m)$ where each such slot is divided into a few sections so that different values of the above pairs can be accommodated.

The last column (Y_n) of section 1 contains the values of $\Theta_n(X_n,Y_n)$ and $Q_n(X_n,Y_n)$. The same space in sections 2,3 contains the values (T_p,C_p) generated in the process.

This whole procedure has been applied to a six-stage DP problem (see Appendix E) with a total of 4,096 possible outcomes. Only 22 points had to be evaluated, 8 out of which are elements of the efficient set.

The Efficient Set for a Decision Network

The purpose of this section is twofold:

a) to find the minimum cost of the minimum time solution, i.e.:

$$C_{T*} = \underset{m}{\mathrm{Min}}\{C_{(T*)m}\}$$

b) to develop time-cost trade-off for the case of certainty, i.e., find the efficient set $\{o_j^*\}$.

The method that achieves the above goals is an adaptation of the efficient set algorithm described previously. The procedure is discussed step by step, as follows.

Step 1. Apply the efficient set algorithm to each DDS (excluding the admissible point test of step 11).

The policy associated with the pair (T_e^i, C_e^i) is denoted by W_e^i, where T_e^i is the time of DDS_i, and C_e^i is the cost of DDS_i associated with T_e^i. For all $T_e^i < T^*$, the one with the smallest C_e^i is selected. Note that U_i^* is a specific entry for W_e^i.

No difficulty arises when applying this procedure to a nonserial DP model of the DDS. Since the algorithm is based upon incremental analysis, any incremental increase in each branch should be considered. For a type II divergence, the value of

$$\Delta q_{ijk} = q_{ijk} - \underset{k}{\text{Min}}\ q_{ijk}$$

is evaluated by taking for $\underset{k}{\text{Min}}\ q_{ijk}$ the value selected at this point for the time solution, which is not necessarily the minimum of the specific row (i.e., it was selected in another branch). This might create in some location $\Delta q_{ijk} < 0$, and these slots should be ignored. (See the example of Appendix B.)

The reason that the admissible point test of step 11 of the efficient set algorithm is excluded is because (T_e, C_e) has to be evaluated for the project as a whole.

Step 2. Construct the "DDS Efficient Set Table."

This table is shown in Fig. 36, and is constructed as follows: the table is divided vertically into two main sections A and B. Section A includes all decision vertices that are common to two or more DDS, and section B includes all other vertices, arranges according to their respective DDS. For each vertex, its $t_{jk(j)}$ and $c_{jk(j)}$ are given, and in addition, the vertices of section A are related to the proper DDS's. In column 3 all time values (T_e^i) are arranged in ascending order, and underlined l's (<u>l</u>) are placed in the proper columns of $(W_e^i0$ of sections A and B. The values of C_e^i, which are associated with a specific DDS_i, are entered in section I only. In column 1, the DDS_i associated with (T_e^i) is shown.

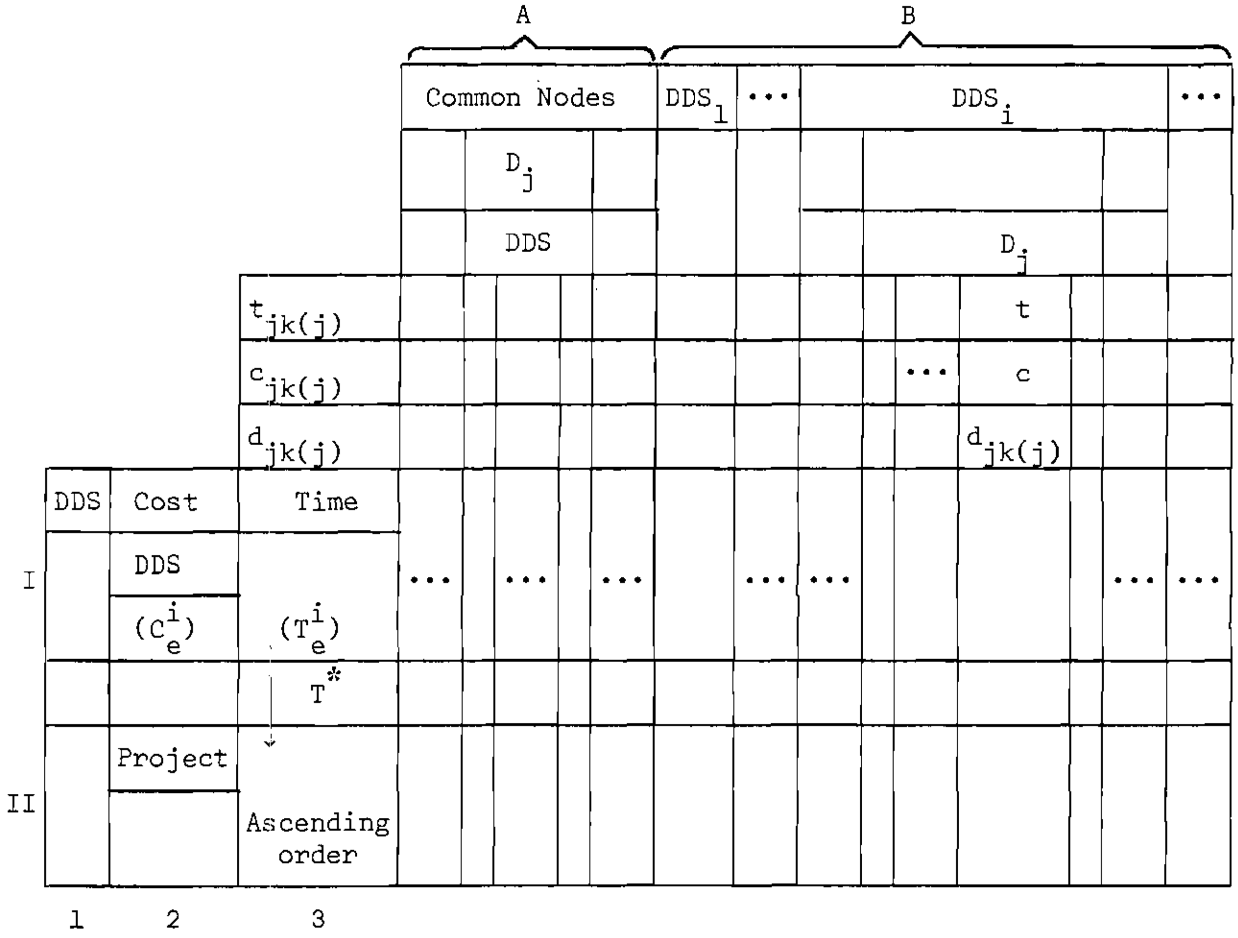


Figure 36. DDS Efficient Set Table

Step 3. Evaluate C_{T*}--the minimum cost of the minimum time.

This is done by using section I of the above table, as follows: no changes can be made in the DDS_i associated with T*, as any change would increase the project time. In all the rest of the $(DDS)_i$ appearing in section I the decision nodes selected can be changed, if this will not change T*. Thus, for DDS_i, one selects the row with $\underset{e}{\mathrm{Min}}(C^i_e)$, and eliminates the rest of them. The $\underline{1}$ in all the remaining rows indicate

$$U^* = \{d^*_{jk(j)}\}$$

associated with the minimum time T^*, and

$$C_{T*} = \sum_{k(j)*} c_{jk(j)*}$$

This procedure, as is, will take care of the case of two or more critical paths.

Step 4. Evaluate $O^* = \{o^*_j\}$.

This is done by using section II of the table as follows.

a) Start with the first value of this section. Consider the DDS_i associated with this T^i_e. Place an X in all columns *of this* DDS_i that have $\underline{1}$ above them and don't have $\underline{1}$ in them from step 2. Place 1's in all other columns of this row that have a $\underline{1}$ in some row above this one.

b) If section A includes $\underline{1}$ and X, that means that a different decision node is selected for a decision vertex that is common to two or more DDS_i. This creates different time values than the ones obtained previously for the other DDS, and this time value may be greater than the one under consideration, and therefore dominating.

To check this, all the values of $t_{jk(j)}$ associated with columns with 1's in them (both 1 and $\underline{1}$) for a specific DDS_i, are added, i.e.:

$$\sum_{\text{(All } k(j) \text{ with 1's) } \epsilon \text{ (DDS)}_i} t_{jk(j)} \qquad \forall\ i.$$

If one of these values is greater than the T_e under consideration, this row is eliminated. This value will reappear in its proper place in the T_e sequence, if it is associated with a promising point.

Step 5. Evaluate C_e as follows.

$$C_e = \sum_{\text{All } k(j) \text{ with 1's (both } \underline{1} \text{ and 1)}} c_{jk(j)}$$

If $C_e < C_{T*}$, the pair (T_e,C_e) is an admissible point (where $T_e = T_e^i$), with W_e--the policy for the whole project associated with this point--being composed of all $d_{jk(j)}$ which include $\underline{1}$ or 1 in their columns.

Step 6. Repeat the procedure, starting at step 4, for the succeeding values of T_e^i, with the following changes:

(a) An X is placed in columns of the DDS_i associated with the specific T_e^i, that do not have $\underline{1}$ in them, and have either a $\underline{1}$ or 1 above them, with no X in between.

(b) For all other DDS_i, a 1 is placed in columns that have either a $\underline{1}$ or 1 above them with no X in between.

(c) The test of step 5 is now:

If: $C_e < C_i \quad \forall\, e^* < i \leq e - 1$,

where e^* is the subscript associated with T^*, i.e., $T_{e^*} = T^*$.

Then: (T_e, C_e) is an admissible point, with the policy W_e as explained above. Note that U^* and V^* are specific entries for W_e.

The above procedure and table combine the results of the Efficient Set Algorithm applied to each DDS_i separately in such a way so as to eliminate all points that might be admissible for a particular DDS_i, but are not admissible when the whole project is considered.

Note that the "Cost Refinement" discussed on page 84 creates no problem here, as the procedure is basically incremental analysis of the whole project. This procedure has been applied to the example of Appendix B.

Constraints and Sensitivity

The solution procedure developed for the case of certainty has the advantage that budget and time constraints can be added, changed or dropped without any additional effort.

Once the Efficient Set is obtained, adding a time constraint amounts to a vertical line on the time-cost trade-off graphical description, whereas a budget constraint is a horizontal line on the same curve, as illustrated in Fig. 37.

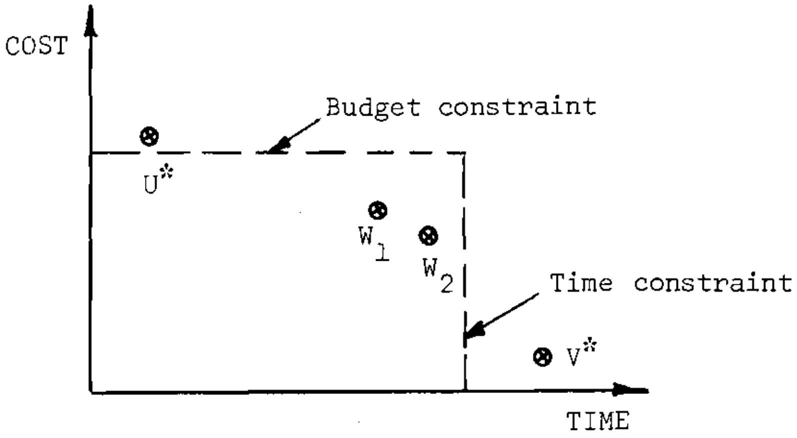


Figure 37. Constraints

This approach has the added advantage in that the sensitivity of changing the constraint can be immediately obtained. For example, in Fig. 37, increasing the budget by a small amount would sharply reduce the total project time.

CHAPTER V

THE CASE OF RISK WITH STOCHASTIC DECISIONS

Introduction

This case is an extension of the deterministic case, discussed in the previous chapter. As before, there are different alternatives of performing some of the activities, each having a different cost, different duration and different technological dependencies. However, each alternative has now a probability associated with it. This probability is a measure of the relative chance that a certain alternative will be chosen as the work progresses. Thus, as was indicated in Chapter I (p. 6) in contrast to the case of certainty, where a selection among alternatives was possible during the planning phase, no such selection is possible now. The selection of a specific alternative depends upon the outcome of the preceding activities, and therefore no preliminary elimination can be made, but instead, the probability of selecting a certain alternative can be stated (typically, this would be a subjective probability). Situations like this are common in research and development projects.

In spite of the fact that neither elimination of alternatives is possible during the planning phase nor reduction of the decision network into a standard network, the decision maker is still in need of some information in order to decide whether to proceed with the project or not.

Specifically, it is suggested that the following information is of importance.

1) Project time and cost extremes.
2) Expected project time and expected project cost.
3) Risk Evaluation.
4) Most probable project network, its time and cost.

It can be seen that this case is essentially an extension of the problem handled by Eisner (17), and is a different approach to the problem suggested by Dean (14). It is felt that the procedure suggested here is more adequate for the stated objective of providing a decision making tool during the planning phase.

Project Time and Cost Extremes

It was pointed out in Chapter I that for this case, developing time-cost trade-off is impossible, as no alternative elimination is possible during the planning phase. However, for decision making purposes, information about the extreme values of project time and cost is of major importance. These values define a closed interval in E^2, called "The Region of Possible Outcomes."

No matter what the outcome will be, it is going to be included in the "Region of Possible Outcomes," as described in Fig. 38. To establish this region, four values have to be evaluated, as follows: minimum time, maximum time, minimum cost, maximum cost.

It should be noted that the cost associated with the minimum time, the time associated with the minimum cost, etc., is of no importance, as one is interested here in setting the boundaries of the

"Region of Possible Outcomes," and not in a specific point.

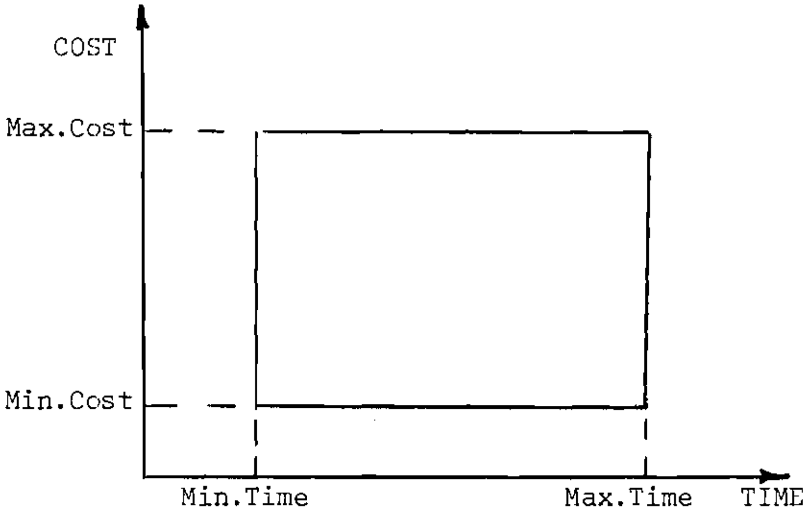


Figure 38. Region of Possible Outcomes

To evaluate these boundaries, the probabilities associated with each stochastic decision vertex are ignored, and the project decision network is handled as in the case of certainty, solving the following four problems.

(1) Minimum time problem.

(2) Minimum cost problem.

(3) Maximum time problem.

(4) Maximum cost problem.

Obviously, the first two problems are exactly the ones solved for the case of certainty, with the exception that the values of C_{T*} and T_{C*} do not have to be evaluated.

The last two problems present no difficulty either, as the solution procedure is exactly the same as for the minimum time and minimum cost problems, with the exception that the optimization process is a maximization one instead of minimization. This whole procedure is shown in Appendix C.

Expected Project Time and Cost

The purpose of this section is to develop another input to the project evaluation process during the planning phase. The procedure is based upon the concept of expected value.

In the literature, there is some debate as to whether the expected value concept, which is based upon "long run average," is applicable in cases of "one time only" situation. Another criticism of the expected value concept is the fact that it bases decisions on the average only, and gives no consideration to the extremes. In spite of these criticisms, expected value is the principle of choice most often used. Furthermore, for this case and the one discussed in Chapter VI, it is going to be only one out of a few inputs to the decision making process.

Expected Project Time

No optimization process as such is involved in finding the expected project time for this case, since no selection among alternatives can be performed during the planning phase. Thus, the decision network cannot be reduced to a standard network in the same way as in the case of certainty.

The concept of expected project time needs some further elaboration, before the method of evaluating it is introduced. Even in the case of a standard network with individual activity times being a random variable (PERT network), the expected time of the project has been approached in various ways, based upon different sets of assumptions. This problem, for the PERT type network, has been discussed by Fulkerson (23) and Elmaghraby (20). Both methods are described as estimates of the true expected value.

A similar approach is used here. Two methods are presented: evaluation of an optimistic expected time, i.e., approaching the true expected time from below, and a pessimistic expected time, approaching it from above. The two methods are associated with different sets of assumptions concerning the project control policies during the execution of the project. The true expected value is somewhere between these two estimates. The two methods are the same and yield the expected value only when the network has independent DDS only, i.e. no common decision vertices.

Optimistic Expected Time. The first case to be considered is the case where the project network can be decomposed into only one DDS and one CDDS as illustrated in Fig. 39.

Associated with each decision node $d_{jk(j)}$ of the DDS is a probability $p_{jk(j)}$. No probability is associated with the elements of CDDS. Obviously:

$$\sum_{k(j)} p_{jk(j)} = 1 \quad \forall \; j$$

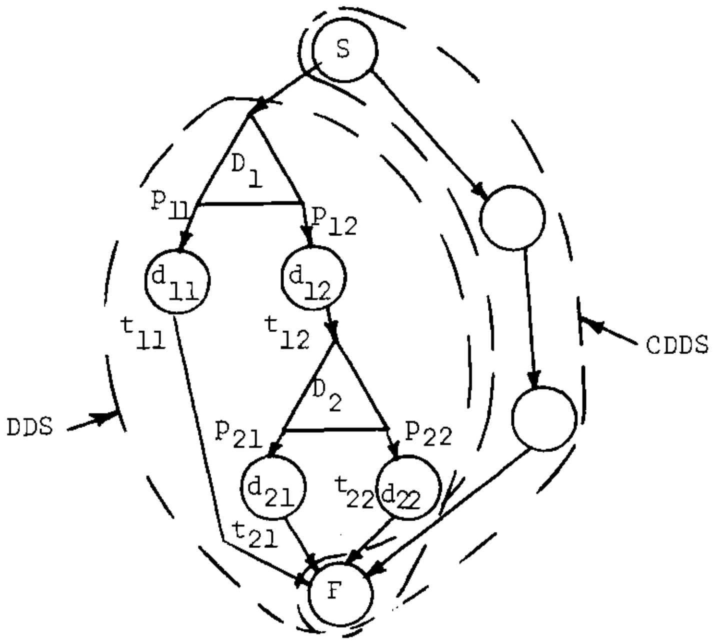


Figure 39. A Stochastic Decisions Network

Let:

$E[T^i]$ be the expected time to get from S to F, associated with DDS_i.

$E[T^o] = T^o$ be the time to get from S to F, for CDDS, and

E[T] be the expected time of the stochastic decisions network.

Then: $E[T] = \text{Max}(E[T^1],T^o)$.

For the case of more than one independent DDS, one gets:

$$E[T] = \underset{i}{\text{Max}}(E[T^i]) \qquad i=0,1,\ldots$$

To evaluate $E[T^i]$ the process associated with this DDS is viewed as a Markov Process, as follows: Each decision node of the DDS is viewed as a "state." The probability associated with each decision node is viewed as a transition probability. Thus, p_{11} represents the transition probability from state "S" to state "11" (i.e., d_{11}), and p_{21} is viewed as the transition probability from state "12" to state "21." In general, let $p_{j|i}$ be the conditional probability that a system which now occupies state i will occupy state j after its next transition. Thus, using the above notation:

$$p_{12} = p_{12|S}$$

$$p_{22} = p_{22|12}$$

and in general, for describing the Markovian transition probabilities, the probability $p_{jk(j)}$ of decision node $d_{jk(j)}$ is regarded now as the transition probability from state (mk(m)) to state (jk(j)) as follows:

$$p_{jk(j)|mk(m)}$$

where jk(j), etc., is regarded as one subscript.

Since the system must be in some state after its next transition, then:

$$\sum_{j} p_{j|i} = 1 \quad \forall \quad i$$

Note that in this case, since the network is acyclic

$$p_{i|i} = 0 \quad i \neq F$$

and since the system is absorbed in F,

$$p_{F|F} = 1$$

Associated with each state is a "return" or "reward." These returns are the time values associated with each decision node of the DDS as evaluated in step 8 of the second level labeling algorithm for the minimum time problem of the previous chapter. To comply with the notation used for the transition probabilities, the time return $t_{jk(j)}$ associated with decision node $d_{jk(j)}$ will take the following general format whenever used in relation with the Markov process.

$t_{j|i}$ - the time return associated with the transition from state i to state j.

Thus, in Fig. 39, t_{22}, when considered for the Markov process will be $t_{22|12}$, etc.

The rationale of viewing the stochastic decision network of this case as a Markov process stems from the internal logic of the network. Referring to Fig. 39, during the planning phase it can be safely claimed that once decision node 12 is realized, the probability of selecting decision node "21" is p_{21}, no matter how decision node "12" is reached.

Using the symbol ◇ to describe a state of the Markov process, the DDS of Fig. 39 can be described as shown in Fig. 40.

Finding the expected time of the DDS of Fig. 39 is equivalent to finding the expected time of the Markov process of Fig. 40.

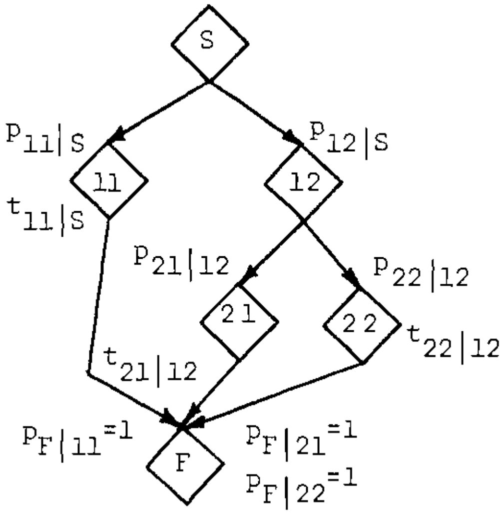


Figure 40. Markov Process of a DDS

A solution procedure to this type of problem has been developed by Howard (31). He called it "Markov Process with Rewards." Using Howard's procedure, one gets:

Let:

$\underline{\underline{T}} = (t_{j|i})$ - m×m "time return matrix" associated with the Markov Process (m is the total number of states).

$\underline{\underline{P}} = (p_{j|i})$ - m×m transition matrix (stochastic matrix) associated with the Markov process.

$\bar{t}_n(i)$ - expected total time return in the next n transitions if the system now is in state i.

$\bar{\underline{T}}_n = (\bar{t}_n(i))$ - expected total time return vector--m×1, if m is the total number of states.

For the Markov process described by Fig. 39 the matrix $\underline{\underline{P}}$ is:

$$\underline{\underline{P}} = \begin{pmatrix} p_{S|S} & p_{11|S} & p_{12|S} & p_{21|S} & p_{22|S} & p_{F|S} \\ p_{S|11} & p_{11|11} & p_{12|11} & p_{21|11} & p_{22|11} & p_{F|11} \\ p_{S|12} & p_{11|12} & p_{12|12} & p_{21|12} & p_{22|12} & p_{F|12} \\ p_{S|21} & p_{11|21} & p_{12|21} & p_{21|21} & p_{22|21} & p_{F|21} \\ p_{S|22} & p_{11|22} & p_{12|22} & p_{21|22} & p_{22|22} & p_{F|22} \\ p_{S|F} & p_{11|F} & p_{12|F} & p_{21|F} & p_{22|F} & p_{F|F} \end{pmatrix}$$

Note that the sum of the probabilities for each row is 1, so that some of the elements may be zero. In a similar way the matrix $\underline{\underline{T}}$ can be constructed.

Using a DP approach, Howard suggests the following recurrence relationships.

$$\bar{t}_1(i) = \sum_j p_{j|i} t_{j|i} \quad \forall \; i$$

$$\bar{t}_n(i) = \sum_j p_{j|i} \left(t_{j|i} + \bar{t}_{(n-1)}(j) \right) \quad \forall \; i \qquad n=1,2...$$

or:

$$\bar{t}_n(i) = \bar{t}_1(i) + \sum_j p_{j|i} \bar{t}_{(n-1)}(j) \quad \forall \; i, \qquad n=1,2...$$

Using vector notation, the last equation can be rewritten as:

$$\underline{\bar{T}}_n = \underline{\bar{T}}_1 + \underline{\underline{P}}\,\underline{\bar{T}}_{n-1}$$

The above procedure always terminates (as proved later) for n* when:

$$\bar{\underline{T}}_{n^*+1} = \bar{\underline{T}}_{n^*}$$

The expected time of the DDS is:

$$E[T^1] = \bar{t}_{n^*}(S)$$

where the superscript 1 indicates that there is only one DDS in this case.

The following theorem shows that this process will always terminate.

Theorem 3. The recursive procedure for finding $\bar{\underline{T}}_{n^*}$ always terminates for a finite $n = n^*$.

Proof. The Markov process associated with this procedure is composed of an absorbing Markov chain. This is so because the stochastic matrix $\underline{\underline{P}}$ has:

a) exactly one absorbing state (state F).

b) from every state it is possible to go to this state (not necessarily in one transition).

Since the Markov chain is absorbing, the probability that the process will be absorbed is 1. Since there is a finite number of states, and all states except state "F" are transient, and since the original network has no cycles, therefore the matrix $\underline{\underline{P}}$ has no cycles, thus the process will be absorbed in a finite number of steps. Once the process is absorbed, the expected return cannot change, and therefore:

$$\bar{\underline{T}}_{n^*+1} = \bar{\underline{T}}_{n^*} \quad \text{Q.E.D.}$$

The discussion so far has not touched on the problem of the optimistic expected time. This problem appears when there is more than one DDS involved, and at least one common decision vertex. Suppose that the stochastic decision network of Fig. 41 is given:

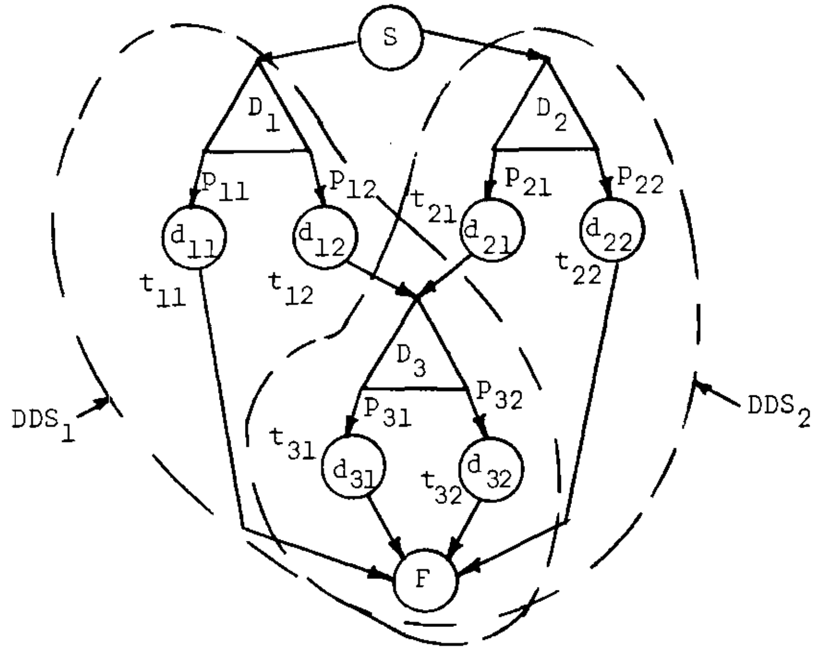


Figure 41. Stochastic Decisions Network With a Common Decision Vertex

Obviously, there are two DDS, with D_3 common to both. The optimistic expected time, and the one that follows--the pessimistic expected time--are based upon different assumptions regarding the policies of the project controller while the project is carried out.

Project Control Policies. The policies considered here are concerned with a common decision vertex, like D_3. Suppose that activity

"12" is completed before any decision has been made concerning activity "21." Due to the uncertainty involved with the execution of activity "21," two possible courses of action can be taken as follows:

(1) The project controller may decide to go ahead with activity "31" or "32" without waiting to see what happens with activity "21."

This is especially true when "12" is completed quite a long time before a decision can be made about activity "21."

(2) The project controller may decide to wait for the outcome concerning activity "21."

The optimistic expected time which yields a lower value of the expected time, is associated with the first policy. The pessimistic expected time with the second.

<u>The Markov Process for the Optimistic Expected Time</u>. If the first policy mentioned before is assumed, the meaning is that each DDS can be regarded as an independent Markov chain. Note that the two DDS cannot be regarded as one Markov chain, since then the "states" will not be mutually exclusive--a basic requirement for a Markov process. Thus, one gets the Markov Processes illustrated in Fig. 42.

In this case, states "31" and "32" are common to the two chains, and:

$$p_{31} = p_{31|12} = p_{31|21}$$

$$p_{32} = p_{32|12} = p_{32|21}$$

i.e., the transition probability from "12" to "31" and "32" is independent of the transition probabilities from "21."

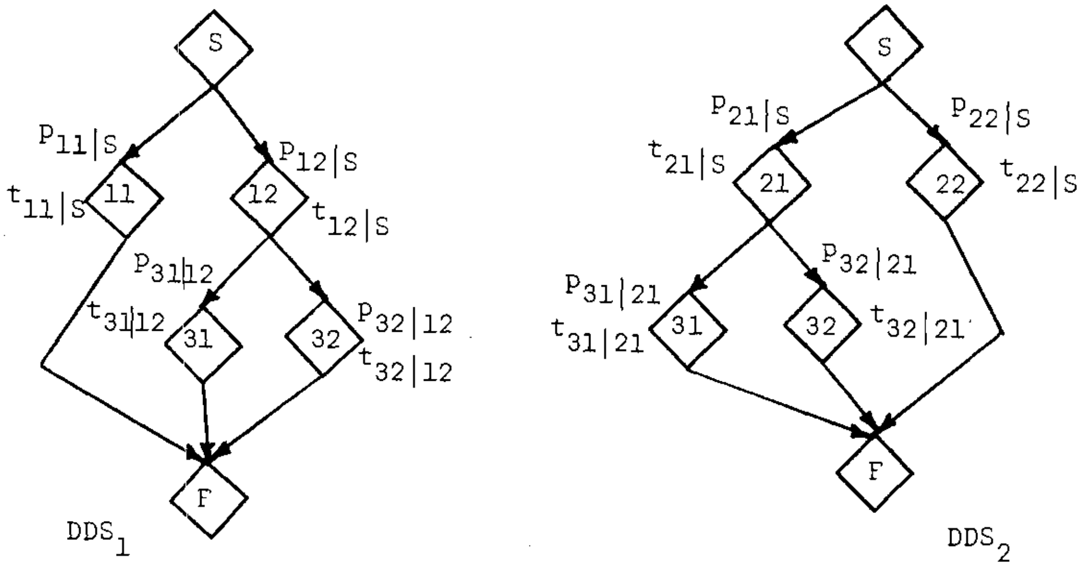


Figure 42. The Equivalent Markov Processes

Generalizing this approach, first the project network is decomposed into DDS with the proper time returns. Then, each DDS is treated as a separate Markov process, and $\bar{t}^{i}_{n*}(S)$ is evaluated for each DDS_i. Thus:

$$E[T^i] = \bar{t}^{i}_{n*}(S)$$

$$E[T] = \underset{i}{\mathrm{Max}}\left(\bar{t}^{i}_{n*}(S), T^{o}\right)$$

The method described above yields the lower bound of the optimistic expected time. This claim is proved in Theorem 4 in the next section. In order to obtain the exact value of the optimistic expected time, the probabilities of path combinations among the various DDS have to be evaluated (see the value of EV in Theorem 4). This amounts to an extensive computational effort even for small networks, and therefore this approach is not recommended.

Theorem 4. The above method of evaluating the optimistic expected time gives a lower bound of this value.

Proof. Consider a stochastic decision network as shown in Fig. 43.

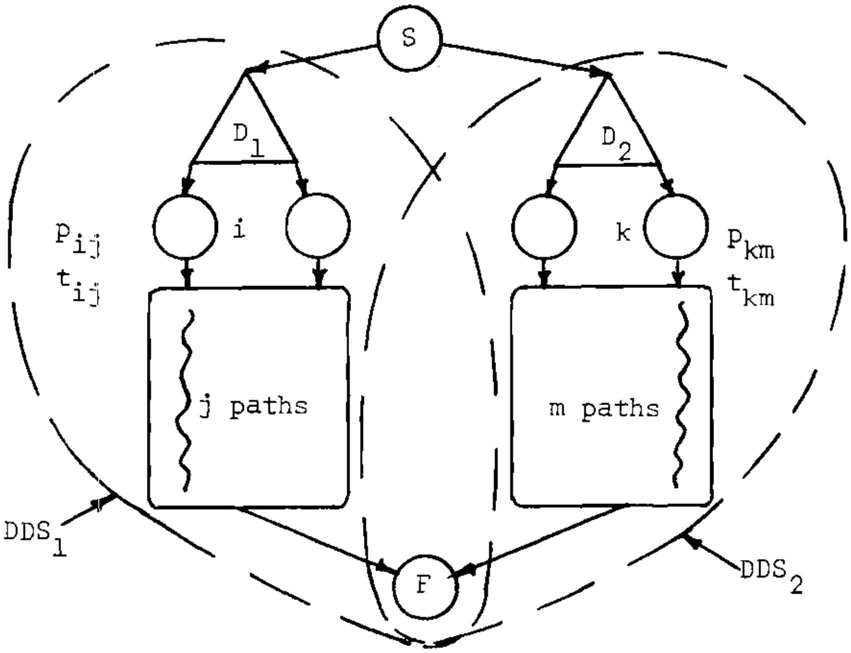


Figure 43. Stochastic Decisions Network: Schematic Illustration

Only the first decision vertex of each DDS is shown. The problem of a common decision node does not exist here, since for the optimistic case each DDS is considered separately.

The solution procedure outlined above for finding the expected time of each DDS yields the same result as the process of enumerating all paths from each decision node to F, evaluating their times and probabilities, and taking the expected value at the first decision vertex. Thus, let:

p_{ij} - probability that the jth path to F of the ith decision node[1] will be realized.

t_{ij} - time associated with this path.

Then the expected time of DDS_1 is:

$$\sum_i \sum_j p_{ij} t_{ij}$$

which is equal to the value obtained by the Markov Process approach. Now, suppose

$$\sum_i \sum_j p_{ij} t_{ij} \geq \sum_k \sum_m p_{km} t_{km}$$

Then, $\sum_i \sum_j p_{ij} t_{ij}$ is the value of E[T] obtained before for this network. Let:

---

[1] For simplicity of presentation, the subscript jk(j) used for a decision node is substituted with a single subscript i for this theorem.

$$\left.\begin{aligned} LB &= \sum_i \sum_j p_{ij} t_{ij} \\ LV &= \sum_k \sum_m p_{km} t_{km} \\ LB &\geq LV \end{aligned}\right\} \text{where LB stands for "Lower Bound" and LV stands for "Lower Value"}$$

and

$$\sum_i \sum_j p_{ij} = 1$$

$$\sum_k \sum_m p_{km} = 1$$

The true optimistic expected value of this network is given by

$$EV = \sum_k \sum_m \sum_i \sum_j p_{ij} p_{km} \mathrm{Max}(t_{ij}, t_{km})$$

Thus, it has to be shown that, $EV \geq LB$. Now since:

$\mathrm{Max}(a,b) = \frac{1}{2}(|a-b| + a + b)$.

Then:

$$EV = \frac{1}{2} \sum_k \sum_m \sum_i \sum_j p_{ij} p_{km} (|t_{ij} - t_{km}| + t_{ij} + t_{km}) =$$

$$= \frac{1}{2} \sum_k \sum_m p_{km} \Big(\sum_i \sum_j p_{ij} (|t_{ij} - t_{km}|) + \sum_i \sum_j p_{ij} t_{ij} + t_{km} \sum_i \sum_j p_{ij}\Big)$$

$$EV = \frac{1}{2} \sum_k \sum_m p_{km} \Big[\sum_i \sum_j |t_{ij} - t_{km}| + LB + t_{km}\Big]$$

$$EV = \frac{1}{2} \sum_k \sum_m \sum_i \sum_j p_{ij} p_{km} |t_{ij} - t_{km}| + \frac{1}{2} LB \sum_k \sum_m p_{km} + \frac{1}{2} \sum_k \sum_m p_{km} t_{km}$$

$$EV = \frac{1}{2} \sum_k \sum_m \sum_i \sum_j p_{ij} p_{km} |t_{ij} - t_{km}| + \frac{1}{2} LB + \frac{1}{2} LV$$

Since:

$$(a-b) \leq |a-b|$$

Then:

$$EV \geq \frac{1}{2} \sum_k \sum_m p_{km} \sum_i \sum_j p_{ij} t_{ij} - \frac{1}{2} \sum_i \sum_j p_{ij} \sum_k \sum_m p_{km} t_{km} + \frac{1}{2} LB + \frac{1}{2} LV$$

$$EV \geq \frac{1}{2} LB - \frac{1}{2} LV + \frac{1}{2} LB + \frac{1}{2} LV$$

$$EV \geq LB \qquad \text{Q.E.D.}$$

The proof can be easily extended to more than one DDS.

The above procedure for the optimistic expected time has to be somewhat modified when diverging stages are considered. The case of type II divergence is shown in Appendix C.

Pessimistic Expected Time. This method is based upon the second project control policy discussed before. Thus, referring to Fig. 41, if "12" is completed before any action can be taken concerning "21," the project controller will wait until a decision is made about "21," and if "21" is selected, he will wait until its completion.

To evaluate the expected time in this case, one has to evaluate the probability and expected time of reaching decision vertex D_3. This is done as follows:

$$\begin{aligned} P(\text{reaching } D_3) &= P(12 \cup 21) = P(12) + P(21) - P(12 \cap 21) \\ &= P(12) + P(21) - P(12)P(21) \\ &= p_{12} + p_{21} - p_{12}p_{21} \end{aligned}$$

The expected time of reaching D_3 is composed of three elements, as follows:

Activity Occurrence	Probability	Equivalent Time
12 but not 21	$P(12) - P(12 \cap 21) = p_{12} - p_{12}p_{21}$	t_{12}
21 but not 12	$P(21) - P(12 \cap 21) = p_{21} - p_{12}p_{21}$	t_{21}
12 and 21	$P(12 \cap 21) = p_{12}p_{21}$	$\text{Max}(t_{12},t_{21})$

and the expected time is:

$$p_{12}t_{12} + p_{21}t_{21} + p_{12}p_{21}\left(\text{Max}(t_{12},t_{21}) - t_{12} - t_{21}\right)$$

Noticing that the other set of probabilities associated with this network is:

$$P(\overline{12 \cup 21}) = P(\overline{12} \cap \overline{21}) = P(11 \cap 22) = p_{11}p_{22}$$

The total expected time becomes

$$E[T] = p_{12}t_{12} + p_{21}t_{21} + p_{12}p_{21}[\mathrm{Max}(t_{12},t_{21}) - t_{12} - t_{21}] +$$

$$+ (p_{12} + p_{21} - p_{12}p_{21})(p_{31}t_{31} + p_{32}t_{32}) +$$

$$+ p_{11}p_{22}[\mathrm{Max}(t_{11},t_{22})]$$

The value obtained using this method is higher than the one in the previous case. The true expected value lies somewhere in between the optimistic and pessimistic expected values.

Generalization of this method is possible; however, it seems that even for a moderate size network there is an immense amount of calculations involved. A simulation approach seems to be more appropriate for this case. This is discussed later.

Example

In Fig. 41, let:

$p_{11} = 0.3$ $p_{12} = 0.7$ $p_{21} = 0.4$ $p_{22} = 0.6$ $p_{31} = 0.2$ $p_{32} = 0.8$

$t_{11} = 10$ $t_{12} = 15$ $t_{21} = 20$ $t_{22} = 10$ $t_{31} = 5$ $t_{32} = 10$

Optimistic Expected Time. It is easy to convert the above probabilities and returns to their Markovian equivalent, yielding, for DDS_1:

$$\underline{\underline{P}} = \begin{array}{c|cccccc|} & S & 11 & 12 & 31 & 32 & F \\ \hline S & & 0.3 & 0.7 & \mathbf{O} & & \\ 11 & & & & & & 1 \\ 12 & & & & 0.2 & 0.8 & \\ 31 & & & & & & 1 \\ 32 & & & \mathbf{O} & & & 1 \\ F & & & & & & 1 \end{array} \qquad \underline{\underline{T}} = \begin{array}{c|cccccc|} & S & 11 & 12 & 31 & 32 & F \\ \hline S & & 10 & 15 & \mathbf{O} & & \\ 11 & & & & & & \\ 12 & & & & 5 & 10 & \\ 31 & & & & & & \\ 32 & & & \mathbf{O} & & & \\ F & & & & & & \end{array}$$

Thus:

$$\underline{\bar{T}}_1 = \begin{bmatrix} \bar{t}_1(S) \\ \bar{t}_1(11) \\ \bar{t}_1(12) \\ \bar{t}_1(31) \\ \bar{t}_1(32) \\ \bar{t}_1(F) \end{bmatrix} = \begin{bmatrix} 13.5 \\ 0 \\ 9 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

and $\underline{\bar{T}}_n$, for different n is:

n =	1	2	3
$\bar{t}_n(S)$	13.5	19.8	19.8
$\bar{t}_n(11)$	0	0	0
$\bar{t}_n(12)$	9	9	9
$\bar{t}_n(31)$	0	0	0
$\bar{t}_n(32)$	0	0	0
$\bar{t}_n(F)$	0	0	0

$n^* = 2$

$$E[T^1] = \bar{t}_{n*}(S) = 19.8$$

By the same method,

$$E[T^2] = 17.6$$

$$E[T] = \text{Max}(19.8,17.6) = \underline{19.8} \text{ (The lower bound)}$$

Pessimistic Expected Time.

$$E[T] = (0.7)(15) + (0.4)(20) + (0.4)(0.7)(\text{Max}(15,20) - 15 - 20) +$$

$$+ (0.7 + 0.4 - 0.4 \times 0.7)[(0.2)(5) + (0.8)(10)] +$$

$$+ (0.3)(0.6)[\text{Max}(10,10)] = \underline{23.48}$$

Expected Project Cost

The same comments that were made before relative to the expected time problem are applicable here. In the same way, two values of the expected cost will be given: an optimistic value and a pessimistic value. The two values are the same and give the true expected cost when the network has independent DDS. It should be noted that the pessimistic estimate of the cost is associated with the optimistic estimate of the time, and vice versa.

The first step for both methods is decomposing the *RCN* into DDS, and evaluating the cost return associated with each alternative (disregarding the probabilities), as was done in the previous chapter.

Pessimistic Expected Cost. The same assumptions that were made for the optimistic expected time are valid here.

The expected cost of each DDS is evaluated using the Markovian procedure introduced for the optimistic expected time with the following obvious change in notations. Let:

$\underline{\underline{C}} = (c_{j|i})$ - m×m "cost return matrix associated with the Markov Process.

$\bar{c}_n(i)$ - expected total cost return in the next n transitions if the system now is in state i.

$\bar{\underline{C}}_n = \left(\bar{c}_n(i)\right)$ - expected total cost return vector - m×1.

Then, in the same way as before:

$$\bar{\underline{C}}_n = \bar{\underline{C}}_1 + \underline{\underline{P}}\,\bar{\underline{C}}_{n-1}$$

and

$$E[C^i] = \bar{c}^{i}_{n*}(S)$$

if C^i is the cost of DDS_i

$$E[C] = \sum_i \bar{c}^{i}_{n*}(S) + C_{PN}$$

where C_{PN} is the total cost of the permanent nodes.

The problem of a lower bound does not exist here, as in this case, using the notations of Theorem 4 for Cost Value, one has:

$$EV = LB + LV$$

The above procedure has to be somewhat modified when diverging stages are considered. The case of type II divergence is shown in Appendix C.

Optimistic Expected Cost. The assumptions associated with this case are the same as those of the pessimistic expected time. Referring to Fig. 41, the difference between this case and the previous one is in considering activities "31" and "32." According to the previous case, these two activities are viewed as performed separately for each DDS, i.e. as a continuation of "12" regardless of "21" and vice versa. The approach here is that if both "12" and "21" are performed, the cost of "31" and "32" should be included only once to avoid "double counting." This means that from the expected cost obtained before, the expected cost of "31" and "32," when *both* 12 and 21 occur, has to be subtracted; i.e. for the network of Fig. 41, one gets:

$$E[C] = \bar{c}^{1}_{n*}(S) + \bar{c}^{2}_{n*}(S) - P(12\cap 21)(p_{31}c_{31} + p_{32}c_{32})$$

or:

$$E[C] = \bar{c}^{1}_{n*}(S) + \bar{c}^{2}_{n*}(S) - p_{12}p_{21}(p_{31}c_{31} + p_{32}c_{32})$$

Generalization of this procedure is relatively simple. The probabilities p_{12}, p_{21} of the previous example will be the probabilities of realizing the decision nodes (states) immediately preceding the common decision vertices. When these states stem from independent Markov processes, the probability of realizing each one of them is obtained as follows:

Let:

$\pi_n(i)$ - probability that the system will occupy state i after n transitions, if its state at n = o is known.

Then:

$$\sum_i \pi_n(i) = 1$$

$$\pi_{n+1}(j) = \sum_i \pi_n(i)p_{j|i} \qquad n=0,1,2...$$

The above set of difference equations is solved separately for each DDS for S through k, where k represents the specific decision node.

The expected cost of the mutual states is readily available from the expected cost solution of the DDS. This is the value of $\bar{c}_{n*}(k)$ if k is the state immediately preceding the common decision vertex. This value can be taken from the $\bar{\underline{C}}_{n*}$ of any DDS, as it is going to be the same.

In evaluating the common states' expected cost that has to be deducted in order to avoid "double counting" one has to consider the possibility of more than two preceding states, as follows:

Let:

k^i - the state immediately preceding the common decision vertex associated with $(DDS)_i$.

$\pi_n(k^i)$ - probability of being in this state, after n transitions (note that n can be different for different DDS).

$E[C_{CN}]$ - total expected cost of all common decision nodes that should be deducted.

Then:

$$E[C_{CN}] = [-\sum_{i<j=2} \pi_n(k^i)\pi_n(k^j) + \sum_{i<j<r=3} \pi_n(k^i)\ \pi_n(k^j)\ \pi_n(k^r)$$

$$+ \ldots + (-1)^{m-1}\ \pi_n(k^1)\ \pi_n(k^2)\ \ldots\ \pi_n(k^m)]\ \bar{c}_{n*}(k)$$

where the decision vertex is common to m DDS's and $\bar{c}_{n*}(k)$ can be taken from the $\bar{\underline{C}}_{n*}$ of any DDS.

The total expected cost is then:

$$E[C] = \sum_i \bar{C}^i_{n*}(s) - E[C_{CN}] + C_{PN}$$

<u>Example</u>

Suppose that in the example solved for the expected time, the numbers represent cost values. Then:

$$k^1 = 12 \qquad k^2 = 21$$

$$\bar{c}_{n*}(12) = \bar{c}_{n*}(21) = 9$$

$$\pi_1(12) = 0.7 \qquad \pi_1(21) = 0.4$$

$$E[C_{CN}] = (0.7)(0.4)9 = 2.52$$

$$E[C]_{ps} = \sum_i \bar{c}^i_{n*}(S) = 19.8 + 17.6 = 37.4$$

$$C_{PN} = 0$$

$$E[C]_{op} = 37.4 - 2.52 = 34.88$$

Some modifications have to be introduced for the case of diverging branches. This is shown in Appendix C.

Risk Evaluation

A different approach to evaluating the project is by obtaining the probability distribution of the time and cost. Analytically, this can be done by enumerating all independent path combinations from "S" to "F" and evaluating their probability, time and cost. This method is impractical even for small networks, especially when common decision vertices are present. Simulation would be a more efficient approach, where each decision vertex with its probability distribution is viewed as a stochastic vertex, so that Monte Carlo simulation can be applied. This approach is demonstrated in the next chapter, for the minimum time and minimum cost networks, which are essentially the equivalent of the stochastic decision network discussed herein. Once the probability distribution of time and cost is obtained, the respective expected values can be easily evaluated avoiding the difficulty of a common decision vertex.

Most Probable Project Network

Another approach to project evaluation during the planning phase is to consider the most probable course of action that might be followed.

Once this is done, the stochastic decisions network is reduced to a standard network, and its time and cost are evaluated.

The procedure of finding the most probable project network is rather simple. Again, the stochastic decisions network is divided into DDS. For each DDS, one starts with the first decision vertex evaluating:

$$p_{1k(1)*} = \underset{k(1)}{\text{Max}}\, p_{1k(1)}$$

and accordingly, decision node $d_{1k(1)*}$ is selected for this decision vertex. Once this is done, the rest of the decision nodes $d_{1k(1)}$ are eliminated, together with the nodes associated with them, according to the procedure described in Chapter III.

The process continues with the next decision vertex (i.e. a forward approach) until node "F" is reached. This is done for each DDS, and in general, decision node $d_{jk(j)*}$ is selected so that:

$$p_{jk(j)*} = \underset{k(j)}{\text{Max}}\, p_{jk(j)}$$

Notice that during this process, a decision vertex that is common to two or more DDS may be eliminated from one DDS; however, one of its decision nodes might be chosen while repeating this selection process for a different DDS (see the example in Appendix C. Decision vertex 6 fits this description).

Once this process is finished, all probabilities are eliminated, and the outcome is a standard network, referred to as the "most probable network." The time and cost of this network are evaluated in the regular fashion, yielding the most probable time and cost. An example showing the above procedure is shown in Appendix C.

It should be noted that this procedure does not yield the maximum probability path(s) from "S" to "F," but the most probable path(s) from "S" to "F." The difference between the two is that the latter seeks the *alternative* with the highest probability at each decision vertex, whereas the first seeks the *path* with the highest probability, and this does not necessarily correspond to selecting the most probable set of decision nodes. The reason for selecting the approach presented here is that, since in this case each decision node has a probability of being selected, it is more realistic to assume that the most probable route will be followed.

Concluding Remarks

The four methods presented here should not be viewed as mutually exclusive, but rather complementing each other. It is anticipated that the first method--project time and cost extremes--will usually be the first to be applied. If the Region of Possible Outcomes turns out to be narrow enough, in many cases a decision can be made upon this basis only. If this region leaves some doubt, the other three methods can be used for obtaining additional input to the decision making process.

It should be noted that the "Region of Possible Outcomes" can be be used also for cases of uncertainty--when no probability distribution is available.

CHAPTER VI

THE CASE OF RISK WITH STOCHASTIC OUTCOMES

Introduction

This case is a different extension of the case of certainty. In this case each decision node is followed by a stochastic vertex with a finite number of outcomes and known probabilities. Once a decision node is selected, the outcome of the stochastic vertex cannot be controlled by the decision maker. During the planning phase, a complete selection among alternatives is not possible anymore. Instead, a strategy[1] can be determined, based upon some desired criteria. As a consequence, neither final elimination of alternatives is possible during the planning phase, nor is reduction of the decision network into a standard network.

In spite of this, a decision has to be made during the planning phase whether to proceed with the project or not. To assist in this decision, seven inputs to this process are developed, as follows:

1) Minimum expected project time and its expected cost.

2) Minimum expected project cost and its expected time.

3) Region of possible outcomes.

4) Range of outcomes for the minimum expected time strategy

---

[1] When dealing with stochastic processes it is preferable to replace the term "policy" by the term "strategy." It is now a question of determining a set of optimal decisions to meet every possible outcome.

and the minimum expected cost strategy.

5) Most probable outcome for the minimum expected time strategy and the minimum expected cost strategy.

6) Evaluation of risk in an optimal strategy.

7) Simulation of the stochastic outcomes network.

These inputs complement each other, and therefore should provide a broad basis for decision making. Furthermore, some of these methods can be used when the decision has to be made in the face of uncertainty--i.e. when no probabilities are available.

Minimum Expected Project Time

The procedure of evaluating the minimum expected project time is an extension of the procedure described for the case of certainty. Instead of evaluating the policy U^* as in the case of certainty, a strategy is determined. The solution procedure is composed of three steps, as follows:

1) Network decomposition.

2) Evaluation of minimum expected project time.

3) Evaluation of the expected project cost associated with the minimum expected project time.

Network Decomposition

The decomposition process for this case is a slight modification of the one described in Chapter IV.

The first level labeling algorithm described for the case of certainty remains unchanged. The outcome of this algorithm is the various DDS's associated with the network.

The second level labeling algorithm of Chapter IV has to be modified.

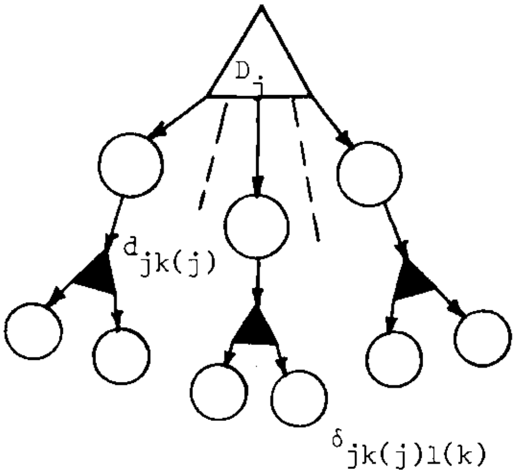


Figure 44. Elements of a Stochastic Outcomes Network

Steps 1-7 remain essentially unchanged, where the stochastic vertices and the outcome nodes are treated like any other node during the labeling process. Step 8 is changed to read as follows:

Step 8. Evaluate the longest path (CP) from each $\delta_{jk(j)l(k)}$ associated with the ith stage to all $d_{jk(j)}$ associated with the (i+1)th stage, provided there is at least one path between the two. In doing so, the duration of $d_{jk(j)}$ associated with the ith stage is added to each of its succeeding $\delta_{jk(j)l(k)}$. When evaluating CP for outcome nodes of the first decision vertex, ES for each outcome node is the one of step 2. For all other cases, ES = 0.

This last step evaluates $t_{jk(j)l(k)}$, the time return associated with outcome node $\delta_{jk(j)l(k)}$. Thus, the outcome of the decomposition process is the required elements for the stochastic DP model of this DDS.

Minimum Expected Project Time

Stochastic discrete dynamic programming is the solution procedure used to find the minimum expected project time. The procedure is best described using the illustration of Fig. 45.

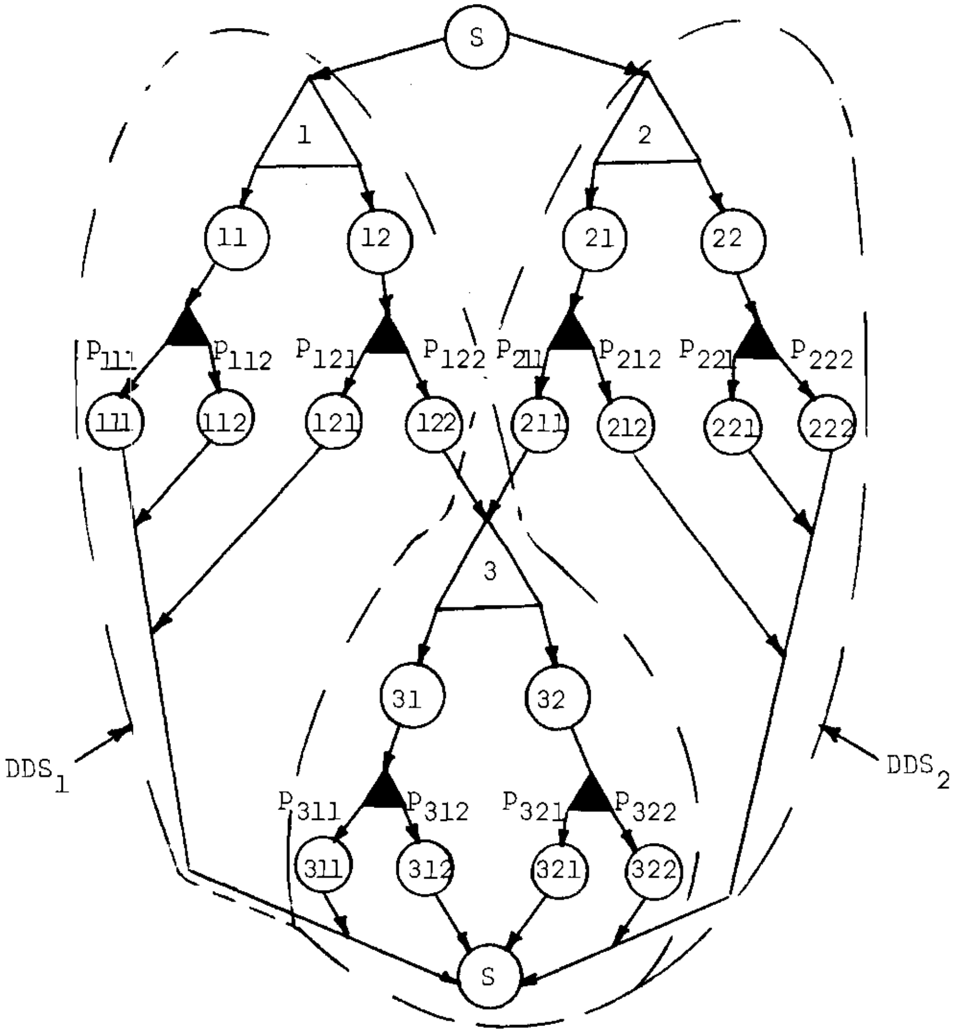


Figure 45. Stochastic Outcomes Network with a Common Decision Vertex

Associated with outcome node $\delta_{jk(j)l(k)}$ of a decision node $d_{jk(j)}$ there is a probability $p_{jk(j)l(k)}$ and a return $t_{jk(j)l(k)}$, which is the value of the critical path obtained during the decomposition process. Following the approach suggested by Nemhauser (51) for a stochastic system, the probability distribution associated with each stochastic vertex is described by a random variable that effects the stage return and transformation. To avoid cumbersome notations, the notation used to denote a stochastic vertex and an outcome node will be used to denote this random variable. Thus:

Let:

$\Delta^i_{jk(j)}$	be the random variable associated with decision node $d_{jk(j)}$ at stage i.
$\delta^i_{jk(j)l(k)}$	be an element of the random variable (associated with outcome node $\delta_{jk(j)l(k)}$ at stage i).
Δ^i	be the set of random variables of stage i.

Then:

$$\Delta^i_{jk(j)} \subset \Delta^i$$

$$\delta^i_{jk(j)l(k)} \in \Delta^i_{jk(j)}$$

The random variables $\Delta^i_{jk(j)}$ are assumed to be independently distributed with probability distribution

$$P(\Delta^i_{jk(j)} = \delta^i_{jk(j)l(k)}) = p^i_{jk(j)l(k)}$$

Note that

$$\sum_{l(k)} P_{jk(j)l(k)} = 1 \qquad \forall\ jk(j)$$

Modifying the notations used for the case of certainty to include the random variable Δ^i, the minimum expected time for each DDS is obtained as follows:

$$\bar{f}_{i(t)}(X_i) = \underset{Y_i}{\text{Min}}\ \bar{Q}_i(X_i,Y_i) \qquad i=1\ldots n$$

$$\bar{Q}_i(X_i,Y_i) = Q_i(X_i,Y_i,\Delta^i)\underline{\underline{P}}^i$$

and:

$$Q_i(X_i,Y_i,\Delta^i) = T_i(X_i,Y_i,\Delta^i) + f_{(i-1)t}(X_{i-1}) \qquad m\times n \text{ matrix}$$

$$\bar{Q}_1(X_1,Y_1) = T_1(X_1,Y_1,\Delta^1)\underline{\underline{P}}^1$$

where:

$\bar{f}_{i(t)}(X_i)$	denotes the minimum expected time at stage i as a function of the input variable.
$\bar{Q}_i(X_i,Y_i)$	denotes the m×r i-stage expected time matrix.
$T_i(X_i,Y_i,\Delta^i)$	is the m×n time return matrix of stage i, composed of $t_{jk(j)l(k)}$.
$\underline{\underline{P}}^i$	is an n×r stochastic matrix of stage i.

$$Y_i = \{y_{ij}\}$$

$$y_{i1} = d_{jk(j)}$$

and $d_{jk(j)}$ is followed by

$$\delta^{i}_{jk(j)l(k)}$$

also:

$$X_{i-1} = Y_i = \{y_{ij}\} \supset \{\delta^{i}_{jk(j)l(k)}\}$$

For stage 2 of DDS_1 of the network illustrated in Fig. 45, $\underline{\underline{P}}^2$ and Q_2 are as follows:

$$Q_2 = (X_2, Y_2, \Delta^2) = S \begin{array}{cccc} \multicolumn{2}{c}{11} & \multicolumn{2}{c}{12} \\ 111 & 112 & 121 & 122 \\ \end{array}$$
$$\left[\begin{array}{cccc} t_{111} & t_{112} & t_{121} & t_{122} + f_{1(t)}(122) \end{array}\right] \quad \underline{\underline{P}}^2 = \begin{array}{c|cc} & 11 & 12 \\ \hline 111 & p_{111} & 0 \\ 112 & p_{112} & 0 \\ 121 & 0 & p_{121} \\ 122 & 0 & p_{122} \end{array}$$

and

$$\bar{Q}_2(X_2, Y_2) = S \begin{array}{cc} 11 & 12 \\ [\bar{t}_{11} & \bar{t}_{12}] \end{array}$$

Let $\bar{f}^{i}_{n(t)}(X_n)$ be the minimum expected time for DDS_i, which is equal to

$$E[T^{i}]^{*} = \bar{f}^{i}_{n(t)}(X_n)$$

Then, the minimum expected time for the whole project is:

$$E[T]^* = \underset{i}{\text{Max}}(E[T^i]^*) \qquad i=0,1...$$

where $E[T^o]^* = T^o$ is the critical path of CDDS.

Associated with the optimal solution is the minimum expected time strategy $\bar{U}^*$ which gives the set of optimal decisions to meet every possible outcome. The optimal strategy for DDS_i will be denoted by $\bar{U}_i^*$. It is convenient to describe $\bar{U}^*$ by a "strategy tree," as shown in a later section. Again, for a common decision vertex, Theorem 2 of Chapter IV holds true also here.

Optimistic and Pessimistic Minimum Expected Time. The problem of two estimates of the minimum expected time--optimistic and pessimistic--encountered in the previous chapter, may arise in some cases here, too. Consider, for example, the project represented by Fig. 45. There are two DDS, and solving for each DDS can yield the following results for the first decision nodes to be selected.

1) {11,21} 2) {12,22} 3) {11,22} 4) {12,21}

The fourth case is different than the first three. For the first three the solution procedure discussed yields the lower bound of the minimum expected time provided T^o is not the dominating value (see Theorem 4). The fourth case creates the problem of optimistic or pessimistic expected time. The procedure previously presented will yield the lower bound of the optimistic expected time. In order to obtain the pessimistic expected time, the problem has to be reworked in a similar way to that described in Chapter V. For the same reasons discussed there, it is not recommended that this approach be taken. Instead, the expected value can be obtained by simulation, as discussed later in this chapter.

Expected Cost of the Minimum Expected Time. Once the strategy yielding the minimum expected time is known, it is possible to evaluate the expected cost associated with this strategy.

The first step in obtaining the expected cost is generating the "Partially Reduced Network" (PRN) defined as follows:

Definition. Partially reduced network (PRN), $G(J^{(p)},A^{(p)})$ of a network $G(J,A)$ is a connected network such that $J^{(p)} \subset J$, $A^{(p)} \subset A$, and:

1) Nodes S and F are elements of this network.

2) All $d^{*}_{jk(j)}$ -- the decision nodes which are elements of the minimum expected time strategy are elements of this network.

3) All nodes J_k such that $\{d^{*}_{jk(j)}\} < J_k$ are elements of this network, except as in (4).

4) All decision vertices are eliminated.

The network reduction is performed using the procedure discussed in Chapter III. Thus, suppose that for the network illustrated in Fig. 45 one gets

$$d^{*}_{jk(j)} = \{11,21,31\}$$

Then, the PRN is as shown in Fig. 46.

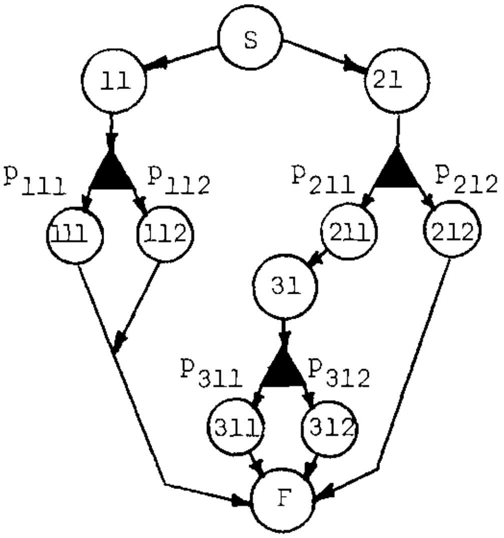


Figure 46. PRN

Once the PRN is obtained, it can be viewed as a network equivalent to the one described in the previous chapter, where each stochastic vertex is now equivalent to a stochastic decision vertex before. Associated with each outcome node $\delta_{jk(j)l(k)}$ there is a cost $c_{jk(j)l(k)}$ (the method of obtaining this cost is described in the minimum expected cost procedure). Thus, finding the expected cost of the minimum expected time is equivalent to the procedure described for the expected cost of a stochastic decisions network, described in Chapter V.

When a non-serial DP model exists, some modifications are required in order to evaluate the minimum expected time and its associated cost. An example for doing this is shown in Appendix D.

Strategy Tree. For large networks, it is convenient to describe the optimal strategy $\bar{U}^*$ using a strategy tree. This is in a sense, a PRN where only the decision nodes and the outcome nodes of the optimal strategy are shown. The strategy tree of the PRN of Fig. 46 is illustrated in Fig. 47

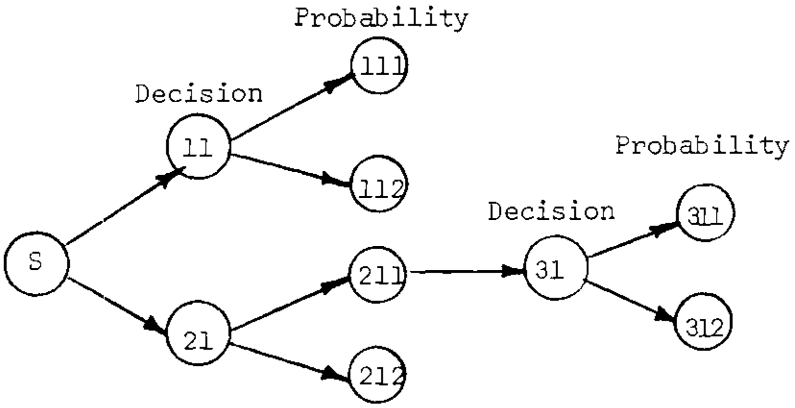


Figure 47. Strategy Tree

Minimum Expected Project Cost

The same comments that were made while discussing the minimum expected time problem are true also for this case. Again, the solution procedure is composed of three steps as follows:

1) Network decomposition.

2) Evaluation of minimum expected project cost.

3) Evaluation of the expected project time associated with the minimum expected project cost.

Network Decomposition

The decomposition procedure for the minimum expected cost is similar to that for the minimum expected time, and is a modification of the minimum cost decomposition procedure of the case of certainty (Chapter IV). Specifically referring to the latter, the decomposition is applied to the RCN, and step 8 is replaced by the following.

Step 8. Starting with the first decision vertex of each DDS, label all successors of each outcome node $\delta_{jk(j)l(k)}$ with $\{jk(j)l(k)\}$ in the manner described in steps 3 and 4 of the second-level labeling algorithm of Chapter IV. If an activity node has two or more labels associated with $\delta_{jk(j)l(k)}$ of *different* D_j, the label associated with the $\delta_{jk(j)l(k)}$ connected with a smaller stage number is dominating, and the rest of the labels should be ignored. If the labels are associated with $\delta_{jk(j)l(k)}$ of the same D_j, or the same stage number, they all remain.

The cost return $c_{jk(j)l(k)}$ associated with each outcome node is:

$$c_{jk(j)l(k)} = \sum_{\text{all } m_i \text{ labeled } \{jk(j)l(k)\}} c_i$$

and $c_{jk(j)l(k)}$ includes the cost of $d_{jk(j)}$.

The outcome of the decomposition process is the required elements for the stochastic DP model of each DDS.

Evaluation of Minimum Expected Project Cost

The procedure for evaluating the minimum expected cost for each DDS is essentially the same as that of the minimum expected time, with some notational modifications. Thus,

Let:

$\bar{f}_{i(c)}(X_i)$	be the minimum expected cost at stage i as a function of the input variable.
$\bar{R}_i(X_i,Y_i)$	be the m×r i-stage expected cost matrix, and

$C_i(X_i,Y_i,\Delta^i)$ be the m×n cost return matrix of stage i, composed of jk(j)l(k).

Then:

$$\bar{f}_{i(c)}(X_i) = \underset{Y_i}{\text{Min}}\ \bar{R}(X_i,Y_i)$$

$$\bar{R}_i(X_i,Y_i) = R_i(X_i,Y_i,\Delta^i)\underline{\underline{P}}^i$$

$$R_i(X_i,Y_i,\Delta^i) = C_i(X_i,Y_i,\Delta^i) + \bar{f}_{(i-1)(c)}(X_{i-1}) \quad \text{an m×n matrix}$$

Let $\bar{f}^i_{n(c)}(X_n)$ be the minimum expected cost for DDS_i, which is equivalent to:

$$E[C^i]^* = \bar{f}^i_{n(c)}(X_n)$$

The minimum expected cost of the total project is:

$$E[C]^* = \sum_i E[C^i]^* + C_{PN}$$

where C_{PN} is the cost of the permanent nodes.

Associated with the optimal solution is the minimum expected cost strategy $\bar{V}^*$. The optimal strategy for DDS_i will be denoted by $\bar{V}^*_i$. As before, it is convenient to describe this strategy by a "strategy tree." Also, for a common decision vertex, Theorem 2 of Chapter IV is valid here too.

Optimistic and Pessimistic Minimum Expected Cost. When a decision vertex is common to two or more DDS, the problem of two estimates of the minimum expected cost can arise here too. (Note that the problem of a lower bound does not exist here.)

Referring to the network illustrated in Fig. 45, in the same way as for the minimum expected time, the problem of optimistic or pessimistic estimate is encountered only for the fourth case, i.e. when the solution yields alternatives 12 and 21. The procedure outlined above gives the *pessimistic* estimate of the minimum expected cost. To find the optimistic estimate, a similar procedure to the one described in the previous chapter has to be followed. Thus, the optimistic minimum expected cost of the project of Fig. 45, assuming that decision node 31 is the optimal strategy if D_3 is realized, is given by:

$$E[C]^{*} = E[C^{1}]^{*} + E[C^{2}]^{*} - p_{122}p_{211}\left(\bar{f}^{1}_{1(c)}(122)\right)$$

Note that

$$\bar{f}^{1}_{1(c)}(122) = \bar{f}^{2}_{1(c)}(211)$$

For the general case, the procedure outlined for the expected cost in Chapter V has to be followed where $\bar{c}_{n^*}(k)$ is substituted with the proper value of $\bar{f}_{i(c)}$. The outcome of this process will yield the value of $E[C_{CN}]^{*}_{j}$--the minimum expected cost of the jth first common decision vertex, that should be deducted. Thus, for the general case, the optimistic estimate is given by:

$$E[C]^* = \sum_i E[C^i]^* - \sum_j E[C_{CN}]_j^* + C_{PN}$$

It is convenient to obtain the PRN before the optimistic estimate is evaluated. The two estimates will yield different values, but will lead to the same strategy. The following theorem proves this assertion.

Theorem 5. Solving for the optimistic minimum expected cost or pessimistic minimum expected cost yields the same optimal strategy $\bar{V}^*$.

Proof. Referring to Fig. 45, suppose that the optimal strategy resulting from solving for the pessimistic minimum expected cost is:

$$\bar{V}^* = \{12,21,31\}$$

(Recall that by Theorem 2, if "31" is part of the optimal strategy obtained by solving for DDS_1, it is going to be part of the optimal strategy when solving for DDS_2.)

Let:

$E[c_{jk(j)}]$ - expected cost associated with decision node $d_{jk(j)}$.

Then, if the optimum strategy turned out to be as stated above, this would mean that:

(1) $E[c_{12}] + E[c_{31}] < E[c_{11}]$ $(DDS)_1$

(2) $E[c_{21}] + E[c_{31}] < E[c_{22}]$ $(DDS)_2$

Since all values are positive, then:

$$\left(E[c_{12}] + E[c_{31}]\right) + \left(E[c_{21}] + E[c_{31}]\right) < E[c_{11}] + E[c_{22}]$$

The left side of the above inequality is the pessimistic minimum expected cost.

Let:

$E[C]^*_{op}$ - optimistic minimum expected cost.

$E[C]^*_{ps}$ - pessimistic minimum expected cost.

Obviously:

$$E[C]^*_{op} < E[C]^*_{ps}$$

and therefore:

$$E[C]^*_{op} < E[C_{11}] + E[C_{22}]$$

For a different strategy, the following three alternatives should be considered:

(a) {11,22,31} (b) {11,21,31} (c) {12,22,31}

Case (a) can be immediately ruled out, as $E[C]^*_{ps}$, $E[C]^*_{op}$, following the original strategy, are better.

For case (b),

$$E[C] = E[c_{11}] + \left(E[c_{21}] + E[c_{31}]\right)$$

From Equation (1) above, since all values are positive, one can form the following inequality.

$$\left(E[c_{12}] + E[c_{31}]\right) + \left(E[c_{21}] + E[c_{31}]\right) < E[c_{11}] + \left(E[c_{21}] + E[c_{31}]\right)$$

or

$$E[C]^*_{ps} < E[C]$$

and obviously:

$$E[C]^*_{op} < E[C]$$

The same results can be shown for case (c) above.

Viewing the network of Fig. 45 as a section of a bigger network with more than two DDS and one common decision vertex, and by repeatedly applying the approach shown above, it can be shown that the same result holds true for the general case. This proves the theorem.

Expected Time of the Minimum Expected Project Cost. The first step in evaluating the expected time of the minimum expected cost is obtaining the PRN associated with the optimal strategy $\bar{V}^*$. Once this is done, the procedure outlined for the expected time problem in the previous chapter can be used, where the stochastic vertex is now the equivalent of a stochastic decision vertex of the previous case, and the return associated with each outcome node $\delta_{jk(j)l(k)}$ is the time return

$t_{jk(j)l(k)}$ evaluated in step 8 of the decomposition procedure for the minimum expected time problem. Notice that, again, the value obtained is the lower bound of the expected time. For obtaining the exact value simulation is recommended, as discussed in a later section.

Some modifications are required when nonserial DP models arise. An example is shown in Appendix D.

Region of Possible Outcomes

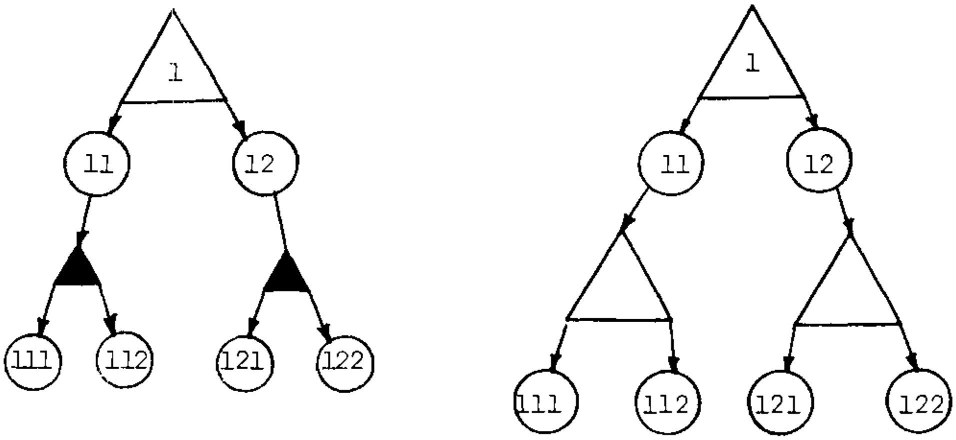
As was indicated in Chapter V, the extreme possible values of the project time and cost are an important input to the decision making process. This argument also holds true here. However, two types of "Region of Possible Outcomes" can be developed: the first is as before, for the whole project. The second is the region of possible outcomes for each "Opening Policy," where an opening policy H is defined as the set of decision nodes selected for the *first* decision vertex of each DDS_i. Referring to the stochastic outcomes network of Fig. 45, it is possible to have four opening policies as follows:

$$H_1 = \{11,21\} \quad H_2 = \{11,22\} \quad H_3 = \{12,21\} \quad H_4 = \{12,22\}$$

Since an opening policy commits the decision maker to a certain course of action for the rest of the project, it is important for him to know what might be the consequences of such a commitment.

Region of Possible Outcome for the Whole Project

In order to establish this region, the probability distribution associated with each stochastic vertex is ignored, and each stochastic vertex is regarded as a decision vertex. By doing so, the stochastic outcomes network is transformed into a decision network, as illustrated in Fig. 48.



(a) Original Network (b) Transformed Network

Figure 48. Network Transformation

Once this is done, the problem is treated in the same manner as that of the case of certainty, where four problems are to be solved:

Minimum Time Problem
Minimum Cost Problem
Maximum Time Problem
Maximum Cost Problem

The procedure developed for the case of certainty is used to solve these four problems, as was done for the case of risk with stochastic decisions.

In addition to the time and cost extremes, this procedure gives the information about the sequence of decisions and stochastic outcomes that will lead to the extreme values, as follows:

(1) Extreme Time Values. The DDS that yields the extreme value has to be examined. First the optimal set of decision nodes of the transformed network is transformed back into decision nodes and stochastic outcomes. The same procedure is applied to the rest of the DDS, realizing that a change of strategy there may still yield the same extreme value. The result yields the chain of events giving the extreme value, and can be viewed as a curtailed strategy tree.

(2) Extreme Cost Values. The process is the same as the preceding one.

Region of Possible Outcomes for an Opening Policy

The procedure is essentially the same as the previous one, with the modification that all nodes succeeding the decision nodes not selected for the first decision vertex are eliminated according to the network reduction procedure of Chapter III.

The above two procedures have been applied to the example of Appendix D.

Range of Outcomes for the Optimal Strategies

One of the common criticisms of the expected value as a criterion of choice is the fact that it does not consider the extremes which might be more important to the decision maker than the expected value. The discussion that follows corrects this deficiency.

Once the minimum expected time or cost are found with their respective strategies, the proper PRN is formed. From here on, the procedure described in the previous section is applied, solving the four problems discussed there for *each* PRN, yielding the following results:

(a) Time and cost extremes of the minimum *expected time* strategy.

(b) Time and cost extremes of the minimum *expected cost* strategy.

An example is shown in Appendix D.

Most Probable Outcome

To further supplement the expected value criterion, the most probable outcome is evaluated for the minimum expected time and minimum expected cost strategies. In a similar manner to that discussed for the case of risk with stochastic decisions, the most probable outcome is preferred to the maximum probability outcome. This is further amplified by considering Fig. 49, where a portion of the PRN of Fig. 45 is shown, with specific values for the probabilities.

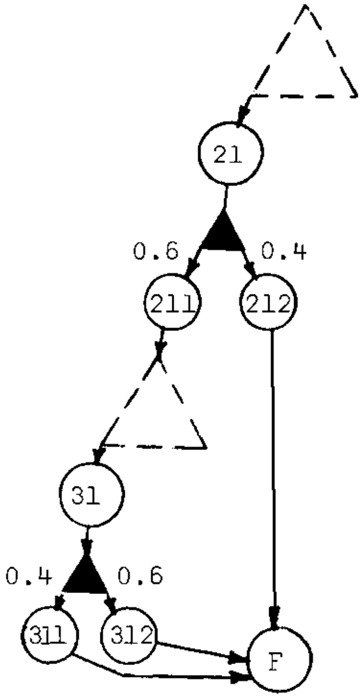


Figure 49. PRN: An Example

Clearly, the maximum probability outcome is 21-212. However, considering the way the process is going to unfold, first decision node 21 is selected, where 211 is more likely to occur, and then decision node 31 is selected, and 312 is more likely to occur. So the most probable path is 21-211, 31-312 with a total probability of 0.36, which is less than 0.4.

To find the most probable outcome, the procedure described in the previous chapter is applicable here, where the PRN of *each* of the optimal strategies is used.

Evaluation of Risk in an Optimal Strategy

The discovery of the optimal strategy may be further supplemented in a case involving risk by knowledge of the probability distribution of time and cost values which may be obtained with this optimal strategy. The method requires enumeration of all path combinations between S and F for the optimal strategy, i.e. using the PRN.

The PRN of Fig. 46 is used as an example to show how this probability distribution can be obtained. Table 1 on the following page is used to evaluate the probability of each path from S to F. For each path, $t_{jk(j)l(k)}$ and $c_{jk(j)l(k)}$ represent the time and cost values, respectively, obtained during the decomposition process.

The sum of all the probabilities is, of course, 1. If a decision node is common to two independent paths from "S" to "F," the probabilities (and expected values) should be adjusted according to the procedure outlined in Chapter V, depending on whether the optimistic or pessimistic values are desired.

Table 1. Path Enumeration

111	112	211	212	311	312	Probability	Time Value	Cost Value
x		x		x		$p_{111}p_{211}p_{311}$	$\text{Max}\{t_{111},(t_{211}+t_{311})\}$	$c_{111}+c_{211}+c_{311}$
x		x			x	$p_{111}p_{211}p_{312}$	$\text{Max}\{t_{111},(t_{211}+t_{312})\}$	$c_{111}+c_{211}+c_{312}$
x			x			$p_{111}p_{212}$	•	•
x			x			$p_{111}p_{211}$	•	•
	x	x		x		$p_{112}p_{211}p_{311}$	•	•
	x	x			x	$p_{112}p_{211}p_{312}$	•	•
	x		x			$p_{112}p_{212}$	•	•
	x		x			$p_{112}p_{212}$	$\text{Max}\{t_{112},t_{212}\}$	$c_{112}+c_{212}$

Once the distribution is obtained, it is easy to evaluate the expected value. The expected cost obtained is the same as the one obtained by the solution procedure described previously, whereas the expected time obtained is the true value which is higher than the lower bound obtained before. An example using the above method is shown in Appendix D.

Obviously, even for a moderate size problem the number of combinations for a complete path enumeration may become so high as to make analytical evaluation impractical. In cases like this it is recommended that simulation be used, as discussed in the next section.

Simulation

Simulation techniques are an attractive approach to stochastic outcomes networks, and supplement the various other approaches discussed before. Specifically, simulation is used to obtain the evaluation of risk in an optimal strategy, and the probability associated with an "opening policy."

Evaluation of Risk in an Optimal Strategy

As has been indicated before, obtaining the time and cost distributions of the optimal strategy requires enumeration of all paths from S to F. The analytical approach can be replaced by a Monte Carlo simulation, applied to the PRN. By using a Monte Carlo selection for each stochastic vertex, only one outcome node is left at each vertex and the rest are eliminated. Once this is done, the PRN can be reduced to a standard network. Thus, for *each* simulation run the stochastic elements are removed, and the time and cost of the standard network can be easily evaluated by finding its critical path and adding the costs of each node of the standard network (including the cost of the permanent nodes).

Repeating this process a large number of times gives the time and cost distributions associated with the optimal strategy that yielded the PRN.

It should be noted that the use of simulation eliminates the problems of optimistic and pessimistic expected values, and that of the lower bound, as for each simulation run the decision network is deterministic.

Since the stochastic decision network of Chapter V is essentially the same as a PRN, the same approach can be used there, too, to obtain project time and cost distributions, thus avoiding the problems associated with the analytical approaches.

Simulation Results. A Monte Carlo simulation, as discussed here, has been applied to the minimum expected time and minimum expected cost strategies of the example of Appendix D. The results of 2000 simulation runs came out to be rather close to the distribution obtained by solving analytically.

As this simulation is relatively simple to apply, it seems to be superior over the analytical approach, even for moderate size networks.

Probability Associated with an Opening Policy

Simulation can be a very powerful technique when applied in connection with the possible opening policies.

In a previous section, the region of possible outcomes of each opening policy was evaluated. This is a deterministic region in E^2 in the sense that no matter what happens after the initial move (represented by the opening policy) is made, the project time and cost values will never be outside this region.

The use of simulation can add another dimension to this region, as follows: Based upon a certain criterion, namely minimum project time or minimum project cost, it is possible to divide the region of possible outcomes into "probability zones," where a probability zone is a subregion of the region of possible outcomes, yielding the following information:

(a) The probability p_{H_i} that the selection of the specific opening policy H_i will yield the optimum result based upon the chosen criterion (either minimum time or minimum cost).

(b) The boundaries of the probability zone.

(c) The probability distribution and expected value of project time and cost if the optimum curtailed strategy of (d) can be followed after the initial move according to H_i has been made.

(d) A curtailed strategy that will lead to the optimum result. This strategy will be denoted by $\bar{\bar{U}}^*$ for the minimum time criterion, and by $\bar{\bar{V}}^*$ for the minimum cost criterion.

A probability zone for opening policy H_i and its related region of possible outcomes is illustrated in Fig. 50. The four parameters of the probability zone need further amplification.

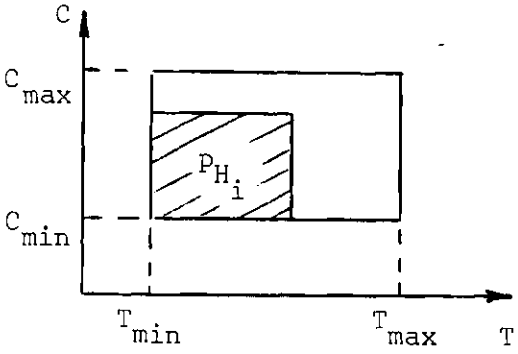


Figure 50. Probability Zone for Policy H_i

The probability P_{H_i} can be interpreted also as the probability that the stochastic outcomes network will unfold in such a way that, starting with H_i will yield the optimum result based upon the chosen criterion. Since the network can unfold in a numerous number of ways, even if selection of H_i will lead to the optimum outcome, the value of this outcome varies according to the various patterns the network can have. That is the reason for the distribution in (c) above. The boundaries of the probability zone are simply the extreme values of this distribution.

The curtailed strategies, $\bar{\bar{U}}^*$ and $\bar{\bar{V}}^*$, give the set of optimal decisions for all outcomes that lead to the optimal result. They are different from $\bar{U}^*$ or $\bar{V}^*$ of the minimum expected value criterion in the sense that they do not show decisions to meet *every* possible outcome. What should be done in case an outcome not represented in $\bar{\bar{U}}^*$ or $\bar{\bar{V}}^*$ occurs is a problem of the execution phase rather than the planning phase; however, no matter what happens, the decision maker still knows that he is bounded by the region of possible outcomes.

Generally speaking, if two opening policies H_i and H_j have the same region of possible outcomes, the one with a higher p_{H_i}, for a narrower probability zone, should be preferred.

The Simulation Process. The mapping of a probability zone into the deterministic region of possible outcomes is done by simulation, and for all practical purposes, simulation is the only way by which it can be done.

The procedure for *each* simulation run is as follows: using Monte Carlo selection process for each stochastic vertex, only one outcome node is left. Then the network reduction procedure of Chapter III is applied, yielding a decision network. The procedure of the case of certainty is applied now, according to the chosen criterion (minimum time or minimum cost) yielding the optimum time and cost values for this simulation run, along with H_i and $\bar{\bar{U}}^*$ or $\bar{\bar{V}}^*$.

Repeating this process for a large number of simulation runs yields the previously described four parameters of the probability zone for each opening policy H_i.

Keeping tally of $\bar{\bar{U}}^*$ or $\bar{\bar{V}}^*$ for all H_i might create some excessive storage requirements and complicate the simulation program. Therefore, it is recommended that curtailed optimal strategy be obtained only for the chosen H_i after this choice has been made (recall that the curtailed optimal strategy is not required for decision making during the planning phase).

Simulation Results. The previous procedure has been applied to the stochastic outcomes network of Fig. 69 in Appendix D, using both a minimum time criterion and a minimum cost criterion with results of 1,000 simulation runs, as shown there. Some interesting conclusions emerge, as follows:

(1) It is possible to identify *dominating opening policies.* Thus, for the minimum time criterion, H_4 is always better than H_7 and H_8. Even more striking are the results for the minimum cost criterion: H_2 and H_6 are the dominating opening policies, and no other opening policy will ever lead to an optimal result.

(2) The minimum expected value strategy is not necessarily the one to be preferred by the decision maker. Referring to the results of the minimum cost simulation, H_6 is the opening policy suggested by the

minimum expected value criterion. However, it seems that a cost conscious decision maker would prefer H_2.

Thus, as can be seen, simulation yields valuable inputs to the decision making process.

Concluding Remarks

All the methods described in this chapter are complementing each other, and in many cases a decision can be made on the basis of part of the inputs suggested, without having to evaluate all of them. A suggested decision making routine utilizing the suggested inputs is indicated below. The process can terminate at any point if the decision maker feels that the amount of information obtained up to such point is sufficient to make a final decision.

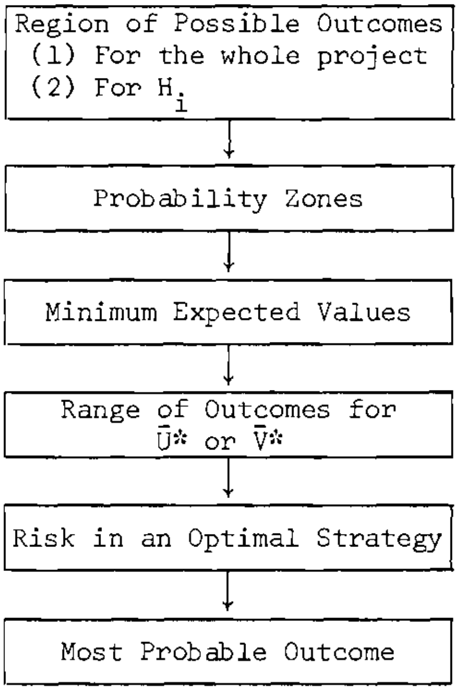


Figure 51. A Decision Making Routine

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The results and conclusions evolving from this study are summarized in the following paragraphs.

A study of the literature of project management and decision networks resulted in the following conclusions:

1. For decision networks, the problem of time cost trade-off, for different sets of decisions, has not been investigated.

2. Solution of decision networks, in terms of optimizing one variable only, has been tried using integer programming or branch and bound techniques, but not dynamic programming.

3. The stochastic cases of decision networks received only slight attention in the literature. All these approaches use expected value as a sole criterion of choice, only one parameter at a time is considered, and a project decision network is rarely treated explicitly.

The research presented herein centered on the planning phase of network based project management, emphasizing the managerial decision making process of evaluating projects. The approach used made no assumptions about the nature of the decision maker, but rather concentrated on generating inputs to the decision making process, enabling the decision maker to reach a decision based upon more than one criterion of choice. These inputs can be summarized as follows:

1. Time cost trade-off for the case of certainty.

2. Region of possible outcomes, expected project time and cost, risk evaluation and the most probable outcome, for the case of risk with stochastic decisions.

3. For the case of risk with stochastic outcomes the following inputs were developed: Region of possible outcomes, probability zones, minimum expected value strategies, range of outcomes for an optimal strategy, risk in an optimal strategy, and the most probable outcome for an optimal strategy.

The research conducted to generate these inputs yields the following conclusions:

1. A decision network can be represented by a dynamic programming model.

2. Dynamic programming is a solution technique that can be applied, with some variations, to all three types of decision networks presented in this research, so as to generate most of the inputs described before. The computation procedure is not complex, and is practical.

3. For the case of certainty, it is possible to introduce budget and time constraints, without any additional effort, after the solution is obtained. The added advantage of the procedure presented herein is that the sensitivity of changing the constraint can be immediately obtained.

4. Dynamic programming proved to be a very efficient approach in locating the efficient set for the case of certainty. One of the claimed

shortcomings of dynamic programming is the fact that a lot of data generated during the solution procedure is not needed for the final result. This additional data turned out to be an important advantage of the dynamic programming method when applied to evaluating the time-cost trade-off.

5. The application of Monte Carlo simulation to the two types of stochastic networks proved to be a very valuable technique. It is especially useful in mapping the probability zones into the region of possible outcomes.

6. It is possible to use the combination of the region of possible outcomes and probability zones to identify dominating opening policies in the case of stochastic outcomes networks.

During the course of this research, at least three important new techniques or concepts were developed, as follows:

1. The Efficient Set algorithm. This algorithm can be applied to any problem where a trade-off among two variables exists and where the problem can be structured as a DP model so that

$$Y_i = X_{i-1}$$

2. The labeling algorithms for formulating the decision networks problems as a DP model.

3. The concept of mapping a probability zone into a deterministic region.

Thus, it can be concluded that this research has developed a methodology for planning with decision networks.

Recommendations

In the course of carrying out this study, several potentially useful areas of research were revealed, as discussed in the remainder of this section.

The immediate extension of this research would be to consider a decision network composed of a mix of activity nodes, decision vertices, stochastic decision vertices and stochastic vertices. The three cases considered in this investigation were "pure cases," consisting of only one type of decision vertex.

Another extension of this research would be the investigation of decision networks during the scheduling and control phases of the project life.

This study did not consider the time value of money. It would be worthwhile to introduce this concept and see how it can be incorporated into the methodology developed here.

For further investigation, it is suggested that the assumption of a single project be relaxed so as to consider the multiproject case. Also, it is suggested that the problem of resource constraints be explored.

The concept of mapping a probability zone into a deterministic zone requires further investigation. Of special interest is the case of overlapping probability zones, when different criteria are used.

Finally, it is felt that the Efficient Set algorithm developed has many applications for other problems. Investigation of such possible applications should prove to be a valuable research effort.

Furthermore, extending this algorithm from the case of two returns to the case of n-returns with trade-off relationships seems to be a worthwhile generalization.

APPENDICES

APPENDIX A

NON-SERIAL DP MODELS

Diverging Branch Models

Two general types of diverging branch models will be considered as follows: type I divergence and type II divergence.

Type I Divergence

Type I divergence has been treated in detail in Chapter IV. All that remains to be shown is how type I divergence can be identified using the second level labeling algorithm. Introducing again the decision network of Fig. 27, and applying to it the second level labeling algorithm, the result is as shown in Fig. 52.

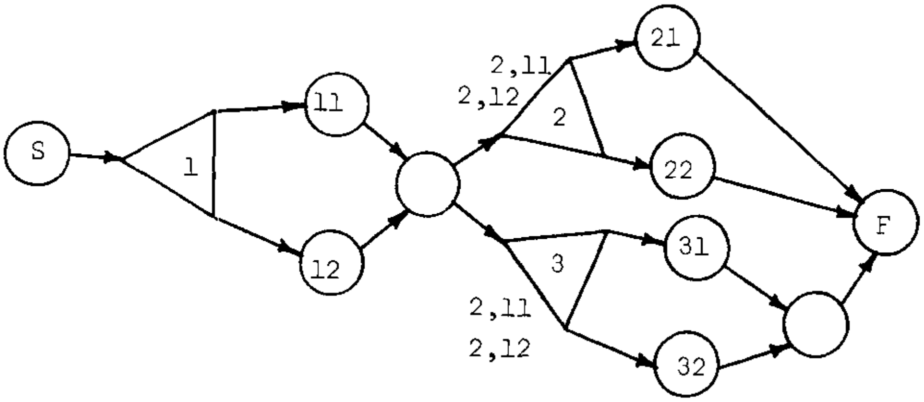


Figure 52. Type I Divergence

Thus, type I divergence exists whenever *all* decision vertices associated with the same stage number of the DP model have exactly the same multiple labels, associated with the same DDS.

Type II Divergence

Type II divergence is defined as follows: Suppose that there are four decision vertices D_1, D_2, D_3, D_4, such that:

$$D_1 < D_2$$

$$D_1 < D_3$$

$$D_1 < D_4$$

and let

$$D_1 = \{d_{11}, d_{12}\}$$

Then, type II divergence is the case when:

$$d_{11} < D_2$$

$$d_{11} < D_3$$

$$d_{12} < D_4$$

For example, consider the following decision network:

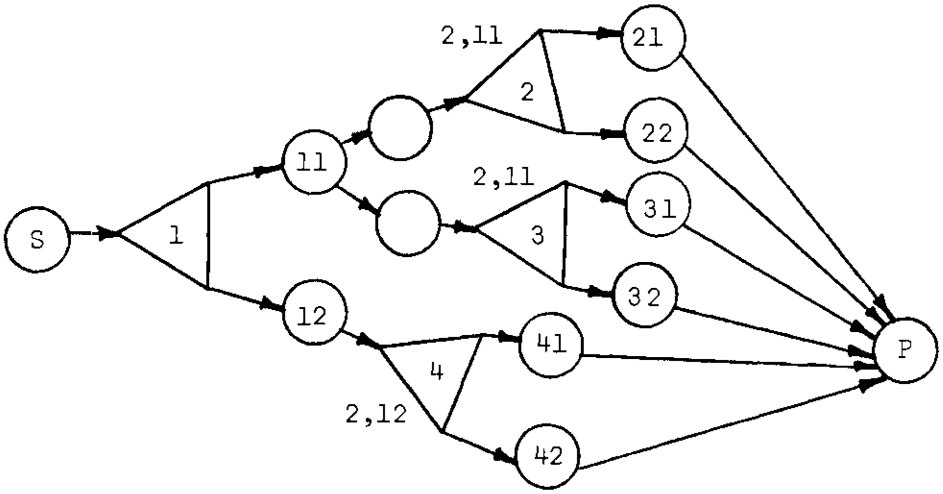


Figure 53. Type II Divergence

The equivalent DP model is as illustrated in Fig. 54.

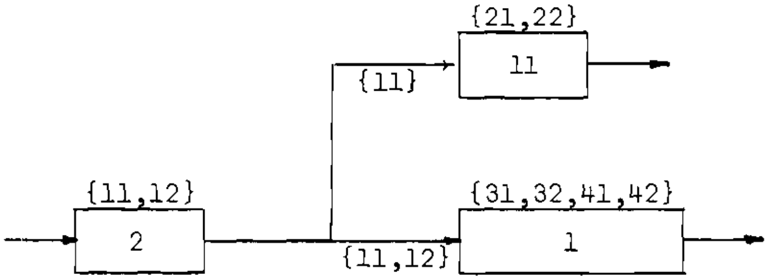


Figure 54. Type II Divergence Equivalent DP Model

Note that it is possible to have a combination of Type I and Type II divergence.

Type II divergence exists whenever *part* of the decision vertices associated with the same stage number have the same labels, as shown in Fig. 53.

Some modification of the solution procedure for type I divergence is required for this case. Referring to Chapter IV (p. 74), now let:

X_{k-1} be the input variable to stage k - 1.

X'_{k-1} be the input variable to stage L1 of the diverging branch, $X'_{k-1} \subset X_{k-1}$.

$\bar{X}_{k-1}$ be the input variable to stage k-1 without the elements of X'_{k-1}.

Then, for the minimum time problem one gets for stage k.

$$Q_k(X_k,Y_k) = T_k(X_k,Y_k) + f_{(k-1)(t)}(\bar{X}_{k-1}) +$$

$$+ \mathrm{Max}[f_{(L1)(t)}(X'_{k-1}),f_{(k-1)(t)}(X'_{k-1})]$$

For the minimum cost problem the equations are essentially the same as before, yielding:

$$R_k(X_k,Y_k) = C_k(X_k,Y_k) + f_{(k-1)(c)}(X_{k-1}) + f_{(L1)(c)}(X'_{k-1})$$

Type I Feedforward

This is an extension of type I divergence, and is treated in a similar way. Fig. 55 illustrates an example of this case, and Fig. 56 shows the equivalent DP model.

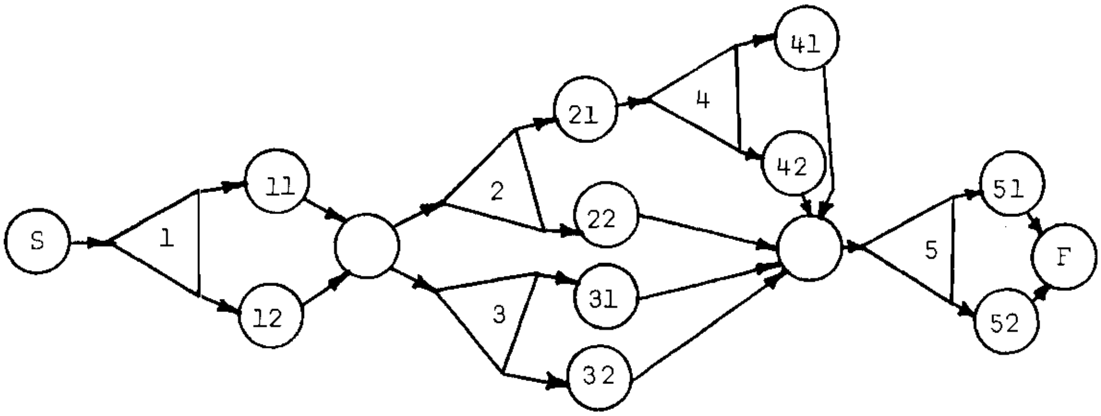


Figure 55. Type I Feedforward

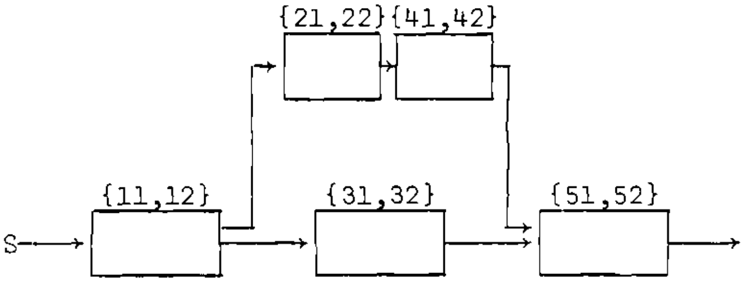


Figure 56. Equivalent DP Model

To solve for the minimum time problem, the DP model of Fig. 56 is transformed into a type I divergence as follows:

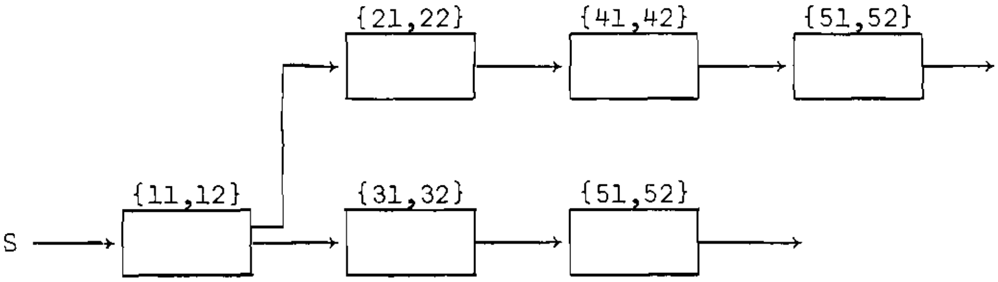


Figure 57. Transformed DP Model

The type I divergence solution procedure for the minimum time problem can be applied now.

The same approach is used for the minimum cost problem, with a different transformed DP model as shown in Fig. 58.

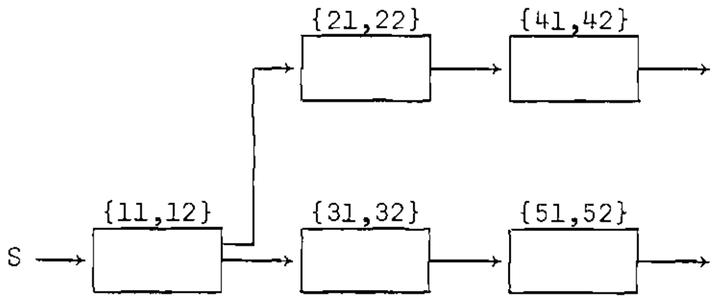


Figure 58. Transformed DP Model-Cost

Type II Feedforward

This is an extension of type II divergence, as illustrated in Fig. 59. The equivalent DP model is shown in Fig. 60. It is transformed into type II divergence in a similar manner to that of type I feedforward. Then, the solution procedure of type II divergence can be applied.

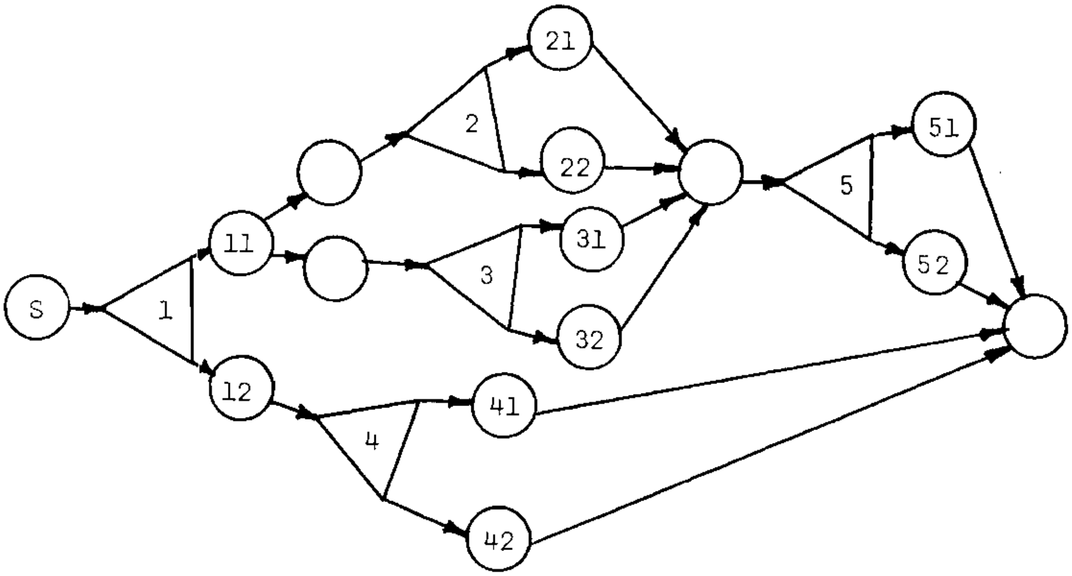


Figure 59. Type II Feedforward

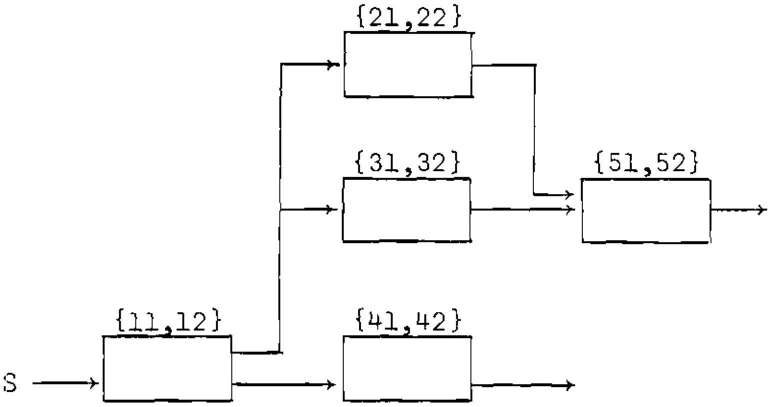


Figure 60. Equivalent DP Model

It is possible to have a mix of the various non-serial models. Cases like this have to be handled along the guide-lines presented above. An example of such a mixed case is given in Appendix D.

APPENDIX B

THE CASE OF CERTAINTY: AN EXAMPLE

General

This appendix includes a complete solution of the decision network of Fig. 17. First, the minimum time problem is solved, then the minimum cost problem, and finally, the time-cost trade-off is developed.

Minimum Time Problem

The decomposition of this decision network was discussed in Chapter IV, yielding DDS_1, DDS_2, DDS_3, DDS_4.

Fig. 61 shows the second level labeling algorithm as applied to DDS_1. The bracketed number adjacent to each label is the time return associated with the proper $d_{jk(j)}$ of the previous stage.

DDS_1 has a type II divergence, as discussed in Appendix A. The divergence occurs at stage 2, and:

$$X_1 = \{11,12\}$$

$$X_1' = \{11\}$$

$$\bar{X}_1 = \{12\}$$

Since the diverging branch is a path from 11 to F, a dummy decision node F is added between 11 and F, to create a stage for the diverging branch. The time return associated with this stage is the critical

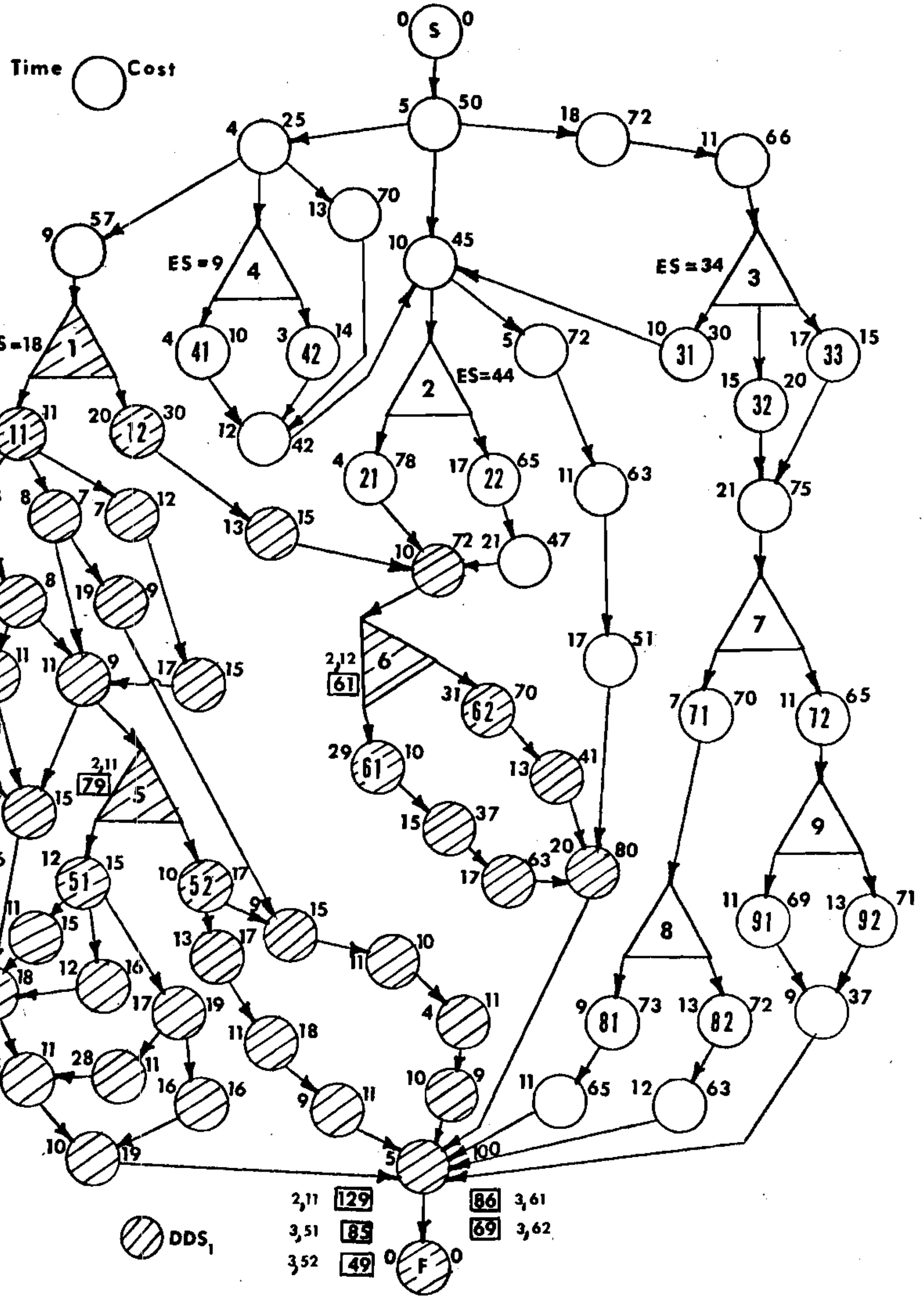


Figure 61. Second Level Labeling Algorithm--Time

path from 11 to F, minus the critical path from 11 to decision vertex 2, i.e.

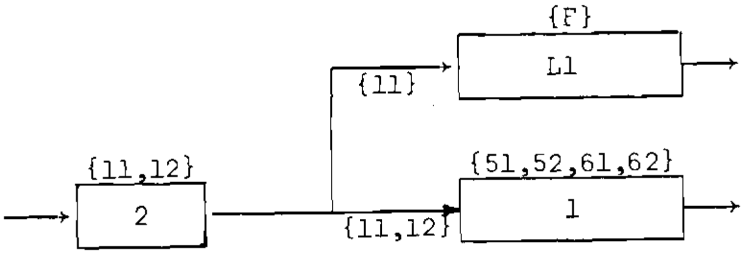
$$129 - 79 = 50$$

By doing so, the computational procedure of type II divergence can be applied.

DDS_3 has also a type II divergence (see Fig. 25), and it is handled in the same manner.

The time returns of all DDS were evaluated in the same way as those of DDS_1, and are shown later along with the DP model of each DDS.

Minimum Time Solution for DDS_1.



$T_i(X_i,Y_i)$

Stage 2			Stage 1					Stage L1	
	11	12		51	52	61	62		F
S	79	61	11	85	49	x	x	11	50
			12	x	x	86	69		

$Q_i(X_i,Y_i)$

Stage 2		
	11	12
S	79 + 50*	61 + 69

Stage 1				
	51	52	61	62
11	85	49	x	x
12	x	x	86	69

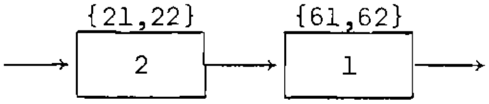
Stage L1	
	F
11	50

*Max(50,49)

$$T^1 = f^1_{2(t)}(S) = 129$$

$$U^*_1 = \{11,52\}$$

Minimum Time Solution for DDS_2.



$T_i(X_i,Y_i)$; $Q_i(X_i,Y_i)$

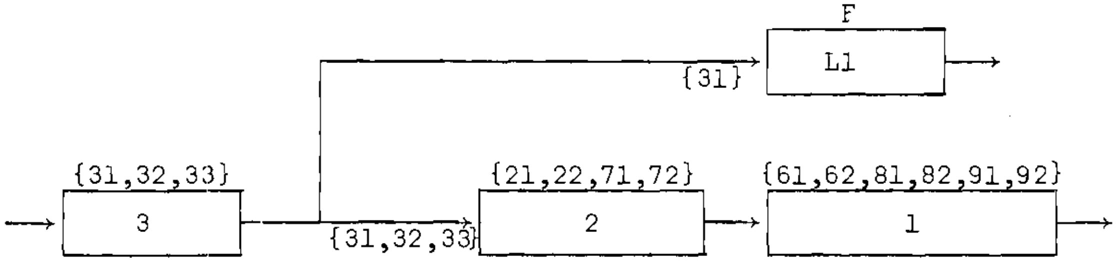
Stage 2		
	21	22
S	58 + 69	92 + 69

Stage 1		
	61	62
21	86	69
21	86	69

$$T^2 = f^2_{2(t)}(S) = 127$$

$$U^*_2 = \{21,62\}$$

Minimum Time Solution for DDS_3.



$T_i(X_i,Y_i);\ Q_i(X_i,Y_i)$

Stage 3			
	31	32	33
S	$+\substack{54\\83^*}$	$\enclose{circle}{+\substack{70\\32}}$	$+\substack{72\\32}$

Stage 2				
	21	22	71	72
31	$\enclose{circle}{+\substack{14\\69}}$	$+\substack{48\\69}$	x	x
32	x	x	$\enclose{circle}{+\substack{7\\25}}$	$+\substack{11\\25}$
33	x	x	$\enclose{circle}{+\substack{7\\25}}$	$+\substack{11\\25}$

Stage 1						
	61	62	81	82	91	92
21	86	$\enclose{circle}{69}$	x	x	x	x
22	86	$\enclose{circle}{69}$	x	x	x	x
71	x	x	$\enclose{circle}{25}$	30	x	x
72	x	x	x	x	$\enclose{circle}{25}$	27

Stage L1	
	F
31	$\enclose{circle}{58}$

*Max(83,58).

$$T^3 = f^3_{3(t)}(S) = 102$$

$$U^*_3 = \{32,71,81\}$$

Minimum Time Solution for DDS_4. Since there is a dominating path by-passing decision vertex 4, this DDS cannot yield any different time value than that of DDS_2. Therefore, the decision node with the smaller cost is selected for vertex 4, leading to:

$$U_4^* = \{41,21,62\}$$

The Critical Path of CDDS. Evaluating the critical path of CDDS (see Fig. 17) yields:

$$T^o = 102$$

Minimum Project Time.

$$T = \underset{i}{\mathrm{Max}}(T^i) = \mathrm{Max}(129,127,102,102) = 129$$

$$U^*_{\mathrm{max\ slack}} = \{11,52,21,62,32,71,81,41\}$$

Minimum Cost Problem

The DP model for this problem is essentially the same as that of the minimum time problem, with the exception that stage returns are values of $c_{jk(j)}$. Fig. 62 shows how the second level labeling algorithm is used to find $c_{jk(j)}$ for each decision node of DDS_1. All activity nodes of Fig. 62 for which the cost has been reduced to zero are permanent nodes.

The diverging branch of DDS_1 is not required for the cost problem, since cost elements are additive. For the more general case this means that the cost of the input to the diverging branch is zero. Same argument holds true for DDS_3.

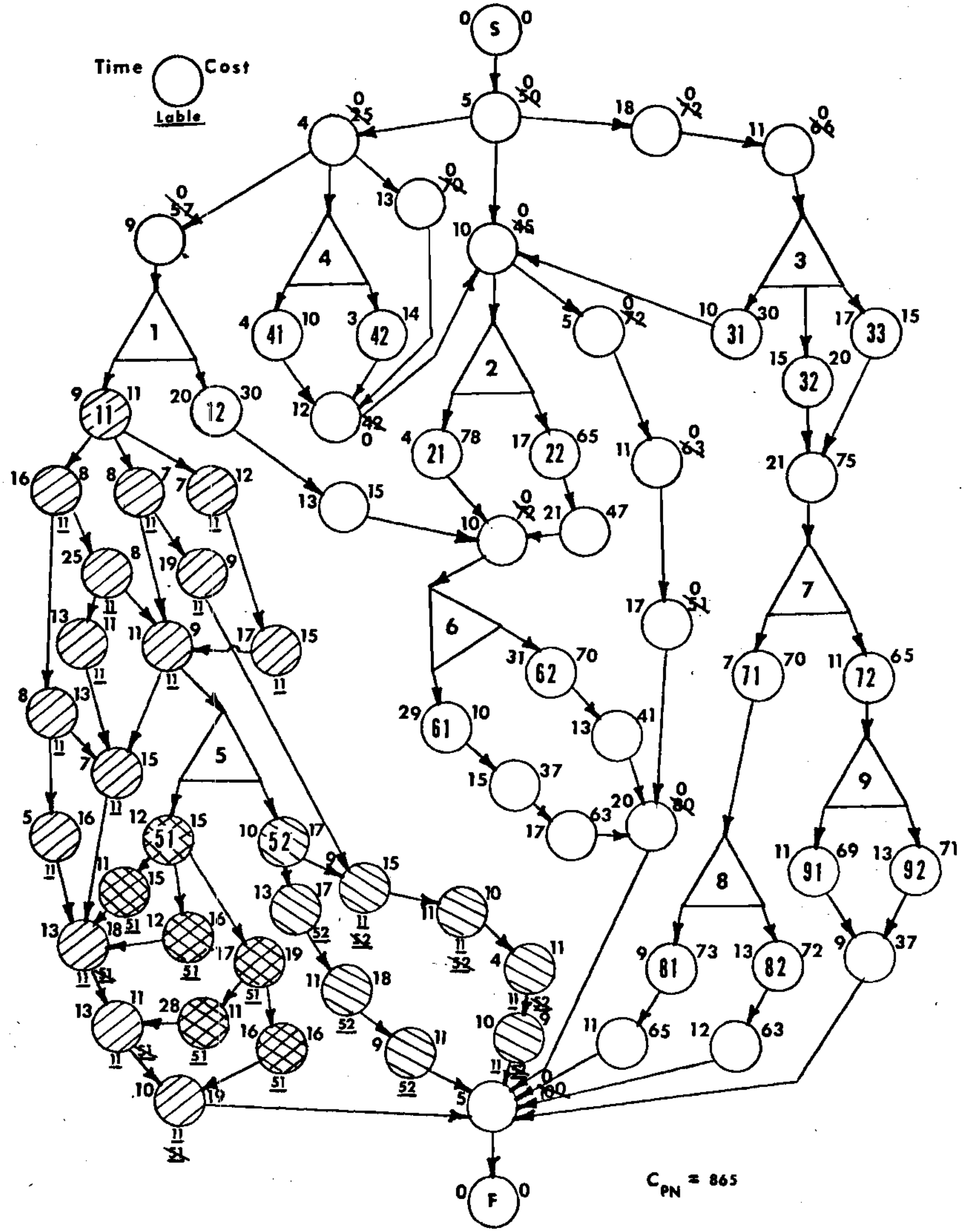
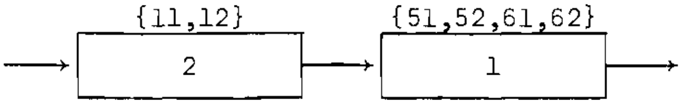


Figure 62. Second Level Labeling Algorithm--Cost

Minimum Cost Solution for DDS_1.



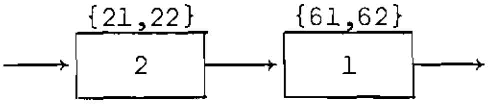
$C_i(X_i,Y_i)$; $R_i(X_i,Y_i)$

	Stage 2			Stage 1			
	11	12		51	52	61	62
S	+227 63	+45 110	11	92	63	x	x
			12	x	x	110	111

$$C^1 = 155$$

$$V_1^* = \{12,61\}$$

Minimum Cost Solution for DDS_2.



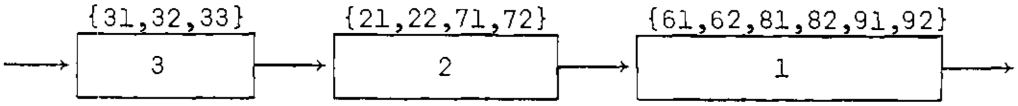
$C_i(X_i,Y_i)$; $R_i(X_i,Y_i)$

	Stage 2			Stage 1	
	21	22		61	62
S	+78 110	+112 110	21	110	111
			22	110	111

$$C^2 = 188$$

$$V_2^* = \{21,61\}$$

Minimum Cost Solution for DDS_3.



$C_i(X_i,Y_1);\ R_i(X_i,Y_i)$

Stage 3	31	32	33
S	+ 30 188	+ 95 171	+ 90 171

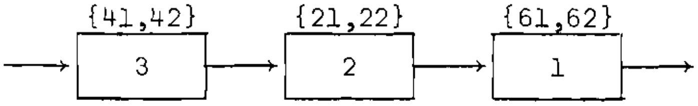
Stage 2	21	22	71	72
31	+ 78 110	+ 112 110	x	x
32	x	x	+ 70 135	+ 65 106
33	x	x	+ 70 135	+ 65 106

Stage 1	61	62	81	82	91	92
21	110	111	x	x	x	x
22	110	111	x	x	x	x
71	x	x	138	135	x	x
72	x	x	x	x	106	108

$$C^3 = 218$$

$$V_3^* = \{31,21,61\}$$

Minimum Cost Solution for DDS_4.



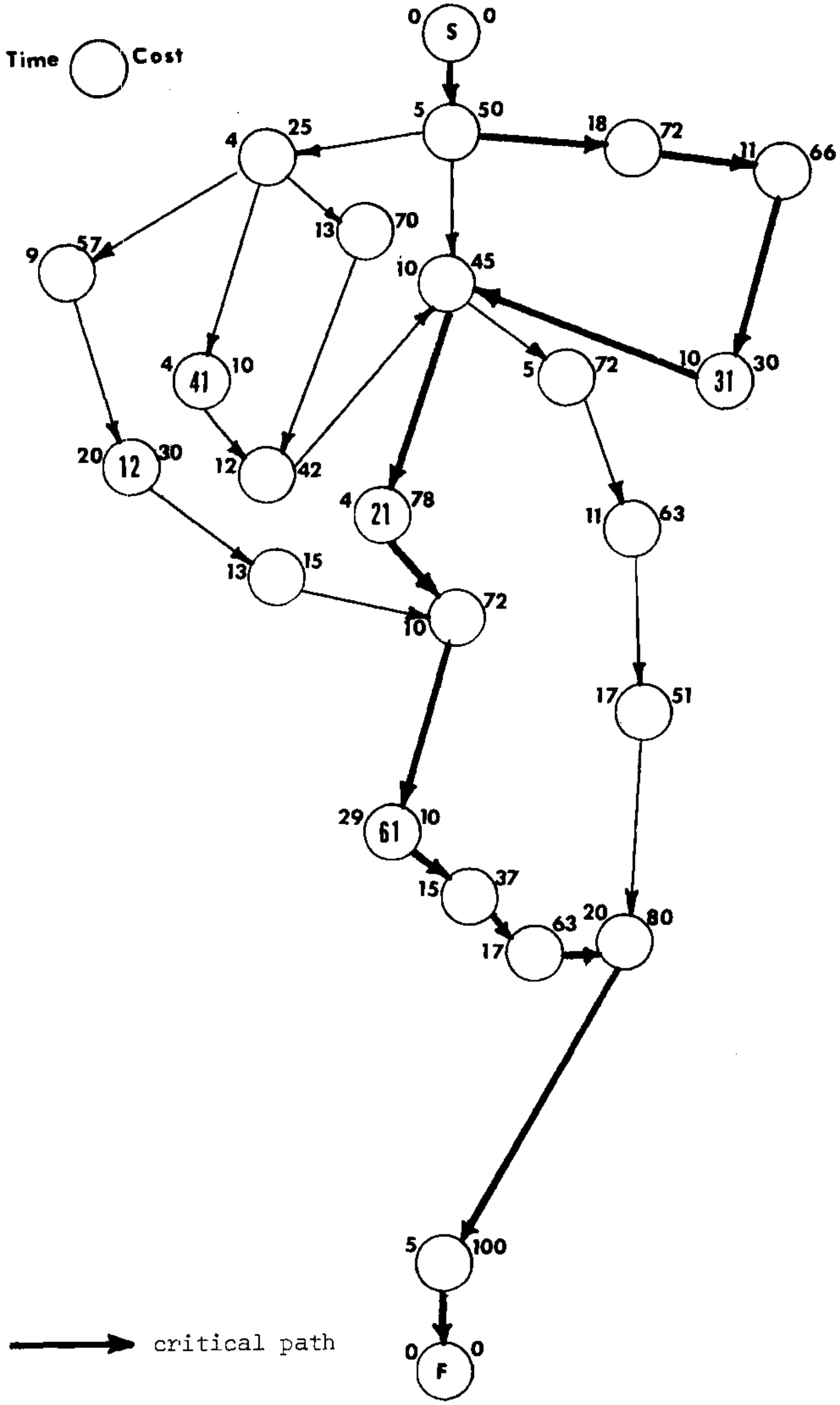


Figure 63. Minimum Cost Standard Network

$C_i(X_i,Y_i)$; $R_i(X_i,Y_i)$

Stage 3			Stage 2			Stage 1		
	41	42		21	22		61	62
S	+ 10 188	+ 14 188	41	+ 78 110	+ 112 110	21	110	111
			42	+ 78 110	+ 112 110	22	110	111

$$C^4 = 198$$

$$V_4^* = \{41,21,61\}$$

Minimum Project Cost.

$$V^* = \{12,61,31,21,41\}$$

$$C^* = \sum_{m_i \epsilon J^*} c_i + C_{PN} = 1138$$

and evaluating the critical path of the minimum cost standard network yields:

$$T_{C^*} = 154$$

Time Cost Trade-Off

Time cost trade-off is first solved separately for each DDS using the methodology developed in Chapter IV, and then the time cost trade-off of the whole project is developed.

Time Cost Trade-Off for DDS_1.

$\Theta_i(X_i,Y_i)$ i=1...n

Stage 2			Stage 1					Stage L1	
	11	12		51	52	61	62		F
S	$\textcircled{+227 \atop 63}$	+45 111	11	92	$\textcircled{63}$	x	x	11	0
			12	x	x	110	$\textcircled{111}$		

$\Delta\Theta_i(X_i,Y_i)$ i=1...n-1

$\Delta Q_i(X_i,Y_i)$ i=1...n-1

Stage 1					Stage L1	
	51	52	61	62		F
11	29 35*	0	x	x	11	0
12	x	x	-1 17	0		

*(85-50)--see remark in step 1 (p.108) of the Efficient Set for a Decision Network.

Efficient Set Tableau

	11	12		51	52	61	62	
$\Theta_n(X_n,Y_n)\longrightarrow$	290	156	11	~~29 35~~	*			
$Q_n(X_n,Y_n)\longrightarrow$	129	130						
	11	12	12			-1 17		
			11					
		155 147	12			-1 17		m=1

$^*\omega_{ijk} > T_{C^*} - T^*$

$(T_1^1,C_1^1) = (129,290)$ $\quad W_1^1 = \{11,52\}$

$(T_2^1,C_2^1) = (130,156)$ $\quad W_2^1 = \{12,62\}$

$(T_3^1,C_3^1) = (147,155)$ $\quad W_3^1 = \{12,61\}$

Time Cost Trade-Off for DDS_2.

$\theta_i(X_i,Y_i)$

Stage 2			Stage 1		
				61	62
			21	110	111
S	+78 111+	112 111			
			22	110	111

$\Delta\theta_i(X_i,Y_i)$

$\Delta Q_i(X_i,Y_i)$

Stage 1		
	61	62
21	-1 17	0
22	-1 17	0

Efficient Set Tableau

21	22		61	62
189	*223	21	-1 17	
127	161			
		22	-1 17	
188		21	-1 17	
144		22	-1 17	

m=1

*$q_{n1k} > T_{C*}$

$(T_1^2,C_1^2) = (127,189)$ $\qquad W_1^2 = \{21,62\}$

$(T_2^2,C_2^2) = (144,188)$ $\qquad W_2^2 = \{21,61\}$

Time Cost Trade-Off for DDS_3.

$\Theta_i(X_i,Y_i)$ i=1...n

	Stage 3		
	31	32	33
S	+30 189	+95 208	+90 208

	Stage 2			
	21	22	71	72
31	+78 111	+112 111	x	x
32	x	x	+70 138	+65 106
33	x	x	+70 138	+65 106

	Stage 1					
	61	62	81	82	91	92
21	110	111	x	x	x	x
22	110	111	x	x	x	x
71	x	x	138	135	x	x
72	x	x	x	x	106	108

$\Delta\Theta_i(X_i,Y_i)$
$\Delta Q_i(X_i,Y_i)$ i=1...n-1

	Stage 2			
	21	22	71	72
31	0	34 34	x	x
32	x	x	0	-34 4
33	x	x	0	-34 4

	Stage 1					
	61	62	81	82	91	92
21	-1 17	0	x	x	x	x
22	-1 17	0	x	x	x	x
71	x	x	0	-3 5	x	x
72	x	x	x	x	0	2 2

Efficient Set Tableau

				Stage 2				Stage 1							
31	32	33		21	22	71	72		61	62	81	82	91	92	
219 137	303 102	298 104	31		34 34	x	x	21	-1 17		x	x	x	x	
			32	x	x		-34 4	22	-1 17		x	x	x	x	
			33	x	x		-34 4	71	x	x	-3 5		x	x	
31	32	33						72	x	x	x	x		~~2~~ ~~2~~	
218 154	269 106	264* 108	31	-1 17	33 51	x	x	21	-1 17		x	x	x	x	
			32	x	x	-3 5	-34 4	22	-1 17		x	x	x	x	m=1
			33	x	x	-3 5	-34 4	71	x	x	-3 5		x	x	
								72	x	x	x	x			
	269 106	264 108	31		~~34~~ ~~34~~	x	x								
			32	x	x		-34 4								m=2
			33	x	x		-34 4								

For all T < T, the one with the lowest cost is selected.

$(T_1^3, C_1^3) = (108,264)$ $W_1^3 = \{33,72,91\}$

$(T_2^3, C_2^3) = (137,219)$ $W_2^3 = \{31,21,62\}$

$(T_3^3, C_3^3) = (154,218)$ $W_3^3 = \{31,21,61\}$

Time cost trade-off for DDS_4 does not have to be evaluated for the same reason that the minimum time of DDS_4 was not evaluated.

Time Cost Trade-Off for the Whole Project. The DDS efficient set table is used for this purpose as shown in Table 2 on the following page.

Thus:

$$U^* = \{11,52,21,62,33,72,91,41\}$$

with $T^* = 129$ and:

$$C_{T^*} = \sum_{\text{All } d^*_{jk(j)}} c_{jk(j)} + C_{PN} = 750 + 865 = 1615$$

Adding $C_{PN} = 865$ to the project costs of Table 2:

$$(T_1,C_1) = (130,1370) \quad \text{with} \quad W_1 = \{21,62,12,41,33,72,91\}$$

$$(T_2,C_2) = (137,1139) \quad \text{with} \quad W_2 = \{21,62,12,41,31\}$$

obviously, the final point is:

$$(T_{C^*},C^*) = (154,1138) \quad \text{with} \quad V^* = \{21,61,12,41,31\}$$

Table 2. DDS Efficient Set Table

		DDS	Common Vertices				DDS_1				DDS_4		DDS_3								
		Vertex	2		6		1		5		4		3			7		8		9	
		DDS	2,3,4		1,2,3,4																
		$t_{jk(j)}$	14 (58)	48 (92)	86	69	79	61	85	49	35	34	54	70	72	7	11	25	30	25	27
		$C_{jk(j)}$	78	112	110	111	227	45	92	63	10	14	30	95	90	70	65	138	135	106	108
		$d_{jk(j)}$	21	22	61	62	11	12	51	52	41	42	31	32	33	71	72	81	82	91	92
	Cost																				
DDS	DDS	TIME																			
~~3~~	~~303~~	~~102~~												~~$\underline{1}$~~		~~$\underline{1}$~~		~~$\underline{1}$~~			
3	264	108													$\underline{1}$		$\underline{1}$			$\underline{1}$	
2,4	189	127	$\underline{1}$			$\underline{1}$					$\underline{1}$										
1	290	129					$\underline{1}$			$\underline{1}$											
	PROJECT																				
1	505	130	1			$\underline{1}$	x	$\underline{1}$		x	1				1		1			1	
3	274	137	$\underline{1}$			$\underline{1}$		1			1		$\underline{1}$		x		x			x	
* ~~2~~		~~144~~	~~$\underline{1}$~~		~~$\underline{1}$~~	~~x~~		~~1~~			~~1~~		~~1~~								
* ~~1~~		~~147~~	~~1~~		~~$\underline{1}$~~			~~$\underline{1}$~~			~~1~~		~~1~~								
3	273	154	$\underline{1}$		$\underline{1}$	x		1			1		$\underline{1}$								

*A path of 154 is created in DDS_3 (see Step 4(b)).

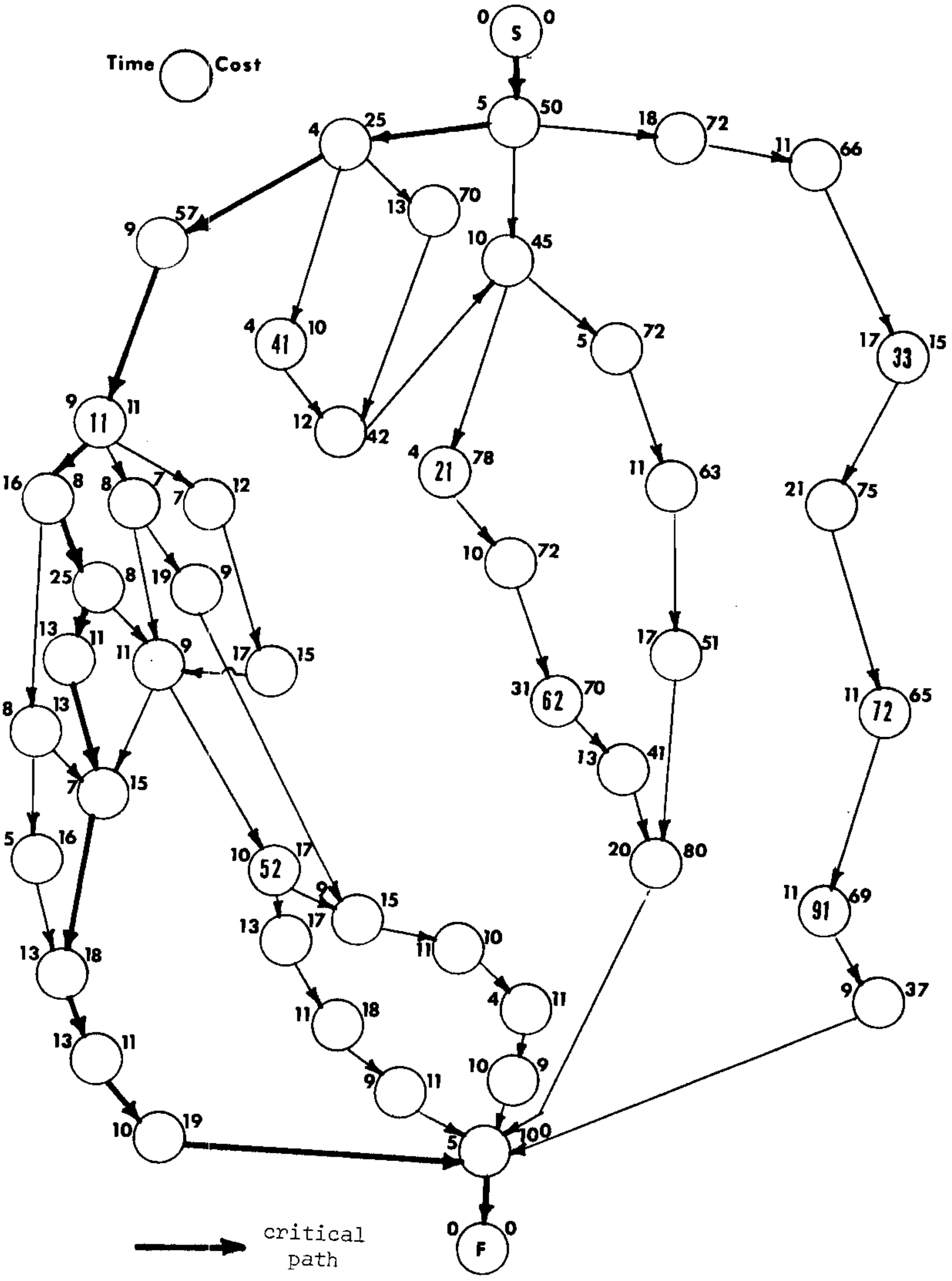


Figure 64. Minimum Time Standard Network

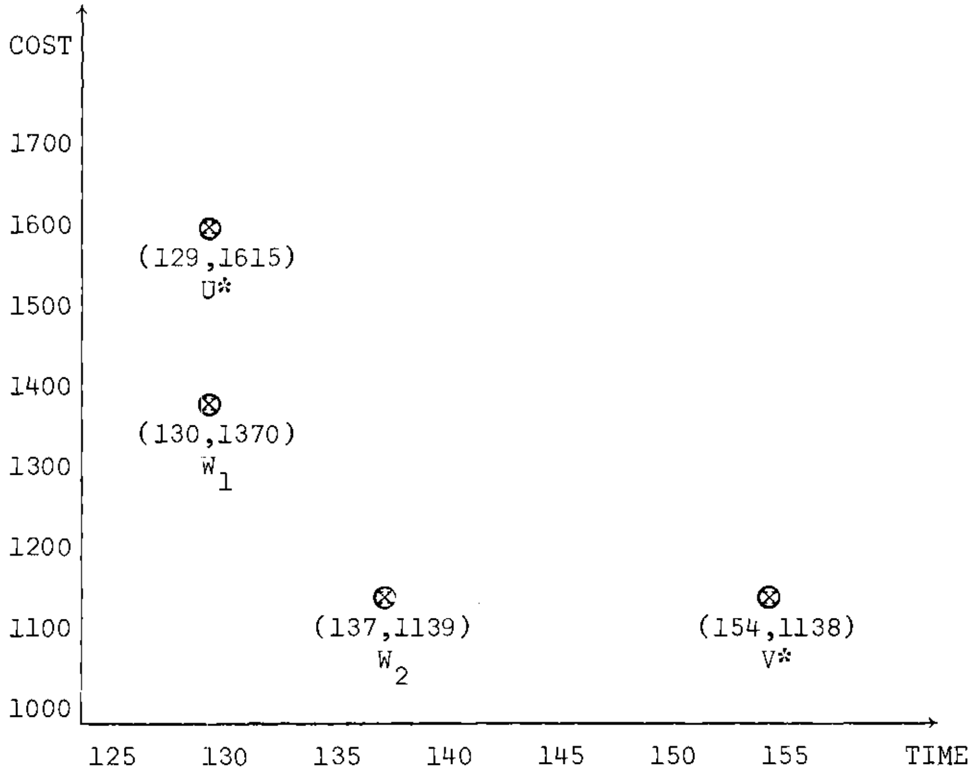


Figure 65. Time Cost Trade-Off

APPENDIX C

THE CASE OF RISK WITH STOCHASTIC DECISIONS: AN EXAMPLE

General

The example treated in this appendix is the network of Appendix B, which is treated as a stochastic decisions network by adding the following probability distribution to each decision vertex.

Decision Vertex	Probability Distribution		
D_1	$p_{11} = 0.3$	$p_{12} = 0.7$	
D_2	$p_{21} = 0.4$	$p_{22} = 0.6$	
D_3	$p_{31} = 0.2$	$p_{32} = 0.5$	$p_{33} = 0.3$
D_4	$p_{41} = 0.4$	$p_{42} = 0.6$	
D_5	$p_{51} = 0.3$	$p_{52} = 0.7$	
D_6	$p_{61} = 0.2$	$p_{62} = 0.8$	
D_7	$p_{71} = 0.8$	$p_{72} = 0.2$	
D_8	$p_{81} = 0.3$	$p_{82} = 0.7$	
D_9	$p_{91} = 0.4$	$p_{92} = 0.6$	

The solution of this case follows the sequence described in Chapter V.

Project Time and Cost Extremes

Minimum Time Problem

From the solution of the case of certainty in Appendix B:

$$T^* = 129$$

Minimum Cost Problem

From the solution of the case of certainty in Appendix B:

$$C^* = 1138$$

Maximum Time Problem

This problem is solved in a similar manner to the minimum time problem of the case of certainty, substituting the minimization process with a maximization one. The result obtained by doing so is:

$$T_{Max} = 188$$

Maximum Cost Problem

In this case, a maximization process is applied to the cost DP model of the case of certainty, yielding the following result:

$$C_{Max} = 1714$$

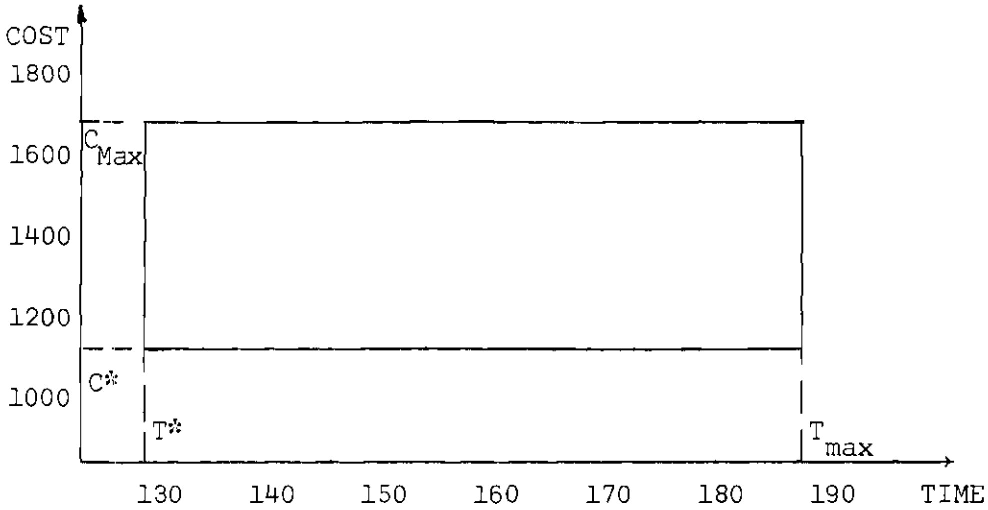


Figure 66. Region of Possible Outcomes

Expected Project Time

Optimistic Expected Time

Before presenting the solution itself, the method of dealing with type II divergence is introduced. Consider the schematic representation of DDS$_1$, shown in Fig. 67. The values shown near each $d_{jk(j)}$ are $t_{jk(j)}$.

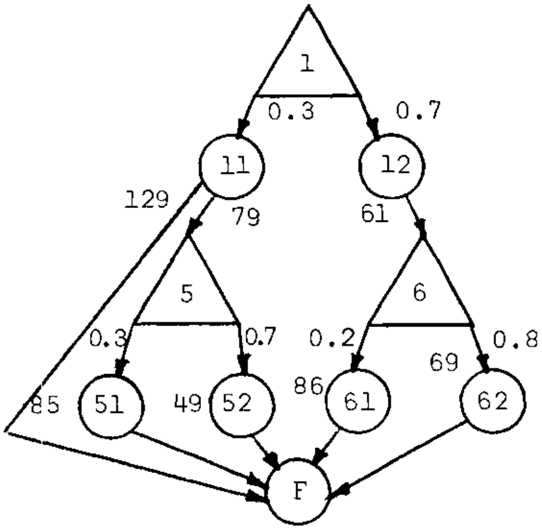


Figure 67. DDS$_1$

The expected time of this DDS_1 is:

$$E[T] = 0.3\{\text{Max}(\overbrace{[(0.3\times85+0.7\times49) + 79]}^{a},\overbrace{129}^{b})\} + 0.7\overbrace{(86\times0.2+69\times0.8) + 61}^{c} =$$

$$= \text{Max}[(0.3a+0.7c),(0.3b+0.7c)]$$

The right side of the above equality suggests how a type II divergence can be treated as a Markov process. This is done by considering two separate Markov processes, one for the states {11,51,52, 12,61,62} and one for {11,12,61,62}. The maximum of the two results is the expected time of this DDS.

Similar modifications are required for other types of non-serial models.

<u>Expected Time of DDS_1</u>. On the basis of the data of the case of certainty, the following matrices can be constructed. Note that two Markov processes are handled, as explained above.

Case 1

$$\underline{\underline{P}} = \begin{array}{c|cccccc} & S & 11 & 12 & 61 & 62 & F \\ \hline S & 0 & 0.3 & 0.7 & 0 & 0 & 0 \\ 11 & 0 & 0 & 0 & 0.2 & 0.8 & 1 \\ 12 & 0 & 0 & 0 & 0 & 0 & 0 \\ 61 & 0 & 0 & 0 & 0 & 0 & 1 \\ 62 & 0 & 0 & 0 & 0 & 0 & 1 \\ F & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \qquad \underline{\underline{T}} = \begin{array}{c|cccccc} & S & 11 & 12 & 61 & 62 & F \\ \hline S & 0 & 129 & 61 & & \text{O} & \\ 11 & & & & 86 & 69 & \\ 12 & & & & & & \\ 61 & & & \text{O} & & & \\ 62 & & & & & & \\ F & & & & & & \end{array}$$

$$\bar{\underline{T}}_n = \begin{array}{c|cccc} & n = & 1 & 2 & 3 \\ \hline \bar{t}_n(S) & & 81.4 & \underline{132.08} & 132.08 \\ \bar{t}_n(11) & & 0 & 0 & 0 \\ \bar{t}_n(12) & & 72.4 & 72.4 & 72.4 \\ \bar{t}_n(61) & & 0 & 0 & 0 \\ \bar{t}_n(62) & & 0 & 0 & 0 \\ \bar{t}_n(F) & & 0 & 0 & 0 \end{array} \qquad \bar{t}_{n*}(S) = 132.08$$

Case 2

$$\underline{\underline{P}} = \begin{array}{c|cccccccc} & S & 11 & 12 & 51 & 52 & 61 & 62 & F \\ \hline S & & 0.3 & 0.7 & & & 0 & & \\ 11 & & & & 0.3 & 0.7 & & & \\ 12 & & & & & & 0.2 & 0.8 & \\ 51 & & & & & & & & 1 \\ 52 & & & & & & & & 1 \\ 61 & & & & 0 & & & & 1 \\ 62 & & & & & & & & 1 \\ F & & & & & & & & 1 \end{array} \qquad \underline{\underline{T}} = \begin{array}{c|cccccccc} & S & 11 & 12 & 51 & 52 & 61 & 62 & F \\ \hline S & & 79 & 61 & & & 0 & & \\ 11 & & & & 85 & 49 & & & \\ 12 & & & & & & 86 & 69 & \\ 51 & & & & & & & & \\ 52 & & & & & & & & \\ 61 & & & & 0 & & & & \\ 62 & & & & & & & & \\ F & & & & & & & & \end{array}$$

$$\bar{\underline{T}}_n = \begin{array}{c|cccc} & n = & 1 & 2 & 3 \\ \hline \bar{t}_n(S) & & 66.4 & \underline{134.72} & 134.72 \\ \bar{t}_n(11) & & 58.8 & 58.8 & 58.8 \\ \bar{t}_n(12) & & 72.4 & 72.4 & 72.4 \\ \bar{t}_n(51) & & 0 & 0 & 0 \\ \bar{t}_n(52) & & 0 & 0 & 0 \\ \bar{t}_n(61) & & 0 & 0 & 0 \\ \bar{t}_n(62) & & 0 & 0 & 0 \\ \bar{t}_n(F) & & 0 & 0 & 0 \end{array} \qquad \bar{t}_{n*}(S) = 134.72$$

$$E[T^1] = \text{Max}(132.08,134.72) = 134.72$$

DDS_2 does not have any divergence, and therefore the procedure described in Chapter V is applicable. DDS_3 has a type II divergence and can be treated as DDS_1. DDS_4 does not have to be considered for the reasons discussed in the case of certainty. Thus, utilizing the data of Appendix B, and applying it as described before to DDS_2 and DDS_3 yields the following results:

$\underline{DDS_2}$ $E[T^2] = 150.8$

$\underline{DDS_3}$ *Case 1* $\bar{t}_{n*}(S) = 107.67$

Case 2 $\bar{t}_{n*}(S) = 117.43$

$$E[T^3] = \text{Max}(107.67,117.43) = 117.43$$

Recalling that $T^o = 102$, the expected project time is:

$$E[T] = \text{Max}(134.72,150.8,117.43,102) = 150.8$$

Pessimistic Expected Time. For the reasons discussed in Chapter V, the pessimistic expected time is not evaluated. Instead, a simulation has to be used, as was done for the example of Appendix D.

Expected Project Cost

Pessimistic Expected Cost

Evaluation of the pessimistic expected cost follows the procedure described in Chapter V. Since decision vertex 2 is certain to occur, it should not be considered for DDS_3, DDS_4. Decision nodes 61 and 62 are common to DDS_1 and DDS_2.

$\underline{DDS_1}$. On the basis of the data of the case of certainty, the following matrices can be formed.

$$\underline{\underline{P}} = \begin{array}{c|cccccccc|} & S & 11 & 12 & 51 & 52 & 61 & 62 & F \\ S & & 0.3 & 0.7 & & & O & & \\ 11 & & & & 0.3 & 0.7 & & & \\ 12 & & & & & & 0.2 & 0.8 & \\ 51 & & & & & & & & 1 \\ 52 & & & & & & & & 1 \\ 61 & & & & & & & & 1 \\ 62 & & & & O & & & & 1 \\ F & & & & & & & & 1 \end{array} \qquad \underline{\underline{C}} = \begin{array}{c|cccccccc|} & S & 11 & 12 & 51 & 52 & 61 & 62 & F \\ S & & 227 & 45 & & & O & & \\ 11 & & & & 92 & 63 & & & \\ 12 & & & & & & 110 & 111 & \\ 51 & & & & & & & & \\ 52 & & & & & & & & \\ 61 & & & & & & & & \\ 62 & & & & O & & & & \\ F & & & & & & & & \end{array}$$

$$\underline{\bar{C}}_n = \begin{array}{c|ccc} n = & 1 & 2 & 3 \\ \hline \bar{c}_n(S) & 99.6 & \underline{198.67} & 198.67 \\ \bar{c}_n(11) & 71.7 & 71.7 & 71.7 \\ \bar{c}_n(12) & 110.8 & 10.8 & 10.8 \\ \bar{c}_n(51) & 0 & 0 & 0 \\ \bar{c}_n(52) & 0 & 0 & 0 \\ \bar{c}_n(61) & 0 & 0 & 0 \\ \bar{c}_n(62) & 0 & 0 & 0 \\ \bar{c}_n(F) & 0 & 0 & 0 \end{array} \qquad \bar{c}_{n*}(S) = 198.67$$

$$E[C^1] = 198.67$$

Applying the same method to DDS_2, DDS_3, DDS_4, where decision vertex 2 is not considered for DDS_3 and DDS_4 yields the following results:

$$E[C^2] = 209.20$$

$$E[C^3] = 293.02$$

$$E[C^4] = 12.4$$

Recalling that $C_{PN} = 865$, the pessimistic expected cost is:

$$E[C] = 198.67 + 209.20 + 295.02 + 12.4 + 865 = \underline{1580.29}$$

Optimistic Expected Cost

The only common decision vertex that should be considered is "6."

$$\pi_1(12) = 0.7$$

$$\pi_1(21 \cup 22) = 1$$

$$\bar{c}_2^2(21) = \bar{c}_2^2(22) = \bar{c}_2^1(12) = 110.8$$

$$E[C_{CN}] = (0.7)(1)(110.8) = 77.56$$

$$E[C] = 1580.29 - 77.56 = \underline{1502.73}$$

Most Probable Project Network

The most probable project network is shown in Fig. 68. Its time and cost is:

$$T = 161$$

$$C = 1437$$

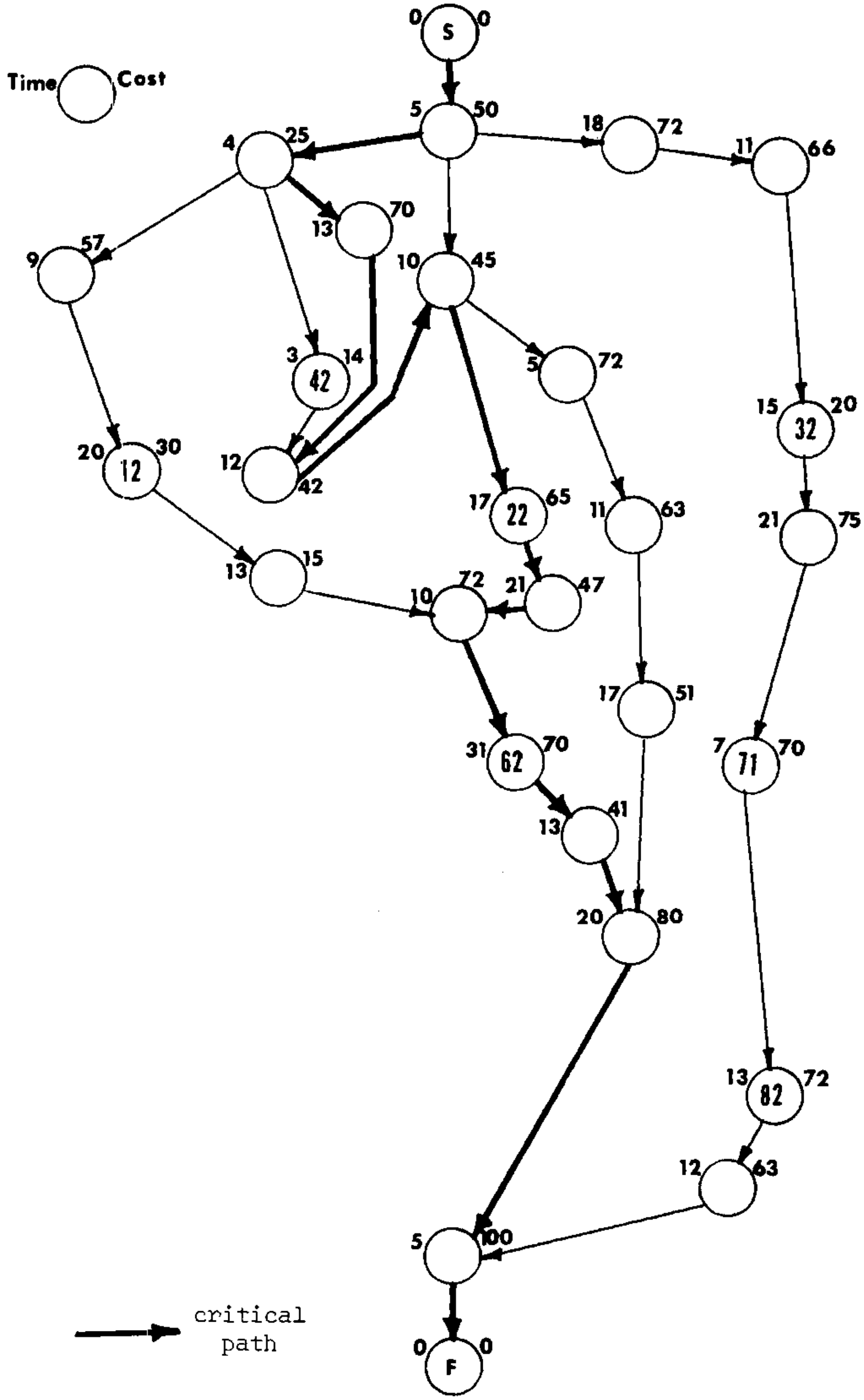


Figure 68. Most Probable Project Network

APPENDIX D

THE CASE OF RISK WITH STOCHASTIC OUTCOMES: AN EXAMPLE

General

This appendix includes a complete solution of the stochastic outcomes network of Fig. 69. Decomposition of the stochastic outcomes network into DDS was done following the first level labeling algorithm of Chapter IV. The cost return and time return associated with each stochastic outcome were obtained following the procedure of Chapter VI, and are shown in the time and cost matrices of the various DDS.

The solution of this case follows the sequence described in Chapter VI.

Minimum Expected Project Time

Following is an evaluation of the minimum expected project time. Shown are the return matrices only. Note the addition of the dummy decision F and the time associated with it, for DDS_1, DDS_2. This was done for the same reasons discussed in the case of certainty.

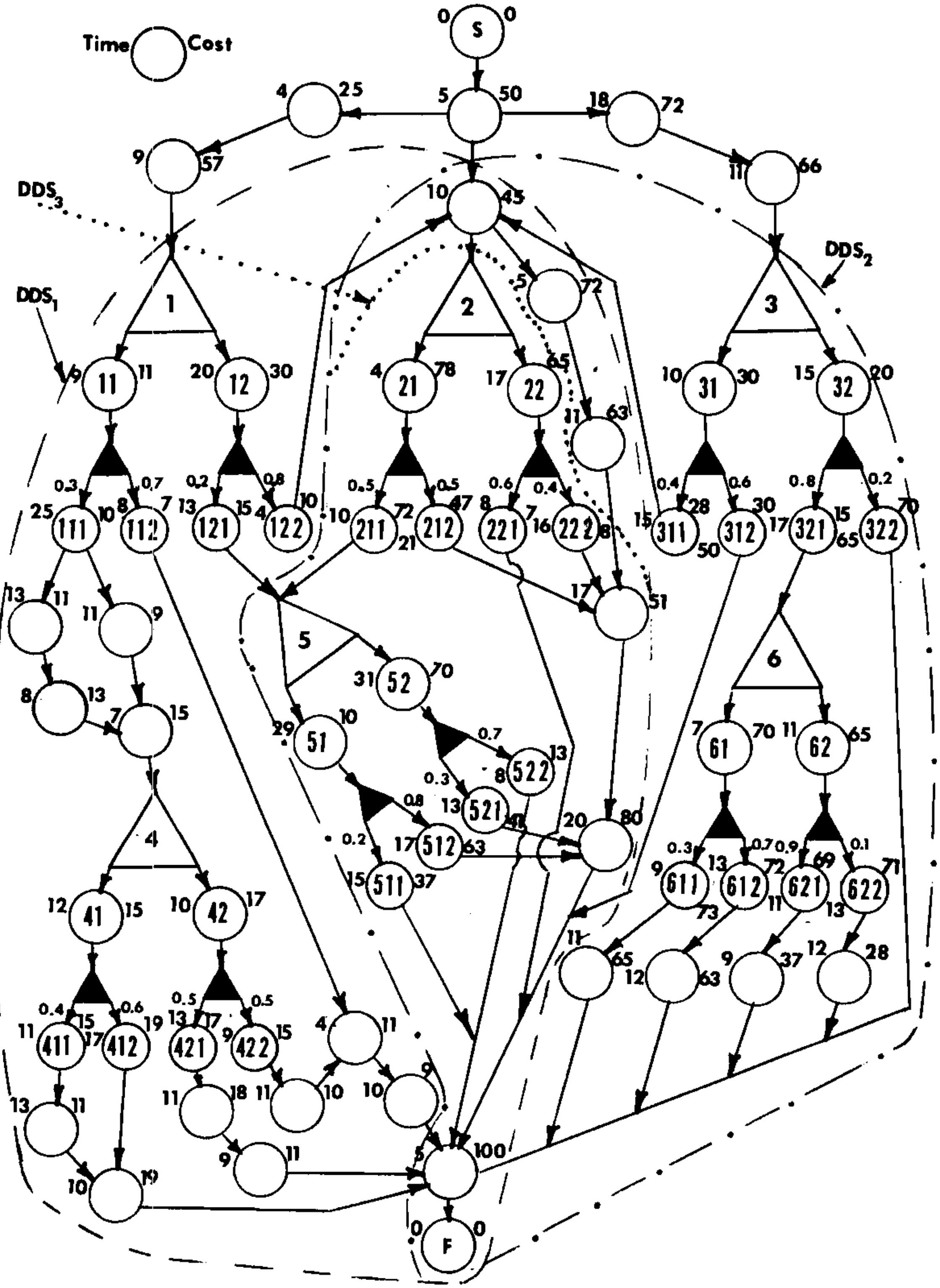


Figure 69. Stochastic Outcomes Network

$\underline{DDS_1}$—

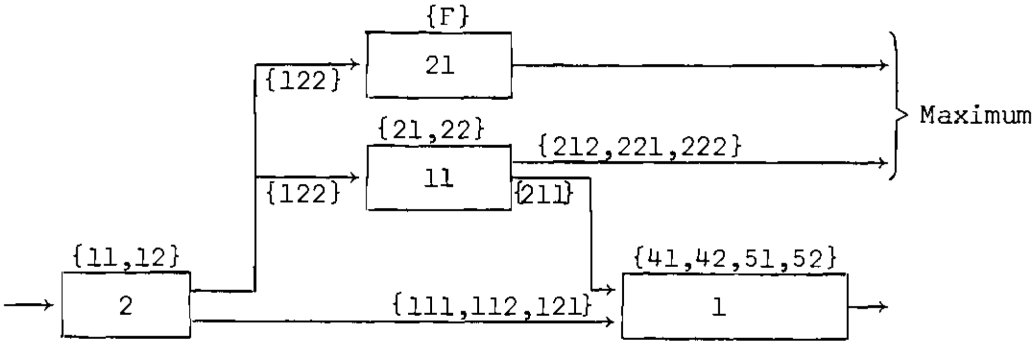


Figure 70. DDS_1 (Type II Divergence and Type II Feedforward)

In order to achieve a compact presentation of the solution, the stochastic matrices $\underline{\underline{P}}^i$ have been imbedded in $Q_i(X_i,Y_i,\Delta^i)$. Thus, $\underline{\underline{P}}^1$ for example, which is equal to:

$$\underline{\underline{P}}^1 = \begin{array}{c|ccccc|} & 41 & 42 & 51 & 52 & F \\ \hline 411 & 0.4 & & & & \\ 412 & 0.6 & & & O & \\ 421 & & 0.5 & & & \\ 422 & & 0.5 & & & \\ 511 & & & 0.2 & & \\ 512 & & & 0.8 & & \\ 521 & & & & 0.3 & \\ 522 & & O & & 0.7 & \\ F & & & & & 1 \end{array}$$

is shown as part of $Q_1(X_1,Y_1,\Delta^1)$.

Table 3. $T_i(X_i,Y_i\Delta^i)$; $Q_i(X_i,Y_i,\Delta^i)$; $\bar{Q}_i(X_i,Y_i)$ for DDS_1

Stage 2				
	11		12	
	111	112	121	122
	0.3	0.7	0.2	0.8
S	80 +46.8	54	51 +58.5	52 +58*
S	75.84		109.8	

Stage 21	
	F
	1
122	58
122	58

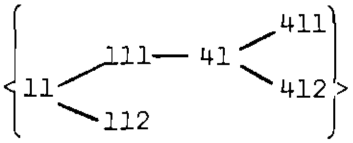
Stage 11				
	21		22	
	211	212	221	222
	0.5	0.5	0.6	0.4
122	16 +58.5	67	30	75
122	69.75		48	

Stage 1									
	41		42		51		52		F
	411	412	421	422	511	512	521	522	
	0.4	0.6	0.5	0.5	0.2	0.8	0.3	0.7	
111	51	44	48	49	x	x	x	x	x
112	x	x	x	x	x	x	x	x	0
121	x	x	x	x	49	71	69	54	
211	x	x	x	x	49	71	69	54	
111	46.8		48.5						
112	x		x		x		x		0
121	x		x		66.6		58.5		
211	x		x		66.6		58.5		

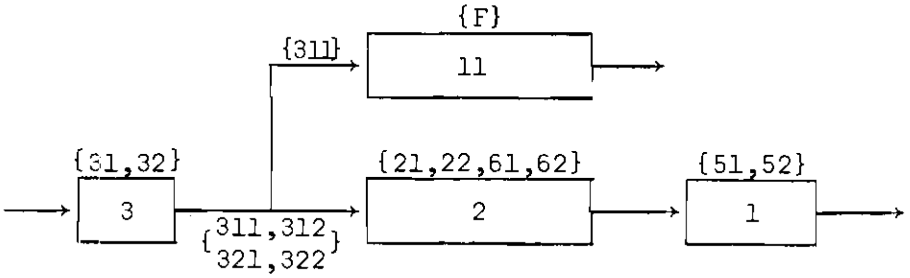
*Max(58,48).

$$E[T^1]^* = f^{-1}_{2(c)}(S) = 75.84$$

$\bar{U}_1^* =$

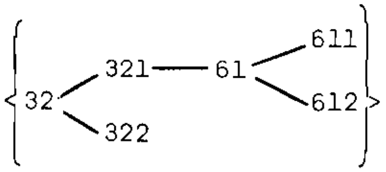


DDS$_2$ (Type II Divergence)



$E[T^2]^* = 103$ (See Table 4.)

$\bar{U}_2^* =$



DDS$_3$

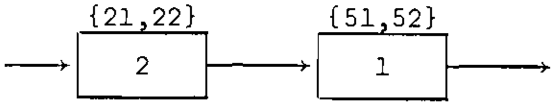


Table 4. $T_i(X_i,Y_i,\Delta^i)$; $Q_i(X_i,Y_i,\Delta^i)$; $\bar{Q}_i(X_i,Y_i)$ for DDS_2

Stage 3					Stage 2										Stage 1					Stage 11	
																					F
	311	312	321	322		211	212	221	222	611	612	621	622	F		511	512	521	522		
	0.4	0.6	0.8	0.2		0.5	0.5	0.6	0.4	0.3	0.7	0.9	0.1	1		0.2	0.8	0.3	0.7		1
					311	14 +58.5	67	30	75	x	x	x	x	x							
S	69 +58*	99	66 +35.5	109	312	x	x	x	x	x	x	x	x	0	211	49	71	69	54	311	58
					321	x	x	x	x	32	37	36	41								
					322	x	x	x	x	x	x	x	x	0							
					311	69.75		48		x		x		x							
S	110.2		103		312	x		x		x		x		x	211	66.6		58.5		311	58
					321	x		x		35.5		36.5		x							
					322	x		x		x		x		0							

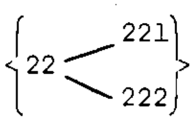
*Max(48,58).

Table 5. $T_i(X_i,Y_i,\Delta^i)$; $Q_i(X_i,Y_i,\Delta^i)$; $\bar{Q}_i(X_i,Y_i)$ for DDS_3

	Stage 2					Stage 1			
	21		22			51		52	
	211	212	221	222		511	512	521	522
	0.5	0.5	0.6	0.4		0.2	0.8	0.3	0.7
S	29 +58.5	82	45	90	211	49	71	69	54
S	84.75		63		211	66.6		58.5	

$$E[T^3]^* = 63$$

$\bar{U}_3^* =$

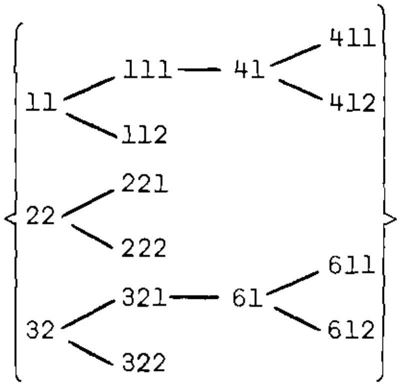


<u>CDDS</u> $T^o = 73$

<u>Minimum Expected Project Time</u>

$$E[T]^* = \text{Max}(75.84,103,63,73) = \underline{103}$$

$\bar{U}^* =$



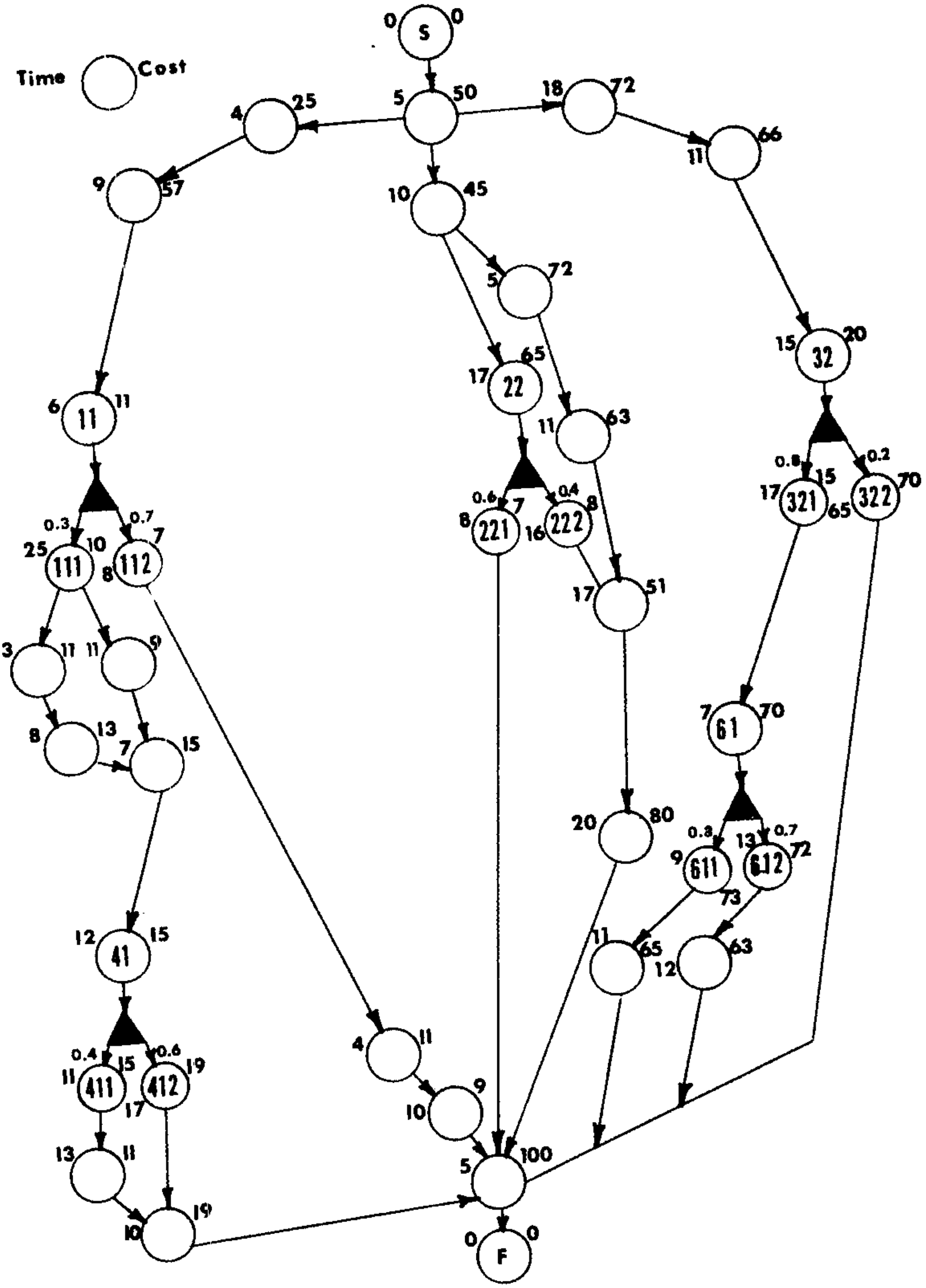


Figure 71. PRN--Time

Expected Cost of the Minimum Expected Time

Using the PRN of Fig. 71 one obtains:

DDS$_1$—

$$\underline{\underline{P}} = \begin{array}{c} \\ S \\ 111 \\ 112 \\ 411 \\ 412 \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} S & 111 & 112 & 411 & 412 & F \end{array} \\ \left(\begin{array}{cccccc} 0 & 0.3 & 0.7 & \multicolumn{2}{c}{O} & \\ & & & 0.4 & 0.6 & \\ & & & & & 1 \\ & & & & & 1 \\ & & O & & & 1 \\ & & & & & 1 \end{array}\right) \end{array} \qquad \underline{\underline{T}} = \begin{array}{c} \\ S \\ 111 \\ 112 \\ 411 \\ 412 \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} S & 111 & 112 & 411 & 412 & F \end{array} \\ \left(\begin{array}{cccccc} & 69 & 38 & O & & \\ & & & 60 & 53 & \\ & & & & & \\ & & O & & & \\ & & & & & \\ & & & & & \end{array}\right) \end{array}$$

$$\underline{\bar{C}}_n = \begin{bmatrix} \bar{c}_n(S) \\ \bar{c}_n(111) \\ \bar{c}_n(112) \\ \bar{c}_n(411) \\ \bar{c}_n(412) \\ \bar{c}_n(F) \end{bmatrix} \quad \begin{array}{cccc} n = & 1 & 2 & 3 \\ & 47.3 & \underline{64.04} & 64.04 \\ & 55.8 & 55.8 & 55.8 \\ & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ & 0 & 0 & 0 \end{array} \qquad \bar{c}_2(S) = 64.04$$

$$E[C^1]_{E[T^1]*} = 64.04$$

Using the same approach for DDS$_2$, DDS$_3$, the following values are obtained:

$$E[C^2]_{E[T^2]*} = 210.72$$

$$E[C^3]_{E[T3]*} = 72.4$$

since:

$$C_{PN} = 207,$$

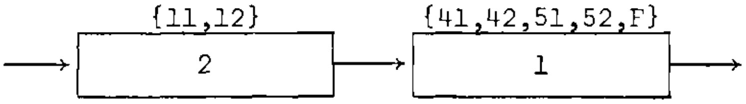
then:

$$E[C]_{E[T]*} = 64.04 + 210.72 + 72.4 + 207 = \underline{554.16}$$

Minimum Expected Project Cost

Decision vertex 2 is common to DDS$_1$, DDS$_2$, DDS$_3$. However, since cost is additive (unlike time), and since decision vertex 2 is certain to be encountered, it should be considered for DDS$_3$ only. (See the "Cost Refinement" of the case of certainty.)

$\underline{DDS_1}$—



$$E[C^1]* = 54.56 \qquad \text{(See Table 6.)}$$

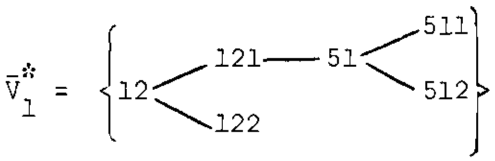


Table 6. $C_i(X_i,Y_i,\Delta^i)$; $R_i(X_i,Y_i,\Delta^i)$; $\bar{R}_i(X_i,Y_i)$ for DDS_1

	Stage 2					Stage 1								
	11		12			41		42		51		52		
	111	112	121	122		411	412	421	422	511	512	521	522	F
	0.3	0.7	0.2	0.8		0.4	0.6	0.5	0.5	0.2	0.8	0.3	0.7	1
S	69 +55.8	38	45 +67.8	40	111	60	53	63	62	x	x	x	x	x
					112	x	x	x	x	x	x	x	x	0
					121	x	x	x	x	47	73	111	83	x
					122	x	x	x	x	x	x	x	x	0
S	64.04		54.56		111	55.8		72.5		x		x		x
					112	x		x		x		x		0
					121	x		x		67.8		91.4		x
					122	x		x		x		x		0

DDS_2 (A Two-Stage Model)

Table 7. $C_i(X_i,Y_i,\Delta^i)$; $R_i(X_i,Y_i,\Delta^i)$; $\bar{R}_i(X_i,Y_i)$ for DDS_2

	Stage 2					Stage 1				
	31		32			61		62		F
	311	312	321	322		611	612	621	622	
	0.4	0.6	0.8	0.2		0.3	0.7	0.9	0.1	
S	58	60	35 +155.5	90	311	x	x	x	x	0
					312	x	x	x	x	0
					321	208	205	171	164	
					322	x	x	x	x	0
S	59.2		170.4		311					
					312					
					321	205.9		155.5		
					322					

$$E[C^2]^* = 59.2$$

$$\bar{V}_2^* = \left\{ 31 \begin{matrix} \nearrow 311 \\ \searrow 312 \end{matrix} \right\}$$

DDS$_3$ (A Two-Stage Model)

Table 8. $C_i(X_i, Y_i, \Delta^i)$; $R_i(X_i, Y_i, \Delta^i)$; $\bar{R}_i(X_i, Y_i)$ for DDS$_3$

<table>
<tr><th colspan="5">Stage 2</th><th colspan="6">Stage 1</th></tr>
<tr><th></th><th colspan="2">21</th><th colspan="2">22</th><th></th><th colspan="2">51</th><th colspan="2">52</th><th></th></tr>
<tr><th></th><th>212</th><th>212</th><th>221</th><th>222</th><th></th><th>511</th><th>512</th><th>521</th><th>522</th><th>F</th></tr>
<tr><td rowspan="4">S</td><td rowspan="4">150
+67.8</td><td rowspan="4">125</td><td rowspan="4">72</td><td rowspan="4">73</td><td>211</td><td>47</td><td>73</td><td>111</td><td>83</td><td>x</td></tr>
<tr><td>212</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td></tr>
<tr><td>221</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td></tr>
<tr><td>222</td><td>x</td><td>x</td><td>x</td><td>x</td><td>0</td></tr>
<tr><td rowspan="4">S</td><td colspan="2" rowspan="4">171.4</td><td colspan="2" rowspan="4">72.4</td><td>211</td><td colspan="2">67.8</td><td colspan="2">91.4</td><td></td></tr>
<tr><td>212</td><td colspan="2">x</td><td colspan="2">x</td><td>0</td></tr>
<tr><td>221</td><td colspan="2">x</td><td colspan="2">x</td><td>0</td></tr>
<tr><td>222</td><td colspan="2">x</td><td colspan="2">x</td><td>0</td></tr>
</table>

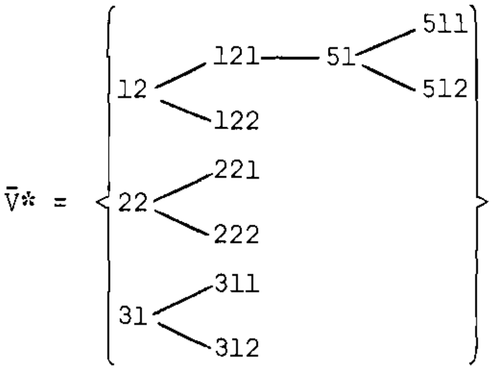
$$E[C^3]^* = 72.4$$

$$\bar{V}_3^* = \left\{ 22 \begin{matrix} \nearrow 221 \\ \searrow 222 \end{matrix} \right\}$$

Minimum Expected Project Cost

Since $C_{PN} = 207$, the minimum expected project cost is:

$$E[C]^* = 54.56 + 59.2 + 72.4 + 207 = 393.16$$



Expected Time of the Minimum Expected Cost

Using the PRN-Cost of Fig. 72, and following the procedure described in Chapter V, the following values are obtained:

DDS$_1$—

Case 1 $\bar{t}_2(S) = 103.52$

Case 2 $\bar{t}_2(S) = 111.52$

then:

$$E[T^1]_{E[C^1]*} = \text{Max}(103.52,111.52) = 111.52$$

DDS$_2$—

Case 1 $\bar{t}_2(S) = 106.2$

Case 2 $\bar{t}_2(S) = 110.2$

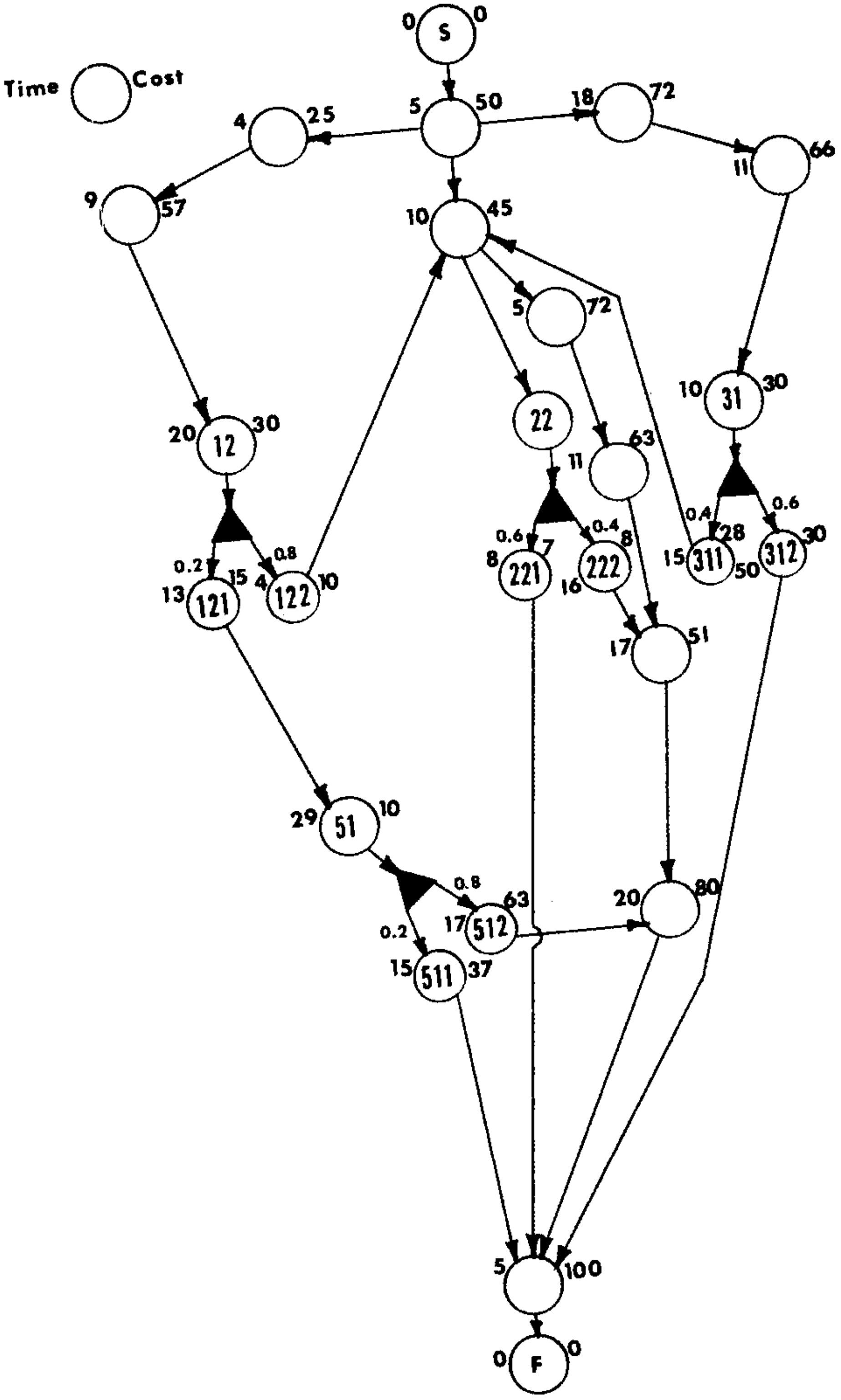


Figure 72. PRN--Cost

then:

$$E[T^2]_{E[C^2]*} = \text{Max}(106.2,110.2) = 110.2$$

DDS$_3$

$$E[T^3]_{E[C^3]*} = 63$$

CDDS

$$T^o = 73$$

The expected time of the minimum expected cost is:

$$E[T]_{E[C]*} = \text{Max}(111.53,110.2,63,73) = \underline{111.53}$$

Region of Possible Outcomes

Region of Possible Outcomes for the Whole Project

Following the method described in Chapter VI and the procedure for the case of certainty (Chapter IV), the following results are obtained.

Time Extremes.

Maximum Time: $T_{\text{Max}} = \text{Max}(137,154,90,73) = 154$

and the associated curtailed strategy is:

{31-311,21-211,51-512}

Minimum Time: $T_{Min} = Max(54,98,45,78) = 98$

and the associated curtailed strategy is:

{11-112,22-221,32-321,61-611}

Cost Extremes.

Maximum Cost: $C_{Max} = 132 + 243 + 261 + 207 = 843$

and the associated curtailed strategy is:

{11-111,21-211,32-321,42-421,52-521,61-611}

Minimum Cost: $C_{Min} = 38 + 58 + 72 + 207 = 375$

and the associated curtailed strategy is:

{11-112,22-221,31-311}

The region of possible outcomes for the whole project is illustrated in Fig. 73.

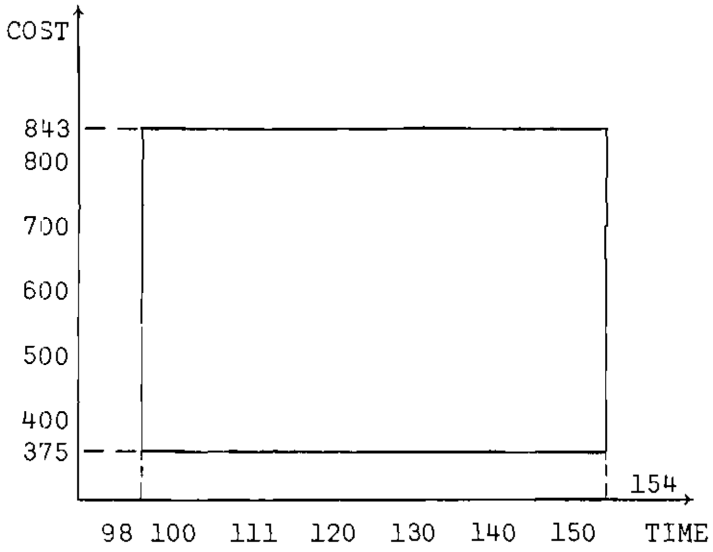


Figure 73. Region of Possible Outcomes

Region of Possible Outcomes for an Opening Policy

Applying the procedure described in Chapter VI to each stochastic outcomes network resulting from the selection of an opening policy, the values summarized in Table 9 are obtained.

Table 9. Region of Possible Outcomes for H_i

Opening Policy	Minimum Time	Maximum Time	Minimum Cost	Maximum Cost
$H_1=\{11,21,31\}$	99	154	408	660
$H_2=\{11,22,31\}$	99	144	375	472
$H_3=\{11,21,32\}$	98	131	460	843
$H_4=\{11,22,32\}$	98	131	407	655
$H_5=\{12,21,31\}$	100	154	430	573
$H_6=\{12,22,31\}$	100	144	377	496
$H_7=\{12,21,32\}$	100	137	462	756
$H_8=\{12,22,32\}$	100	137	409	679

These regions are illustrated in Figs. 74 and 75 along with the simulation results.

Most Probable Outcome

Minimum Expected Time Strategy

Referring to Fig. 71, the most probable path for each DDS is:

$\underline{\mathrm{DDS}_1}$ (11-112) with $T^1 = 54$ $C^1 = 38$

$\underline{\mathrm{DDS}_2}$ (32-321,61-611) with $T^2 = 104$ $C^2 = 240$

$\underline{\mathrm{DDS}_3}$ (22-221) with $T^3 = 45$ $C^3 = 72$

Recalling that $T^o = 73$ and $C_{PN} = 207$, the most probable values for the minimum expected time strategy are:

$$T = \mathrm{Max}(54,104,45,73) = 104$$

$$C = 38 + 240 + 72 + 207 = 557$$

Minimum Expected Cost Strategy

Referring to Fig. 72, the following values are obtained.

$\underline{\mathrm{DDS}_1}$ (12-122,22-221) with $T^1 = 110$ $C^1 = 40$

$\underline{\mathrm{DDS}_2}$ (31-312) with $T^2 = 99$ $C^2 = 60$

$\underline{\mathrm{DDS}_3}$ (22-221) with $T^3 = 45$ $C^3 = 72$

The most probable values for the minimum expected cost strategy are:

$$T = \mathrm{Max}(110,99,72,73) = 110$$

$$C = 40 + 60 + 72 + 207 = 379$$

Evaluation of Risk in an Optimal Strategy

Tables 10 and 11 show the paths probabilities for the two optimal strategies. For comparison, the simulation results discussed in the following section are shown here, too.

Evaluating the expected values for the minimum expected time strategy (Table 10), one gets:

$$E[T] = 110.14$$

$$E[C] = 347.16 + 207 = 554.16 \qquad (\text{where } C_{PN} = 207)$$

Note that the expected cost is the same as the one obtained before, where the value of E[T] is the exact value vs. the lower bound (103) that was previously obtained.

The expected values for the minimum expected cost strategy (Table 11) are:

$$E[T] = 123.7$$

$$E[C] = 186.16 + 207 = 393.16 \qquad (\text{where } C_{PN} = 207)$$

Table 10. Path Enumeration for the Minimum Expected Time Strategy

	111	112	211	412	221	222	321	322	611	612	Probability	Time	Cost	Simulation Results
Time	80	54	51	44	45	90	66	109	32	37				
Cost	69	38	60	53	72	73	35	90	208	205				
	0.3	0.7	0.4	0.6	0.6	0.4	0.8	0.2	0.3	0.7				
	x		x		x			x			0.0144	131	291	0.017
	x		x			x		x			0.0096	131	292	0.004
	x		x		x		x		x		0.01728	131	444	0.013
	x		x		x		x			x	0.04032	131	441	0.029
	x		x			x	x		x		0.01152	131	445	0.008
	x		x			x	x			x	0.02688	131	442	0.044
	x			x	x			x			0.0216	124	284	0.012
	x			x		x		x			0.0144	124	285	0.018
	x			x	x		x		x		0.02592	124	437	0.020
	x			x	x		x			x	0.06048	124	434	0.072
	x			x		x	x		x		0.01728	124	438	0.026
	x			x		x	x			x	0.04032	124	435	0.029
		x			x			x			0.084	109	200	0.082
		x				x		x			0.056	109	201	0.058
		x			x		x		x		0.1008	98	353	0.085
		x			x		x			x	0.2352	103	350	0.278
		x				x	x		x		0.0672	98	354	0.082
		x				x	x			x	0.1568	103	351	0.123
											1.0000			1.000

Table 11. Path Enumeration for the Minimum Expected Cost Strategy

	121	122	511	512	221	222	311	312	Probability	Time	Cost	Simulation Results
Time	451	(110) 52	49	71	(45) 30	(90) 75	69	99				
Cost	45	40	47	73	72	73	58	60				
Prob.	0.2	0.8	0.2	0.8	0.6	0.4	0.4	0.6				
	x			x		x		x	0.0384	122	251	0.030
	x			x	x			x	0.0576	122	250	0.067
	x			x		x	x		0.0256	144	249	0.011
	x			x	x		x		0.0384	127	248	0.044
		x				x		x	0.1920	127	173	0.190
		x			x			x	0.2880	110	172	0.296
		x				x	x		0.1280	144	171	0.135
		x			x		x		0.1920	127	170	0.198
	x		x			x		x	0.0096	100	225	0.010
	x		x		x			x	0.0144	100	224	0.001
	x		x			x	x		0.0064	144	223	0.018
	x		x		x		x		0.0096	127	222	0.000
									1.000			1.000

Simulation

Evaluation of Risk in an Optimal Strategy

Using the PRN-Time and PRN-Cost of Figs. 71 and 72, 2000 simulation runs were performed, as described in Chapter VI, in order to obtain the probability distribution of time and cost values. The results of these simulation runs are shown in Tables 10 and 11.

Probability Associated with an Opening Policy

Following the procedure described in Chapter VI, 1000 simulation runs were performed for each of the two criteria, namely the minimum time criterion and the minimum cost criterion. The results are summarized in Tables 12, 13, 14, 15, and Figs. 74, 75.

The curtailed strategies $\bar{\bar{U}}^*$ are shown for policies H_1 through H_4 only, because of the reasons mentioned in Chapter VI.

Table 12. Simulation Results: Minimum Cost Criterion

<table>
<tr><td>H_i</td><td colspan="4">H_2</td><td colspan="10">H_6</td></tr>
<tr><td>P_{H_i}</td><td colspan="4">0.705</td><td colspan="10">0.295</td></tr>
<tr><td>T</td><td>99</td><td colspan="2">127</td><td>144</td><td colspan="2">100</td><td colspan="2">110</td><td colspan="2">122</td><td colspan="2">127</td><td colspan="2">144</td></tr>
<tr><td>P(T)</td><td>.594</td><td colspan="2">.236</td><td>.17</td><td colspan="2">.026</td><td colspan="2">.315</td><td colspan="2">.101</td><td colspan="2">.396</td><td colspan="2">.162</td></tr>
<tr><td>E[T]</td><td colspan="4">113.25</td><td colspan="10">123.18</td></tr>
<tr><td>C</td><td>168</td><td>169</td><td>170</td><td>171</td><td>170</td><td>171</td><td>172</td><td>173</td><td>222</td><td>224</td><td>248</td><td>249</td><td>250</td><td>251</td></tr>
<tr><td>P(C)</td><td>.235</td><td>.172</td><td>.327</td><td>.266</td><td>.15</td><td>.135</td><td>.315</td><td>.20</td><td>.02</td><td>.023</td><td>.033</td><td>.027</td><td>.047</td><td>.05</td></tr>
<tr><td>E[C]</td><td colspan="4">169.45</td><td colspan="10">186.07</td></tr>
</table>

NOTE: C_{PN} = 207 should be added to all values of E[C].

Table 13. Simulation Results: Minimum Time Criterion

<table>
<tr><td>H_i</td><td colspan="3">H_1</td><td colspan="3">H_2</td><td colspan="10">H_3</td><td colspan="5">H_4</td><td colspan="6">H_5</td></tr>
<tr><td>P_{H_i}</td><td colspan="3">0.15</td><td colspan="3">0.172</td><td colspan="10">0.145</td><td colspan="5">0.239</td><td colspan="6">0.046</td></tr>
<tr><td>T</td><td colspan="3">99</td><td>99</td><td colspan="2">124</td><td colspan="3">98</td><td colspan="2">102</td><td colspan="2">103</td><td colspan="3">109</td><td>98</td><td colspan="2">102</td><td>103</td><td>109</td><td>105</td><td colspan="2">115</td><td>119</td><td colspan="2">120</td></tr>
<tr><td>P(T)</td><td colspan="3">1</td><td>.994</td><td colspan="2">.006</td><td colspan="3">.406</td><td colspan="2">.42</td><td colspan="2">.151</td><td colspan="3">.023</td><td>.39</td><td colspan="2">.40</td><td>.07</td><td>.14</td><td>.195</td><td colspan="2">.154</td><td>.347</td><td colspan="2">.304</td></tr>
<tr><td>E[T]</td><td colspan="3">99</td><td colspan="3">99.1</td><td colspan="10">100.67</td><td colspan="5">101.47</td><td colspan="6">115.7</td></tr>
<tr><td>C</td><td>223</td><td>295</td><td>331</td><td>170</td><td>171</td><td>255</td><td>253</td><td>325</td><td>361</td><td>369</td><td>406</td><td>441</td><td>475</td><td>477</td><td>478</td><td>514</td><td>200</td><td>316</td><td>317</td><td>350</td><td>353</td><td>225</td><td>297</td><td>313</td><td>333</td><td>338</td><td>341</td></tr>
<tr><td>P(C)</td><td>0.58</td><td>0.167</td><td>.263</td><td>.965</td><td>0.03</td><td>.005</td><td>.09</td><td>.03</td><td>.03</td><td>.213</td><td>.234</td><td>.0195</td><td>.0195</td><td>.193</td><td>.02</td><td>.151</td><td>.138</td><td>.37</td><td>.045</td><td>.067</td><td>.38</td><td>.347</td><td>.152</td><td>.108</td><td>.282</td><td>.086</td><td>.025</td></tr>
<tr><td>E[C]</td><td colspan="3">262.2</td><td colspan="3">170.04</td><td colspan="10">413.85</td><td colspan="5">316.26</td><td colspan="6">288.5</td></tr>
</table>

<table>
<tr><td>H_i</td><td colspan="5">H_6</td><td colspan="13">H_7</td><td colspan="10">H_8</td></tr>
<tr><td>P_{H_i}</td><td colspan="5">0.098</td><td colspan="13">0.067</td><td colspan="10">0.083</td></tr>
<tr><td>T</td><td>100</td><td>105</td><td>110</td><td>120</td><td>127</td><td colspan="3">105</td><td colspan="2">109</td><td colspan="2">115</td><td colspan="4">119</td><td colspan="2">120</td><td colspan="2">100</td><td>102</td><td colspan="2">105</td><td colspan="2">109</td><td>110</td><td colspan="2">120</td></tr>
<tr><td>P(T)</td><td>.03</td><td>.112</td><td>.775</td><td>.013</td><td>.07</td><td colspan="3">.09</td><td colspan="2">.05</td><td colspan="2">.07</td><td colspan="4">.42</td><td colspan="2">.37</td><td colspan="2">.06</td><td>.05</td><td colspan="2">.012</td><td colspan="2">.012</td><td>.746</td><td colspan="2">.120</td></tr>
<tr><td>E[T]</td><td colspan="5">110.35</td><td colspan="13">117.33</td><td colspan="10">110.06</td></tr>
<tr><td>C</td><td>172</td><td>173</td><td>224</td><td>260</td><td>288</td><td>363</td><td>371</td><td>375</td><td>408</td><td>443</td><td>451</td><td>459</td><td>479</td><td>480</td><td>496</td><td>516</td><td>521</td><td>524</td><td>202</td><td>290</td><td>318</td><td>352</td><td>355</td><td>370</td><td>407</td><td>434</td><td>443</td><td>471</td></tr>
<tr><td>P(C)</td><td>.775</td><td>.07</td><td>.03</td><td>.112</td><td>.013</td><td>.149</td><td>.223</td><td>.0135</td><td>.194</td><td>.0135</td><td>.0135</td><td>.06</td><td>.10</td><td>.06</td><td>.03</td><td>.06</td><td>.0135</td><td>.07</td><td>.072</td><td>.012</td><td>.313</td><td>.072</td><td>.325</td><td>.05</td><td>.05</td><td>.084</td><td>.011</td><td>.011</td></tr>
<tr><td>E[C]</td><td colspan="5">184.96</td><td colspan="13">426.8</td><td colspan="10">343.57</td></tr>
</table>

Note: $C_{PN} = 207$ should be added to all values of E[C].

Table 14. Simulation Results: $\bar{\bar{U}}^*$

H_i	T	C	$\bar{\bar{U}}^*$	H_i	T	C	$\bar{\bar{U}}^*$
H_1	99	223	$\begin{Bmatrix}11-112\\21-212\\31-312\end{Bmatrix}$	H_3	98	514	$\begin{Bmatrix}11-112\\21-211-52-522\\32-321-61-611\end{Bmatrix}$
	99	295	$\begin{Bmatrix}11-112\\21-211-51-511\\31-312\end{Bmatrix}$		98	478	$\begin{Bmatrix}11-112\\21-211-51-511\\32-321-61-611\end{Bmatrix}$
	99	331	$\begin{Bmatrix}11-112\\21-211-52-522\\31-312\end{Bmatrix}$		98	406	$\begin{Bmatrix}11-112\\21-212\\32-321-61-611\end{Bmatrix}$
H_2	99	170	$\begin{Bmatrix}11-112\\22-221\\31-312\end{Bmatrix}$		102	477	$\begin{Bmatrix}11-112\\21-211-52-522\\32-321-62-621\end{Bmatrix}$
	99	171	$\begin{Bmatrix}11-112\\22-222\\31-312\end{Bmatrix}$		102	369	$\begin{Bmatrix}11-112\\21-212\\32-321-62-621\end{Bmatrix}$
	124	255	$\begin{Bmatrix}11-111-41-412\\22-222\\31-312\end{Bmatrix}$		102	441	$\begin{Bmatrix}11-112\\21-211-51-511\\32-321-62-621\end{Bmatrix}$
H_4	102	316	$\begin{Bmatrix}11-112\\22-221\\32-321-62-621\end{Bmatrix}$		103	475	$\begin{Bmatrix}11-112\\21-211-51-511\\32-321-61-612\end{Bmatrix}$
	98	353	$\begin{Bmatrix}11-112\\22-221\\32-321-61-611\end{Bmatrix}$		109	253	$\begin{Bmatrix}11-112\\21-212\\32-322\end{Bmatrix}$
	103	350	$\begin{Bmatrix}11-112\\22-221\\32-321-61-612\end{Bmatrix}$		109	325	$\begin{Bmatrix}11-112\\21-211-51-511\\32-322\end{Bmatrix}$
	109	200	$\begin{Bmatrix}11-112\\22-221\\32-322\end{Bmatrix}$		109	361	$\begin{Bmatrix}11-112\\21-211-52-522\\32-322\end{Bmatrix}$

Table 15. Simulation Results: $\bar{\bar{V}}^*$

H_i	T	C	$\bar{\bar{V}}^*$	H_i	T	C	$\bar{\bar{V}}^*$
H_2	99	171	$\begin{Bmatrix} 11-112 \\ 22-221 \\ 31-312 \end{Bmatrix}$	H_6	100	224	$\begin{Bmatrix} 12-121-51-511 \\ 22-221 \\ 31-312 \end{Bmatrix}$
	99	170	$\begin{Bmatrix} 11-112 \\ 22-222 \\ 31-312 \end{Bmatrix}$		110	172	$\begin{Bmatrix} 12-122 \\ 22-221 \\ 31-312 \end{Bmatrix}$
	127	168	$\begin{Bmatrix} 11-112 \\ 22-221 \\ 31-311 \end{Bmatrix}$		122	250	$\begin{Bmatrix} 12-121-51-512 \\ 22-221 \\ 31-312 \end{Bmatrix}$
	144	169	$\begin{Bmatrix} 11-112 \\ 22-222 \\ 31-311 \end{Bmatrix}$		122	251	$\begin{Bmatrix} 12-121-51-512 \\ 22-222 \\ 31-312 \end{Bmatrix}$
					127	222	$\begin{Bmatrix} 12-121-51-511 \\ 22-221 \\ 31-311 \end{Bmatrix}$
					127	170	$\begin{Bmatrix} 12-122 \\ 22-222 \\ 31-311 \end{Bmatrix}$
					127	248	$\begin{Bmatrix} 12-121-51-512 \\ 22-221 \\ 31-311 \end{Bmatrix}$
					127	173	$\begin{Bmatrix} 12-122 \\ 22-222 \\ 31-312 \end{Bmatrix}$
					144	249	$\begin{Bmatrix} 12-121-51-512 \\ 22-222 \\ 31-311 \end{Bmatrix}$
					144	171	$\begin{Bmatrix} 12-122 \\ 22-222 \\ 31-311 \end{Bmatrix}$

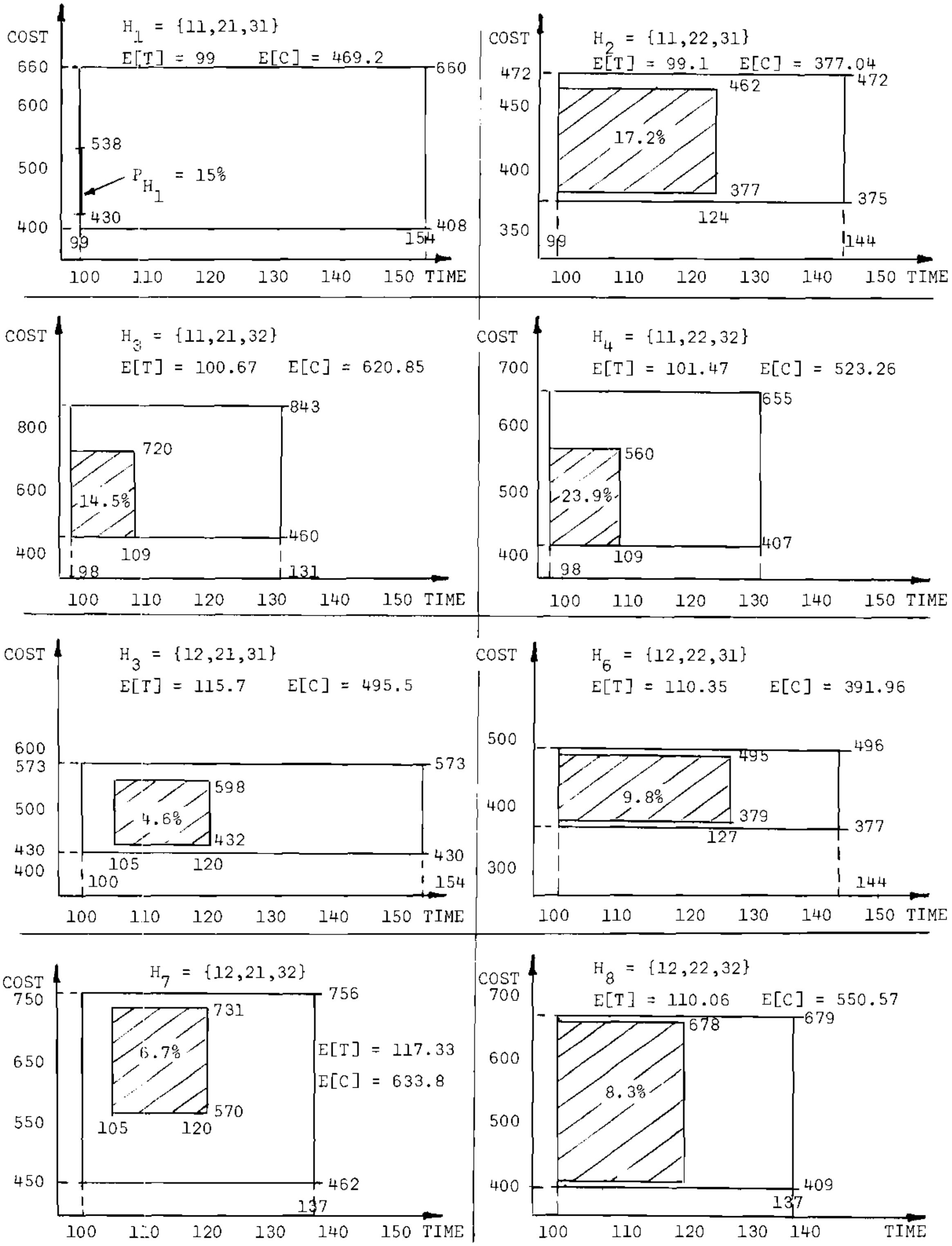


Figure 74. Region of Possible Outcomes and Probability Zones for Opening Policy H_i: Minimum Time Criterion

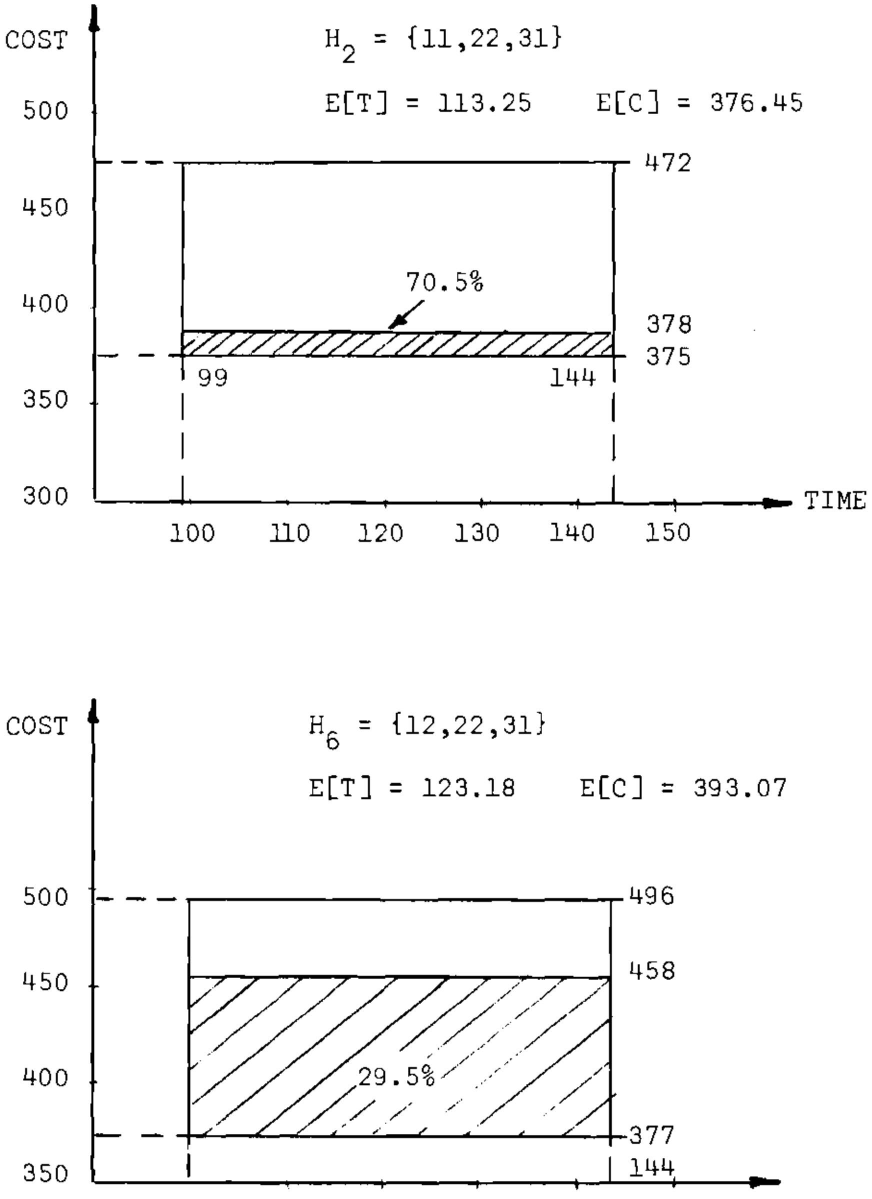


Figure 75. Region of Possible Outcomes and Probability Zones for Opening Policy H_i: Minimum Cost Criterion

APPENDIX E

EFFICIENT SET ALGORITHM: AN EXAMPLE

This appendix includes an example of a six-stage DP model of the type shown in Fig. 31 of Chapter IV. The efficient set algorithm-time version is used. Table 16 shows the return matrices of this model, along with $Q_i(X_i,Y_i)$, $\Theta_i(X_i,Y_i)$ and $\Delta\Theta_i(X_i,Y_i)$, $\Delta Q_i(X_i,Y_i)$. The minimum cost solution (not shown) of this problem yields:

$$T_{C*} = 56$$

$$C* = 99$$

The efficient set tableau is shown in Table 17. Note that all starred elements are circled. For ease of reading the table, only elements conforming to condition b(4) of step 7 of the algorithm are crossed.

Table 18 shows the admissible point test, and Fig. 76 illustrates the efficient set.

Table 16. DP Model Matrices

$T_i(X_i,Y_i)$; $Q_i(X_i,Y_i)$

Stage 6			
1	2	3	4
+ 8 23	+ 6 24	+ 6 29	+ 9 30
31	30	35	39

Stage 5				
	5	6	7	8
1	+ 5 18	+ 10 21	+ 8 24	+ 7 27
2	+ 6 18	+ 6 21	+ 9 24	+ 5 27
3	+ 12 18	+ 9 21	+ 5 24	+ 7 27
4	+ 12 18	+ 14 21	+ 10 24	+ 6 27

Stage 4				
	9	10	11	12
5	+ 4 14	+ 7 18	+ 6 15	+ 4 22
6	+ 15 14	+ 8 18	+ 6 15	+ 10 22
7	+ 11 14	+ 14 18	+ 9 15	+ 8 22
8	+ 14 14	+ 15 18	+ 12 15	+ 10 22

Stage 3				
	13	14	15	16
9	+ 6 11	+ 14 9	+ 3 11	+ 6 10
10	+ 9 11	+ 11 9	+ 7 11	+ 10 10
11	+ 8 11	+ 9 9	+ 6 11	+ 5 10
12	+ 12 11	+ 13 9	+ 14 11	+ 15 10

Stage 2				
	17	18	19	20
13	+ 7 4	+ 8 6	+ 10 8	+ 5 10
14	+ 5 4	+ 9 6	+ 6 8	+ 6 10
15	+ 8 4	+ 5 6	+ 8 8	+ 12 10
16	+ 6 4	+ 10 6	+ 14 8	+ 12 10

Stage 1				
	21	22	23	24
17	4	6	7	5
18	9	6	8	15
19	8	9	14	10
20	10	12	15	13

$C_i(X_i,Y_i)$; $\Theta_i(X_i,Y_i)$

+ 21 117	+ 24 127	+ 22 102	+ 16 142
138	151	124	158

1	+ 20 97	+ 17 100	+ 50 80	+ 25 93
2	+ 30 97	+ 35 100	+ 40 80	+ 25 93
3	+ 18 97	+ 15 100	+ 22 80	+ 30 93
4	+ 45 97	+ 30 100	+ 25 80	+ 20 93

5	+ 11 86	+ 15 84	+ 12 68	+ 30 91
6	+ 25 86	+ 18 84	+ 32 68	+ 40 91
7	+ 20 86	+ 15 84	+ 12 68	+ 30 91
8	+ 42 86	+ 18 84	+ 25 68	+ 12 91

9	+ 34 55	+ 18 55	+ 32 54	+ 20 50
10	+ 25 55	+ 24 55	+ 30 54	+ 28 50
11	+ 32 55	+ 42 55	+ 28 54	+ 18 50
12	+ 24 55	+ 36 55	+ 42 54	+ 28 50

13	+ 25 30	+ 50 32	+ 15 30	+ 20 12
14	+ 25 30	+ 40 32	+ 35 30	+ 30 12
15	+ 30 30	+ 22 32	+ 15 30	+ 18 12
16	+ 20 30	+ 25 32	+ 30 30	+ 45 12

17	30	12	15	10
18	40	32	18	25
19	30	12	15	20
20	12	25	18	42

$\Delta\Theta_i(X_i,Y_i)$
$\Delta Q_i(X_i,Y_i)$

1	0	0 8	13 9	1 11
2	0	10 3	-7 9	-9 8
3	13 1	12 1	0	21 5
4	0	-10 5	-37 4	-29 3

5	0	2 7	-17 3	24 8
6	11 8	-2 5	0	31 1
7	26 1	19 8	0	41 6
8	35 1	9 6	0	10 5

9	3 3	-13 9	0	-16 2
10	-4 2	-5 2	0	-6 2
11	19 4	29 3	14 2	0
12	-12 1	0	5 3	-13 3

13	0	27 3	-10 7	-23 4
14	0	17 6	10 5	-13 7
15	6 1	0	-9 5	-24 11
16	0	7 6	10 12	7 12

17	0	-18 2	-15 3	-20 1
18	8 3	0	-14 2	-7 9
19	0	-18 1	-15 6	-10 2
20	0	13 2	6 5	30 3

Table 17. Efficient Set Tableau

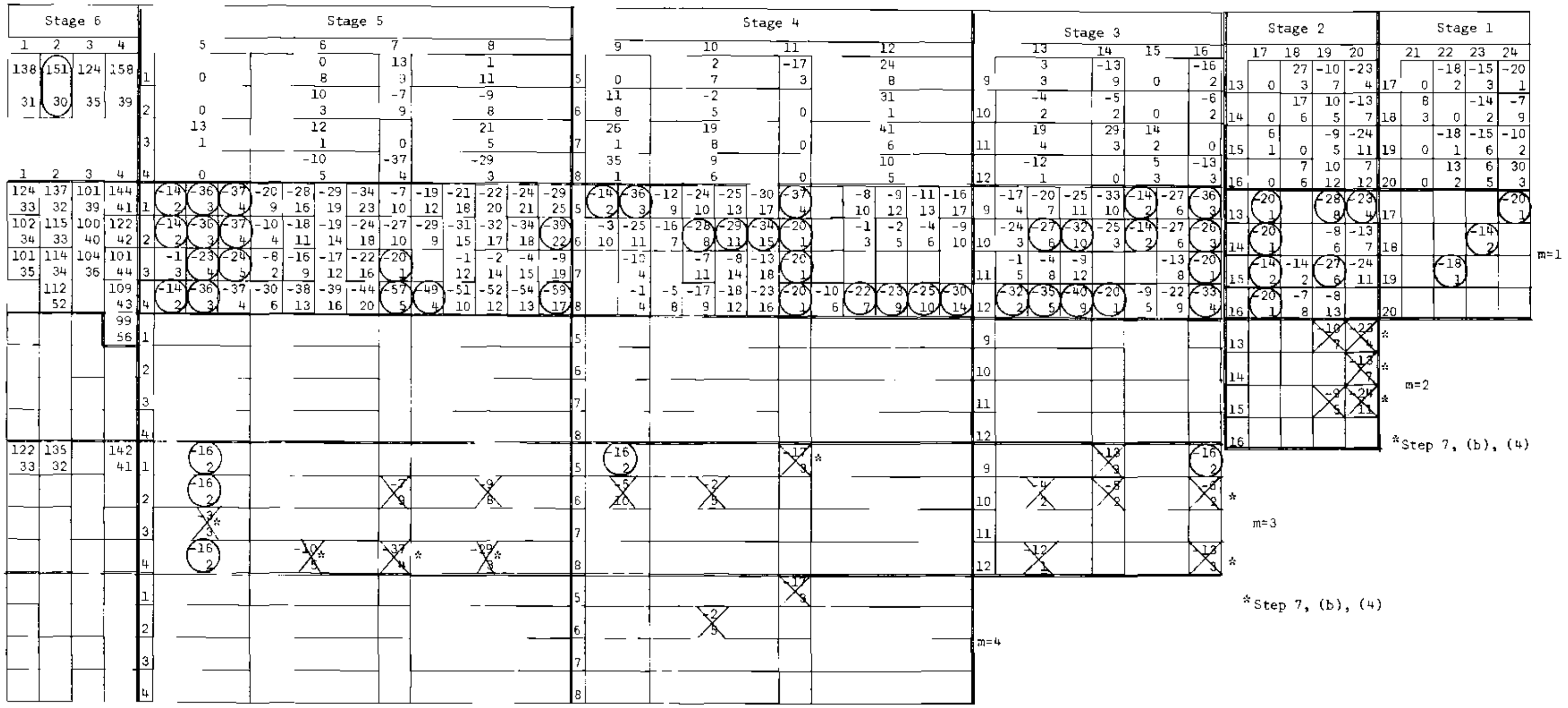


Table 18. Admissible Point Test

	T	C	o_j^*	W_e
1	30	151	(30,151)	{2,5,9,15,18,22} = U^*
2	31	138	(31,138)	{1,5,9,15,18,22} = W_1
3	32	135	(32,135)	{2,5,9,16,17,21} = W_2
4	32	137		
5	33	115	(33,115)	{2,5,9,16,17,24} = W_3
6	33	122		
7	33	124		
8	34	102	(34,102)	{1,5,9,16,17,24} = W_4
9	34	114		
10	35	101	(35,101)	{1,5,11,16,17,24} = W_5
11	35	124		
12	36	104		
13	39	101		
14	39	158		
15	40	100	(40,100)	{3,5,11,16,17,24} = W_6
16	41	142		
17	41	144		
18	42	122		
19	43	109		
20	44	101		
21	52	112		
22	56	99	(56,99)	{4,8,12,13,19,22} = V^*

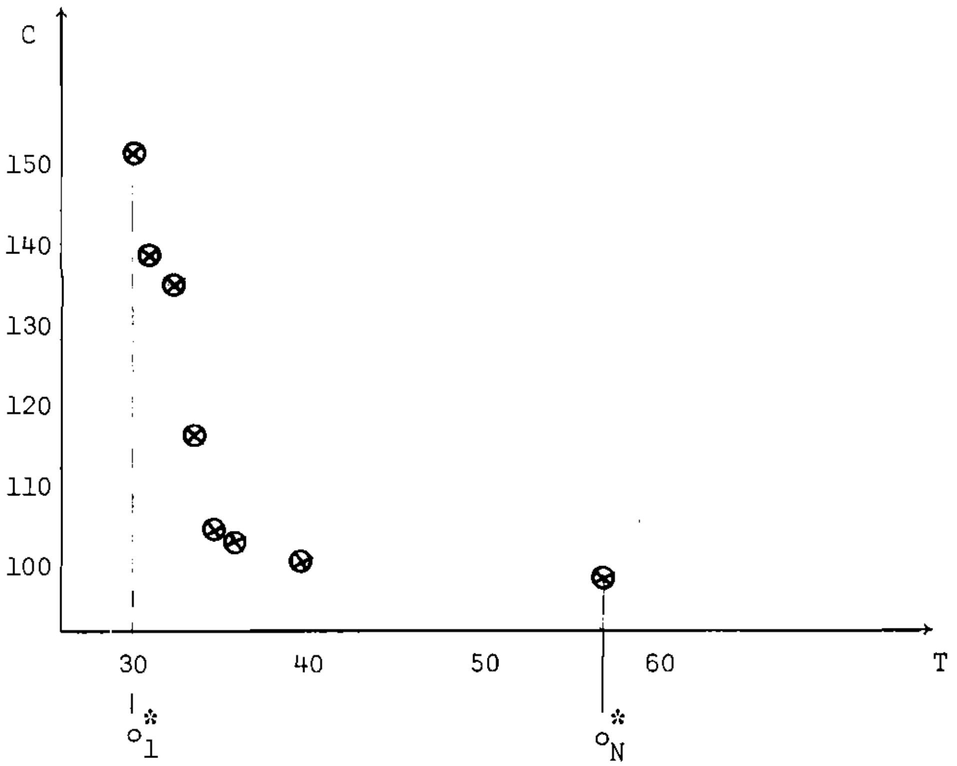


Figure 76. The Efficient Set

APPENDIX F

NOTATIONS AND ABBREVIATIONS

A - set of arcs.

$B = (b_{ij})$ - the predecessor matrix.

C - project cost.

C^* - minimum project cost.

C_i - project ith cost value.

C^i - cost of DDS_i.

C_{T^*} - minimum cost of the minimum time solution.

C_{PN} - cost of the permanent nodes.

$\underline{\underline{C}} = (c_{j|i})$ - m×m cost matrix associated with the Markov Process.

$\underline{\bar{C}}_n = \{\bar{c}_n(i)\}$ - expected total cost return vector - N×1.

$C_i(X_i,Y_i)$ - cost matrix for stage i.

$C_i(X_i,Y_i,\Delta^i)$ - cost matrix for stage i for the stochastic outcomes network.

(CP) - critical path.

CDDS - conjunctive decision dependent subnetwork.

c_i - cost associated with activity node m_i.

$c_{jk(j)}$ - cost associated with decision node $d_{jk(j)}$.

$c_{jk(j)l(k)}$ - cost associated with outcome node $\delta_{jk(j)l(k)}$.

DDS_i - the ith decision dependent subnetwork.

$D = \{D_j\}$ - the set of decision vertices.

$D_j = \{d_{jk(j)}\}$ - the jth decision vertex (decision set).

$d_{jk(j)}$ - the k(j)th decision node (decision alternative) of decision vertex D_j.

E^2 - a two-dimensional space.

$E[\ ]$ - expected value.

$E[\ ]^*$ - minimum expected value.

F - the last node of the network.

$f_{i(t)}(X_i)$ - the minimum time at stage i as a function of the input variable.

$f_{n(t)}(X_n)$ - minimum n stage time return.

$f^i_{n(t)}(X_n)$ - minimum n stage time return for DDS_i.

$f_{i(c)}(X_i)$ - the minimum cost at stage i as a function of the input variable.

$f_{n(c)}(X_n)$ - minimum n stage cost return.

$f^i_{n(c)}(X_n)$ - minimum n stage cost return for DDS_i.

$\bar{f}_{..}(\ )$ - expected values for all f's above.

$G(\ )$ - a directed network.

$g_i(X_i,Y_i)$ - state transformation at stage i.

H_i - the ith opening policy.

$J = \{j_i\}$ - set of nodes of a decision network.

$J^{(c)}$ - set of nodes of CDDS.

$J^{(p)}$ - set of nodes of PRN.

$J^{(s)} = \{j_i^{(s)}\}$ - set of nodes of a DDS.

J^* - set of nodes of the standard network.

$L(Z_k)$ - the length of the kth path from S to F.

$M = \{m_i\}$ - set of activity nodes.

N - set of permanent nodes.

$O = \{o_j\}$ - set of all possible outcomes.

$O^* = \{o_j^*\}$ - the efficient set.

$P(\ )$ - probability of . . .

$\underline{\underline{P}} = \{p_{j|i}\}$ m×m transition matrix (stochastic matrix) associated with the Markov process.

$\underline{\underline{P}}^i$ - n×r stochastic matrix of stage i.

PRN - partially reduced network.

$p_{jk(j)}$ - the probability associated with decision node $d_{jk(j)}$.

$p_{j|i}$ - the conditional probability that a system which now occupies state i will occupy state j after its next transition.

$p_{jk(j)l(k)}$ - probability associated with outcome node $\delta_{jk(j)l(k)}$.

$Q_i(X_i,Y_i) = (q_{ijk})$ - the i-stage time matrix.

$\bar{Q}_i(X_i,Y_i)$ - m×r i-stage expected time matrix.

$Q_i(X_i,Y_i,\Delta^i)$ - i-stage time matrix for the stochastic outcomes network.

$\Delta Q_i(X_i,Y_i) = (\Delta q_{ijk})$ - the time increment matrix for stage i.

$R_i(X_i,Y_i) = (r_{ijk})$ - the i-stage cost matrix.

$R_i(X_i,Y_i,\Delta^i)$ - the i-stage cost matrix for the stochastic outcomes network.

$\bar{R}_i(X_i,Y_i)$ - m×r i-stage expected cost matrix.

RCN - reduced cost matrix.

S - the first node of the network.

T - project time.

T^* - minimum project time.

T_i - project ith time value.

T^i - time of DDS_i.

T_{C^*} - time of minimum cost solution.

$\underline{\underline{T}} = (t_{j|i})$ - m×n time matrix associated with the Markov process.

$\bar{\underline{T}}_n = (\bar{t}_n(i))$ - expected total time return vector - N×1.

$T_i(X_i,Y_i)$ - time matrix for stage i.

$T_i(X_i,Y_i,\Delta^i)$ - time matrix for stage i for the stochastic outcomes network.

(T_p,C_p) - time-cost pair of a promising point.

t_i - duration associated with activity node m_i.

$t_{jk(j)}$ - duration associated with decision node $d_{jk(j)}$.

$t_{jk(j)l(k)}$ - duration associated with outcome node $\delta_{jk(j)l(k)}$.

U_i^* - optimal minimum time policy for DDS_i.

U^* - optimal minimum time policy for the whole project.

$\bar{U}_i^*$ - optimal minimum expected time strategy for DDS_i.

$\bar{U}^*$ - optimal minimum expected time strategy for the whole project.

$\bar{\bar{U}}^*$ - curtailed strategy for the minimum time criterion.

$V_i^*,V^*,\bar{V}_i^*,\bar{V}^*,\bar{\bar{V}}$ - same as above for cost values.

W_e^i - the policy associated with the pair (T_e^i,C_e^i).

W_e - the policy associated with (T_e,C_e).

$X_i = \{x_{ij}\}$ - state variable of stage i.

$Y_i = \{y_{ij}\}$ - decision variable of stage i.

Δ^i - the set of random variables of stage i.

$\Delta^i_{jk(j)} = \{\delta^i_{jk(j)l(k)}\}$ - the random variable associated with decision node $d_{jk(j)}$ at stage i.

$\delta_{jk(j)l(k)}$ - outcome node associated with decision node $d_{jk(j)}$.

$\Theta_i(X_i,Y_i) = (\theta_{ijk})$ - the i-stage cost matrix of the minimum time solution.

$\Delta\Theta_i(X_i,Y_i) = (\Delta\theta_{ijk})$ - the cost increment matrix of the minimum time solution.

$\Omega_i^m(X_i,Y_i) = (\omega_{ijk_r}^m)$ - the time change matrix for stage i when the procedure starts at stage m.

$\Phi_i^m(X_i,Y_i) = (\phi_{ijk_r}^m)$ - the cost change matrix for stage i, when the procedure starts at stage m.

BIBLIOGRAPHY

1. Alpert, L., Orkand, D. S., *A Time-Resource Trade-Off Model for Aiding Management Decisions,* Operations Research Inc. Technical Paper No. 12, Silver Springs, Maryland, 1962.

2. Balas, E., "Minimaximal Path in a Disjunctive PERT Network," *Theory of Graphs International Symposium,* Rowe, 1966, Gordon & Breach, N. Y., 1967.

3. Carruthers, J. A., Battersby, A., "Advances in Critical Path Methods," *Operational Research Quarterly,* Vol. 17, 1966, pp. 359-80.

4. Charnes, A., Cooper, W. W., "A Network Interpretation and a Directed Subdual Algorithm for Critical Path Scheduling," *Journal of Industrial Engineering,* Vol. 13, No. 4, 1962.

5. Charnes, A., Thompson, G. L., "Critical Path Analysis Via Chance Constrained and Stochastic Programming," *Operations Research*, Vol. 12, 1964, pp. 460-470.

6. Chilcolt, J. F., Thursfield, P., "Network Planning Under Uncertainty: Some Application in Research Management," Operational Research Society Annual National Conference, Shrivenham, U. K., 1965.

7. Chilcolt, J. F., "Probability Networks: Their Application to Budgeting for Research," A paper presented at a meeting of the Operational Research Society at the British Coal Utilization Research Association on Jan. 29, 1965.

8. Clark, C. E., "The Greatest of a Finite Set of Random Variables," *Operations Research,* Vol. 9, 1961, pp. 145-162.

9. Clark, C. E., "The PERT Model for Distribution of an Activity Time," *Operations Research,* Vol. 10, 1962, pp. 405-406.

10. Clark, J. Lawrence, "Simulation in Capital Investment, *Journal of Industrial Engineering,* Vol. XIX, No. 10, 1968.

11. Crowstone, W. B., Thompson, G. L., "Decision CPM: A Method for Simultaneous Planning, Scheduling and Control of Projects," *Operations Research,* Vol. 15, No. 3, 1967, pp. 407-426.

12. Crowston, W. B. S., *Decision Network Planning Models*, Ph.D. Dissertation, Carnegie Mellon University, 1968.

13. Davis, J. G., *The Nonsimultaneity Constraint in Network Based Project Management Systems*, Ph.D. Dissertation, Georgia Institute of Technology, 1967.

14. Dean, Burton V., "Stochastic Networks and Research Planning," Proceedings of the 4th International Conference on Operational Research, 1966.

15. *DOD/NASA Guide to PERT/COST Systems Design*, Office of the Secretary of Defense and the National Aeronautics and Space Administration, June, 1962.

16. Drezner, S. M., Pritsker, A. A. B., "Network Analysis of Countdown," Rand Memorandum RM-4976, March, 1966.

17. Eisner, H., "A Generalized Network Approach to the Planning and Scheduling of a Research Program," *Operations Research*, Vol. 10, Jan.-Feb., 1962.

18. Elmaghraby, S. E., "On Generalized Activity Network," *Journal of Industrial Engineering*, Vol. 17, 1966, pp. 621-631.

19. Elmaghraby, S. E., "An Algebra for the Analysis of Generalized Activity Network," *Management Science*, Vol. 10, 1964, pp. 494-514.

20. Elmaghraby, S. E., "On the Expected Duration of PERT Type Networks," *Management Science*, Vol. 13, No. 5, Jan., 1967.

21. Ford, L. R., Fulkerson, D. R., *Flows in Networks*, Princeton University Press, 1962.

22. Fulkerson, D. R., "A Network Flow Computation for Project Cost Curves," *Management Science*, Vol. 7, 1961, p. 167.

23. Fulkerson, D. R., "Expected Critical Path Lengths in PERT Networks," *Operations Research*, Vol. 10, 1962, pp. 808-817.

24. Gessford, G. N., *An Investigation of the Utility of Critical Path Time and Cost Analysis as an Aid to Budgeting and Administrative Control by Construction Firms*, Ph.D. Thesis, University of California, L. A., 1966.

25. Graham, P., "Profit Probability Analysis of R&D Expenditures," *Journal of Industrial Engineering*, Vol. 16, 1965, pp. 186-191.

26. Grubbs, F. F., "Attempts to Validate Certain PERT Statistics, or: Picking on PERT," *Operations Research*, Vol. 10, 1962, pp. 912-915.

27. Hartley, H. O., Worham, A. W., "A Statistical Theory for PERT Critical Path Analysis, *Management Science,* Vol. 12, 1966, pp. 469-81.

28. Hicks, Philip E., Jain, Suresh K., "Project Management Using GPSS/360," Department of Industrial and Systems Engineering, Florida University, Clearing House, AD-688-954, May, 1969.

29. Hespos, R. F., Strassman, P. A., "Stochastic Decision Trees for the Analysis of Investment Decisions," *Management Science,* Series B, Vol. 11, No. 10, 1965, pp. 244-259.

30. Hess, S. W., Quigley, H. A., "Analysis of Risk in Investments Using Monte Carlo Techniques," Chemical Engineering Progress Symposium Series, Vol. 59, No. 42, 1963.

31. Howard, R. A., *Dynamic Programming and Markov Processes,* The MIT Press, Cambridge, Mass., 1960.

32. Kaufman, A., *Graphs, Dynamic Programming and Finite Games,* Academic Press, New York, 1967.

33. Kelly, J. E., Walker, J. R., "Critical Path Planning and Scheduling," Proceedings, Eastern Joint Computer Conference, Boston, 1959, pp. 160-173.

34. Kelly, J. E., Walker, M. R., *Critical Path Planning and Scheduling--An Introduction,* Mauchly Associates Inc., Ambler, Pa., 1959.

35. Kelly, J. E., "Critical Path Planning and Scheduling--Mathematical Basis," *Operations Research,* 1961, pp. 296-320.

36. King, W. R., Wittevrangel, D. M., Hezel, K. D., "On the Analysis of the Critical Path Time Estimating Behavior," *Management Science,* Vol. 14, 1967, pp. 79-84.

37. King, W. R., Wilson, T. A., "Subjective Time Estimates in Critical Path Planning--A Preliminary Analysis, *Management Science,* Vol. 13, 1967, pp. 307-320.

38. Kleinschmidt, A. C., Moore, W. R., Tamashunas, V. M., "Note on Introduction of Cash Flow Into Critical Path," *International Journal of Production Research,* Vol. 3, No. 2, 1964, pp. 85-90.

39. Krishnamoorthy, M., "Critical Path Method--A Review," Technical Report No. 1968-4, Department of Industrial Engineering, The University of Michigan, Ann Arbor, Mich., 1968.

40. Levy, F. K., Thompson, G. L., Wiest, J. D., "Mathematical Basis of the Critical Path Method," In *Industrial Scheduling* by Muth and Thompson, Prentice-Hall, 1963.

41. MacCrimmon, K. R., Ryavec, C. A., "An Analytical Study of the PERT Assumptions," *Operations Research,* Vol. 12, No. 1, 1964.

42. MacCrimmon, K. R., Ryavec, C. A., "An Analytical Study of the PERT Assumptions," Rand Memorandum RM-3408-PR, 1962.

43. Magee, John F., "Decision Trees for Decision Making," *Harvard Business Review,* July-August, 1964.

44. Magee, John F., "How to Use Decision Trees in Capital Investment," *Harvard Business Review,* September-October, 1964.

45. Malcom, D. G., Roseboom, J. H., Clark, C. E., Fazar, W., "Application of a Technique for Research and Development Program Evaluation," *Operations Research*, Vol. 7, 1959, pp. 646-669.

46. Malcolm, D. G., "Reliability Maturity Index (RMI)--An Extension of PERT into Reliability Management," *Journal of Industrial Engineering,* January-February, 1963, pp. 3-12.

47. Martin, J. J., "Distribution of the Time Through a Directed Acyclic Network," *Operations Research,* Vol. 13, 1965, pp. 46-66.

48. Meyer, W. L., Shaffer, L. R., "Extending CPM for Multiform Project Time Cost--Curves," *Journal of the Construction Div. Proceedings of the American Society of Civil Engineering,* Vol. 91, No. CO14319, May, 1965, pp. 45-67.

49. Moder, J. J., Philips, C. R., *Project Management with CPM and PERT,* Reinhold Publishing Corp., N. Y., 1964.

50. Murray, J. E., "Considerations of PERT Assumptions," *IEEE Transactions,* EM-10, 1963, pp. 94-99.

51. Nemhauser, George L., *Introduction to Dynamic Programming,* John Wiley & Sons Inc., N. Y., 1966.

52. Parikh, S. C., Jewell, W. S., "Decomposition of Project Networks," *Management Science,* Vol. 11, January, 1965.

53. Phillips, C. R., "Fifteen Key Features of Computer Programs for CPM and PERT," *Journal of Industrial Engineering,* Vol. 15, January-February, 1964.

54. Pritsker, A. A. B., Whitehouse, G. E., "GERT: Graphical Evaluation and Review Technique--Part II," *Journal of Industrial Engineering,* Vol. XVIII, No. 6, 1966.

55. Pritsker, A. A. B., Happ, W. W., "GERT: Graphical Evaluation and Review Technique--Part I," *Journal of Industrial Engineering,* Vol. XVIII, No. 5, 1966.

56. Pritsker, A. A. B., "GERT: Graphical Evaluation and Review Technique," Rand Memorandum RM-4973, 1966.

57. Roper, D. E., "Critical Path Scheduling, *Journal of Industrial Engineering,* Vol. 15, No. 2, 1964, pp. 51-59.

58. Trilling, D. R., "Job Shop Simulation of Orders that are Networks," *Journal of Industrial Engineering,* Vol. 17, 1966, pp. 59-71.

59. Van Slyke, R. M., "Monte Carlo Methods and the PERT Problem," *Operations Research,* Vol. 11, 1963, pp. 839-60.

60. Van Slyke, R., "Uses of Monte Carlo in PERT," Rand Memorandum, RM-3367, 1963.

61. Whitehouse, G. E., *Extensions, New Developments and Applications of GERT,* Ph.D. Dissertation, Arizona State University, 1966.

62. Wolfe, P. M., *A Zero One Programming Approach to Multiproject Scheduling with Limited Resources,* Ph.D. Dissertation, Arizona State University, 1968.

VITA

Daniel Sipper was born in Tel Aviv, Israel, where he attended elementary and high schools. He received his B.Sc. (cum laude) in Mechanical Engineering from the Technion-Israel Institute of Technology in 1956, and a Diplom Ingenieur degree from the same Institute in 1958. He obtained his M.S. in Mechanical Engineering from Columbia University, New York, in 1960, and also received a Graduate Diploma in Business Administration from the Hebrew University of Jerusalem in 1967.

During the period 1956-1967 his experience includes engineering design, production control, industrial construction supervision, contract administration, engineering and economic evaluation of new industrial projects, feasibility studies and management system analysis with various organizations both in Israel and the United States. During this period he was also affiliated, on a part-time basis, with the Technion-Israel Institute of Technology.

In Fall 1967, Daniel Sipper joined the faculty of the School of Industrial and Systems Engineering of the Georgia Institute of Technology, and at the same time entered the Ph.D. program of the same school. He completed the degree requirements in the Fall of 1970.

He is married and has two sons.