

UNCLASSIFIED

AD 454007

DEFENSE DOCUMENTATION CENTER

FOR

SCIENTIFIC AND TECHNICAL INFORMATION

CAMERON STATION ALEXANDRIA, VIRGINIA



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

DDC AVAILABILITY NOTICE

Questions may be referred to the DDC for copies of this report from DDC.

November 1964

Technical Report No. 47

CATALOGED BY DDC

AS AD No. 4848937

**METHODS FOR EXTENDING THE RANGE
of
NON-COMPUTER CRITICAL PATH APPLICATIONS**

JOHN W. FONDAHL

Associate Professor of Civil Engineering

**Prepared Under Contract No. NBy-45818
for the Bureau of Yards and Docks
U.S. Navy**

**Distributed by
THE CONSTRUCTION INSTITUTE**



**Department of CIVIL ENGINEERING
STANFORD UNIVERSITY**

454007

**METHODS FOR EXTENDING THE RANGE
of
NON-COMPUTER CRITICAL PATH APPLICATIONS**

**John W. Fondahl
Associate Professor of Civil Engineering**

**Prepared Under Contract No. NBy-45818
for the Bureau of Yards and Docks
U.S. Navy**

**Department of Civil Engineering
Stanford University
Stanford, California
November, 1964**

Department of Civil Engineering
Stanford University
Stanford, California

© 1964 by John W. Fondahl

All rights reserved

Printed in the United States of America

Reproduction in whole or in part is permitted
for any purpose of the United States Government.

INDEX

Chapter		Page
	SUMMARY	1
I	INTRODUCTION	3
	Relation to Previous Report	3
	Realization of Objectives - Previous Report	3
	Objectives of This Report	8
	Nomenclature Changes	12
	Development of This Report	15
II	NETWORK DIAGRAMMING	18
	Function of the Diagram	18
	Diagramming Systems	19
	Development of Precedence Diagramming	23
	Factors Considered in Choice of Diagramming	
	System - General	25
	Simplicity	25
	Ease of Revision	30
	Numbering System	33
	Adaptability to Computers	36
	Representation of Events	37
	Time-scale Plotting	41
	Current Practice	45
	Constructing the Rough Diagram	46
	Redrawing the Diagram	49
	Sequence-Step Diagrams	50
	Time-Scale Diagrams	53
III	BASIC NETWORK ELEMENTS AND MECHANICS	56
	General	56
	Network Nodes	58
	Network Lines	62
	Lag	64
	Date Convention	66
	Transmission of Scheduling Changes	67

Chapter		Page
	Determination of Critical Path	69
	Lag Changes	72
	Network Interaction Limit (NIL)	73
	Interpretation of Apparent Negative Lag	77
	Concurrent Changes	78
	Determination of Floats	80
	Applications of Network Mechanics	83
IV	AN ANALYSIS OF DATA ELEMENTS AND APPLICATIONS	84
	Evaluation of Scheduling Data	84
	Data Elements	85
	Durations	85
	Early Start Times	85
	Early Finish Times	86
	Late Start Times	87
	Late Finish Times	88
	Free Floats	90
	Total Floats	90
	Lags	91
	Criticality	91
	Data Requirements and Applications - General	92
	Original Planning on Rough Diagram	93
	Master Plan Presentation on Redrawn Diagram	94
	Time-Cost Tradeoffs	99
	Resource Allocation	100
	Field Schedules	100
	Job Control	103
	Summary and Recommendations	103
V	UPDATING PROCEDURES	106
	General	106
	Duration Changes	107
	Rescheduling Changes	114
	Logic Changes	115
	Summary	123
VI	THE DEVELOPMENT AND APPLICATION OF A NETWORK ANALYZER	124
	Analyzer Development - General	124
	Electrical Network	126

Chapter		Page
	Construction and Application of Network	
	Analyzers	128
	Model I	128
	Model II	130
	Model III (Proposed)	137
	Data Tabulation	142
VII	THE USE OF SUBNETWORKS	149
	General	149
	Single Terminal Interfaces Only	149
	Intermediate Interfaces	150
	Dateline Cutoffs - General	160
	Dateline Cutoffs - Procedures	164
	CLOSURE	179

LIST OF ILLUSTRATIONS

Figure No.	Title	Page
1	Diagramming Methods - Arrow Diagramming	22
2	Diagramming Methods - Precedence Diagramming . . .	23
3	Diagram Development	27
4	Overlapping Operations - Precedence Diagram	28
5	Overlapping Operations - Arrow Diagram	28
6	Illustration of Logic Error	29
7	Illustration of Numbering Problem	34
8	Precedence Diagram with Event Nodes	38
9	Additional Event Requirement for Event-Labeled Arrow Diagram	39
10	Double Labeling for Event-Labeled Arrow Diagram . .	41
11	Time-Scaled Arrow Diagram	44
12	Time-Scaled Precedence Diagram	44
13	Initial Rough Network Diagram	49
14	Sequence Step Network Diagram	51
15	Illustrative Example - Original Job Plan	69
16	Effect of Duration Increase of Activity No. 2 . . .	69
17	Effect of Rescheduling Activity No. 5	71
18	Effect of Duration Increase Equal to NIL	74
19	Effect of Duration Decrease Equal to NIL	76
20	Effect of Further Duration Decrease of Activity No. 2	78
21	Computation of Total Floats - Single Following Activity	81
22	Computation of Total Floats - Multiple Following Activities	81
23A	Node Data - Sheet 1	96
23B	Node Data - Sheet 2	97
24	Sequence Line Data - Sheet 1	98
25	Summary of Data Requirements	104

Figure No.		Page
26A	Updating for Duration Change - Cycle 1 (Cycle 2 Similar)	108
26B	Updating for Duration Change - Cycle 3	112
27	Updating for Scheduling Change	115
28	Updating for Deletion of An Activity	117
29	Updating for Addition of An Activity	119
30A	Updating Involving Multiple Changes - Step 1 . . .	121
30B	Updating Involving Multiple Changes - Step 2 . . .	121
30C	Updating Involving Multiple Changes - Step 3 . . .	122
31	Network Analyzer Model I - Front View	129
32	Network Analyzer Model I - Rear View	131
33	Network Analyzer Model II - Exterior View	132
34	Network Analyzer Model II - Interior View and Removable Patchboards	133
35A	Record of Change No. 1	145
35B	Record of Change No. 2	146
35C	Record of Change No. 3	147
36	Detailed Subnet with Interface and Milestone Events	153
37	Condensed Subnetwork	154
38	Integrated Master Network	158
39	Updated Detailed Subnetwork	159
40	Master Project Network	165
41	Subnet A with Interface Tie Nodes	167
42	Subnet A with Dual Value Interface Data	170
43	Subnet A Expanded and Updated	173
44	Subnet A Updated to Show New Critical Path	175
45	Subnet B	177

SUMMARY

This report covers a continuation of the research described in an earlier report entitled "A Non-Computer Approach to the Critical Path Method for the Construction Industry." Its purpose is to present improvements to several of the procedures previously developed and to offer additional methods that will extend the range of the non-computer approach. Chapter I discusses these objectives and certain nomenclature changes.

The decision to retain the diagramming system presented in the earlier report, which is considered by some to be "non-standard," is discussed at length in Chapter II. This is a basic matter since the detail of the procedures to be presented is dependent on the type of network diagram adopted. The deciding factor is that of simplicity. In Chapter III, the basic network elements and the mechanics by which changes are transmitted through the network are examined. Three approaches are proposed for increasing the size of network which can feasibly be updated by non-computer procedures. This ability to update data is not only important as a function by itself, but it is a key step in several other techniques, such as time-cost tradeoffs and resource allocation. Therefore, these approaches are significant in accomplishing the objectives of the report. They are, (1) greater selectivity in data requirements; (2) efficient methods for updating only that data affected by a change rather than recomputing all data; and (3) network breakdown that permits large networks to be reduced to smaller subnets without sacrificing essential properties of the overall diagram.

Chapter IV considers in detail greater selectivity in data requirements and analyzes the various elements of data that can be provided as well as the various applications of this data. Recommendations are made for (1) considerably reducing the amount of data that is to be maintained in an up-to-date condition after each change occurs, and

(2) updating the remainder at much less frequent intervals. Chapter V develops procedures for updating that data that is affected by a change without recomputing the complete set of data. It is based on a strictly manual method. Chapter VI describes an alternative method of updating based on the use of an electrical network analyzer that has been developed under this project to facilitate this work. Chapter VII discusses three methods of network breakdown which make it possible to reduce a large network to a size that is feasible to handle by non-computer procedures. One of them, a new method based on dateline cut-offs, is particularly useful since it permits updating, analysis, and expansion of detail, if desired, for that portion of the project that is to be performed in the immediate future, but does not require complete updating or a similar degree of detail for the remainder of the network.

Having developed new methods for updating critical path network data in this report, further studies are now underway that will apply these methods for the improvement of advanced planning techniques. A future report will present additional non-computer procedures for time-cost tradeoffs, resource allocation, and other potentially valuable cost reduction tools. However, immediate benefits may be achieved by applying the material in this report for the purposes that are discussed, including field office updating, extension of non-computer methods to larger networks, and a much more efficient means of performing the time-cost tradeoff procedures that have been presented in the earlier report previously mentioned.

CHAPTER I

INTRODUCTION

Relation to Previous Report

This report is a continuation of the work that resulted in an earlier report entitled "A Non-Computer Approach to the Critical Path Method for the Construction Industry." The previous work was done under BuDocks Research Contract NBy-17798 and has been distributed as Technical Report No. 9 of the Department of Civil Engineering, Stanford University. To date, some 9,000 copies of this report have been distributed by the University, and many more through military and other channels.

Where it appears important in discussing new subjects to restate material that appeared in the earlier report, this will be done. An opportunity has also been provided to revise the earlier material in cases where there are improved procedures or changes in viewpoint. In general, however, this report is intended as an extension of the previous one, and it is recommended that both be studied together. Since frequent reference will be made to the earlier report, it will be referred to by the abbreviation "Tech. Report 9."

Realization of Objectives - Previous Report

At the beginning of Tech. Report 9, three goals were stated. These were:

- (1) To present a non-computer method for obtaining the benefits of critical path scheduling that it is practical to apply to many of the projects encountered by the construction contractor.
- (2) To develop the possibilities inherent in a step-by-step, manual solution to overcome some of the shortcomings of computer programmed solutions.

- (3) To offer the reader an opportunity to understand the theory and the assumptions of the Critical Path Method by discussing them and presenting a complete solution to an illustrative problem.

It was also stated that the purpose of the report was "to offer a stepping stone between conventional procedures and these more sophisticated practices." The latter referred to the use of computer methods. The extent to which these several goals have been realized affects the objectives of the present report, so that an examination and evaluation is in order before stating new goals.

It appears that objective (1), the presentation of a non-computer method practical to apply to many actual construction projects, has been successfully accomplished. In 1961, when Tech. Report 9 was published, there was very little detailed information readily available concerning the application of critical path techniques. Most articles dwelt on the benefits of these methods without providing useful working information. A few management consulting firms were offering workshops and furnishing instructional manuals to those who participated. However, they were reaching representatives from only a limited group from the industry, these mostly from larger organizations dealing with very complex projects. The Special Projects Office, U.S. Navy, had published detailed reports concerning the application of PERT. However, at that time, PERT offered a probabilistic approach better suited to controlling such undertakings as the Fleet Ballistic Missile Program rather than ordinary construction work.

Tech. Report 9 was made available to industry, and, as mentioned, approximately 9,000 copies have been requested to date and furnished at cost. Several universities have used it as a text and several schools and organizations have adopted it for short courses. Several management consulting firms have based workshop sessions both domestically and abroad on the report. Also, several contractors have incorporated its methods in their company manuals on planning and scheduling procedures. The Corps of Engineers, U.S. Army, has reprinted it for their Fort Belvoir School. These facts indicate that Tech. Report 9 has provided

needed and useful information. They also indicate that objective (3), the opportunity to understand the theory and assumptions of the method through the presentation and discussion of an illustrative problem, has apparently been successful as well.

Two articles¹ summarizing Tech. Report 9 were published in late 1961, and these also have received wide distribution. A later article² in the same publication by the vice-president in charge of construction of Wigton-Abbott Corporation, Plainfield, N. J., stated:

It was not until Professor John Fondahl's article appeared in the November and December, 1961, issues of THE CONSTRUCTOR that we realized the non-computer approach, when used on larger projects, was acceptable to others. His manual opened the way to more and broader applications.

This tends to support the conclusion that objectives (1) and (3) of the previous report have been met.

Objective (2) of Tech. Report 9, the development of the possibilities inherent in a step-by-step, manual solution to overcome shortcomings of computer solutions, has not been as successfully accomplished. This involved the formulation of a procedure for time-cost tradeoffs. The report demonstrated that these procedures need not be as complex as was generally believed, and it demonstrated that computer methods have certain inherent, objectionable features that can be eliminated by step-by-step methods, thus permitting a continuous application of judgment. However, the non-computer procedure that was presented is tedious and cumbersome to apply and is practical only for small networks. Moreover it required the same input data that would be required for computer processing, and the requirement for this amount and type of data is one of the important objections to the computer approach. Finally, the subject of resource allocation, which is actually closely related, was not considered at all in Tech. Report 9.

¹John Fondahl, "Can Contractors' Own Personnel Apply CPM Without Computers?" The Constructor, November and December, 1961.

²Joseph B. Hoffmaier, "Noncomputer CPM: It's Feasible Even on Some Large Projects," The Constructor, December, 1962.

It is true that some successful applications of the time-cost tradeoff procedures as presented in Tech. Report 9 have been reported by contractors. On the other hand, computer application to them does not appear to have made much headway since 1961, and few successful efforts have been reported. Nonetheless, no valid claim can be made for the complete success of objective (2).

The time-cost tradeoff procedures that originated in Tech. Report 9 were described in detail, along with computer procedures, in the very excellent 246-page company manual¹ of Peter Kiewit Sons' Co. The concluding remarks in that manual probably give an accurate evaluation from most contractors' viewpoint when they state:

The manual method described here offers the user a better understanding of the basic theory behind the dynamic phase of CPM than do computer approaches, and it also retains more "judgment-control" by the planner. The use of small, simplified networks and application of a "conventional estimate" start will reduce the computational effort required, but it remains a very tedious and time-consuming routine in which errors can easily occur.

Tech. Report 9, in addition to setting the three goals already discussed, gave as its purpose the function of a "stepping stone" between conventional procedures and more sophisticated (computer-oriented) procedures. There have been many indications that the report successfully fulfilled this function for numerous users. Typical comments from contractors are that they have used the non-computer methods presented in the report, as a means of introducing critical path techniques to their own personnel. Having acquired a familiarity with the principles and value of these techniques, a number of these same contractors have moved on to computer applications for those jobs that justify them.

This point is well made in the article by Mr. Hoffmier of Wigton-Abbott Corporation, already cited.² Following the quotation given earlier, he continues:

¹Robert E. Sargent, "Manual of Planning and Scheduling," pp. 107-120. Peter Kiewit Sons' Co., 1963.

²Joseph B. Hoffmier, "Noncomputer CPM: It's Feasible Even on Some Large Projects," The Constructor, December, 1962.

We then began applying noncomputer CPM to other projects. We realized that the computer is here to stay, however, and our CPM program for Sinclair, although computed manually, is computer-oriented, as all our CPM programs will be in the future. This gives us the option of running the data through the computer, should we desire.

We already are applying noncomputer CPM to jobs even larger than the Sinclair project, and our plans for the future include the extension of CPM techniques to our engineering activities.

Obviously, the manual technique will become less desirable as the projects to which it is applied become more complex and as the number of jobs on which it is used increases. There is a point where a computer becomes desirable. The only problem is that there is considerable disagreement over what that point is.

My own opinion is that the manual system becomes cumbersome when the project involves more than 400 or 500 "arrows," or activities. Others may have a different experience. The chances are, when the contractor is ready for a computer, he will know it.

As a further example, a construction superintendent with the Damon G. Douglas Co., Inc., Newark, N. J., made these statements in an article describing his company's progress in the use of critical path techniques:¹

Douglas's involvement with critical pathing began early this year and deepened steadily as experience proved the method's worth. At first, several jobs, both large and small, were planned and scheduled by the noncomputer approach to CPM suggested by Prof. John Fondahl of Stanford University in his articles which appeared last year in THE CONSTRUCTOR. Dividends were almost immediate.

The author goes on to say that his company subsequently investigated and added computer processing to its operating procedures, subject to the following instructions:

As a rough "rule of thumb" we have decided that projects which would require less than from 25 to 50 activities will not be critically pathed. Projects consisting of from 25 to 200 activities may be scheduled by the Critical Path Method, but the schedules will be computed manually. All larger projects shall be both critically pathed and calculated by a computer.

¹ Damon G. Douglas, Jr., "CPM Helps General, Subs and Suppliers Talk Same Scheduling Language," The Constructor, December, 1962.

Objectives of This Report

In general the objectives of this report are very similar to those set for Tech. Report 9. Where the goals of the previous report were not fully accomplished, as in the case of the development of a practical manual solution for time-cost tradeoffs, further efforts have been made to develop more workable procedures. Where the goals of the previous report were quite successfully accomplished, as in the case of the presentation of non-computer methods for the basic critical path techniques, further efforts have been made to extend the practical range of application of these methods to larger networks. The previous goal of providing the reader a full understanding of the theory and assumptions for all procedures discussed will, of course, be maintained in this report.

A principal difference in the viewpoint of this report is that the less sophisticated, non-computer procedures described here are intended to serve as more than a "stepping stone" between current practice and highly sophisticated computer oriented methods. It is believed that non-computer methods can, and will for some time to come, provide valuable approaches to better planning, scheduling, and control efforts. In some cases they will fulfill these functions by offering the best way to obtain the desired information. In other cases they will provide the base for valuable supplementary procedures that may be used in combination with computer-oriented procedures. This report, then, is concerned less with the use of non-computer methods as temporary, transitional procedures to be tried during the learning process, and more with their use for furnishing useful information and serving as the most practical means of developing this information.

Stated differently it is not the purpose of this report to advocate non-computer methods where computer methods would serve better. Rather its aim is to extend the range of practicality of non-computer applications and, when that range is exceeded, to use them in combination with computer procedures to gain the advantages inherent in both approaches.

It is the author's opinion, supported by many in the construction industry, that the full potential of critical path techniques will not

be realized until they become commonplace working tools on a day-to-day basis at the job office level. It is certainly true that considerable improvement has already been accomplished in job planning and control by the application of these techniques at the outset of jobs and at periodic updating intervals. These efforts are of major importance and should very effectively "skim off the cream" insofar as cost reduction possibilities are concerned. But substantial further improvement and attendant cost reduction will still be possible through continual replanning and critical questioning of the details of job performance during every day of the entire job span. The critical path techniques furnish the mechanics for undertaking this constant replanning and analysis in a systematic manner.

In general, such project replanning involves proposed scheduling changes that may affect project duration, cost, and resource utilization. Often activities may be resequenced in a different order; unexpected opportunities will develop that will permit improving an activity's performance; certain resource units will become idle and their reassignment may permit expediting other activities at little or no increase in costs; and crew sizes may be manipulated to keep the job in better balance without actually interrupting any of the work. Moreover, activities are seldom such discrete entities as the planner, out of practical necessity, indicates on an original network diagram. For example, for most activities there is generally a certain amount of preparatory work that can be performed in advance of its indicated time. Should temporarily idled manpower or equipment be available, this preparatory work can be done on a "fill-in" basis, so that the estimated duration for the main portion of the activity may be considerably reduced. To a similar degree, the finish-up portions of most activities may actually be performed subsequent to commencing following operations. Again, this can reduce the actual time of performance of the main portion of the activity below original estimates.

Statements such as these border on heresy insofar as network diagramming principles are concerned. The criticism, "Why weren't activities broken down sufficiently that no overlapping exists?" will be

raised since such detail is considered fundamental to all critical path techniques. However, it seems only honest to recognize that no job can be broken down to this degree. It is true that the steps in a project can be performed in the theoretical manner indicated by the network diagram, i.e. no activity will be commenced until the preceding one is 100% complete. But it is also true, in reality, that some preparatory or finish-up work, not worth showing as a discrete breakdown on a network chart, may be performed earlier or later than the main activity if there is sufficient reason to do so. Since CPM is only a means to an end, possibilities for cost and time reduction for the overall project offer a sufficient reason for departing from the original plan. This then, is another example of the type of detailed analysis that might be accomplished by daily job-level replanning.

While computer procedures efficiently develop new data showing the results of proposed changes and while they will always play an important role where the network is large or the technique requires extensive calculations, non-computer procedures also can conceivably be of considerable importance. If procedures can be developed that can be applied with a reasonable amount of effort and time, they would offer two advantages highly important to job-site detailed planning.

The first of these advantages would be the ability to develop answers during the course of the same planning session at which alternatives are proposed, rather than breaking the continuity of such planning, and postponing the evaluation of results until a computer run can be made and data returned to the planners. This would appear to be advantageous regardless of whether computer processing requires a delay of several days or on-site facilities demand an interruption of only a couple of hours. It is hardly conceivable that non-computer methods can furnish, within the span of time of a single planning session, the full amount of detail that would be generated by a computer solution to a given problem. But it may be possible to develop the essential information that is necessary to accept or reject a proposal and then to proceed with consideration of further replanning. And it may be possible to develop all of the required new data for the relatively short span of

time between the planning session and the date at which a routine periodic computer updating run is to be made.

The second advantage of non-computer procedures was emphasized in Tech. Report 9. It is the ability to exercise full judgment control at each intermediate step leading to the final answer. For jobs of considerable size it is often claimed that computer processing is essential for the application of procedures as complex as time-cost tradeoffs or resource allocation. The sacrifice of a considerable degree of judgment control and of adopting simplifying assumptions is recognized as an unavoidable compromise. Although it follows that the answers should not be assumed to be the best possible solution, they sometimes are so viewed because they carry the apparent authority of a printout from the electronic computer. Actually it is quite possible that the results obtained by the computer can be improved by the application of manual methods. Moreover, non-computer methods which retain full judgment control at each intermediate step of calculation and do not demand the same degree of simplifying assumption, may possibly be used for the initial development of solutions as well. Again, although it is not practical to furnish the full array of data available through computer processing, the essential information for correct decisions may be developed.

It is the purpose of this report, then, to attempt to develop procedures that will make these functions possible. Perhaps such procedures will also actually prove to be mere "stepping stones" until the day when the services of computers are readily available at job level. There is considerable evidence already of a strong trend in this direction. Time-sharing equipment with remote console stations may make two-way conversation with the computer more practical. TBX communications to a home office computer center permit rapid transmittal of input and output data from and to the job site. Installation of currently available computers at job office level on large projects has been reported. Finally, smaller, less expensive computers are being developed. There are already reports of the use of these small, "friendly" computers with which a planner can talk back and forth in

the matter that would be ideal for all applications of critical path techniques. All in all, however, there still appears to be a considerable gap to be spanned before these friendly computers make their appearance at a cost and with a capacity that will make them standard equipment for job level planning for projects of all sizes.

Nomenclature Changes

Several changes in nomenclature will be made in this report in order to replace some of the terms used in Tech. Report 9 with more appropriate ones. The purpose of these changes is to convert to more descriptive labeling or to be more consistent with the terminology of others and with that generally accepted in practice.

First, references will be to "critical path techniques" rather than to the more restrictive labels such as CPM or PERT. For example, the title of this report includes this more general designation while the title of Tech. Report 9 included the label "Critical Path Method." On the one hand, there are dozens of variations on the original CPM and PERT systems that have been developed by different governmental agencies and private industrial firms and that generally have been given titles consisting of easily remembered combinations of initials. Robert Miller in his recent book on PERT¹ gives 29 such titles in a glossary in the appendix, and his list is admittedly very far from complete. On the other hand, the two basic systems, CPM and PERT, which developed independently but about the same time and with certain basic differences, have tended to merge.

The Department of Defense has apparently settled on the title PERT for all its procedures. However, insofar as construction projects are concerned it has essentially converted PERT to what was originally labeled CPM. Historically, a basic characteristic of PERT was its probabilistic approach using three-time estimates for each activity. Today a handbook description of NASA - PERT² states, "the principal difference

¹Robert W. Miller, "Schedule, Cost, and Profit Control with PERT," McGraw-Hill, 1963.

²National Aeronautics and Space Administration, "NASA PERT and Companion Cost System Handbook," October 30, 1962.

between the basic PERT concepts and NASA - PERT concepts is that NASA - PERT uses a single time estimate for activities in the network whereas basic PERT uses a three-time estimate system." The other basic characteristic of PERT was its emphasis on event labeling rather than activity labeling. This too is being at least considerably modified. NASA and Corps of Engineers specifications which call for PERT network diagrams, while still requiring that certain milestone events be shown, permit or require the labeling of activities.

It remains to be seen whether a single designation will emerge from the present confusion. However, all of these techniques are basically similar and have the same fundamental approach. They depend on the concept of the representation of a job plan by a network diagram, and they include the concept of a critical path through this network which permits classification of the component activities into either critical or non-critical depending on whether or not they fall on the critical path. It is therefore accurate, and at the same time more general, to refer to all of these techniques as "networking techniques" or as "critical path techniques." The author chose to avoid the label "networking techniques" because some writers have used this term and "arrow diagramming" as synonymous and interchangeable. The result is that many might associate networking techniques only with systems based on arrow diagrams. Since this report will use a networking system different than arrow diagramming, a term that might convey the opposite impression has been avoided.

A second major change in nomenclature from that used in Tech. Report 9 concerns the designation of the diagramming method that has been adopted in these two reports. In the previous report this system was referred to as the "circle-and-connecting-line notation" in contrast to the "arrow notation" used more commonly for both CPM and PERT. One writer has referred to it as the "operation and interrelation line" system.¹ Professor Joseph J. Moder of Georgia Institute of Technology

1

Robert C. McLean, "Readers' Comment," Engineering News-Record, April 25, 1963.

and Mr. Cecil R. Phillips of Management Science Atlanta, Inc. have suggested another set of designation¹ that have the merit of providing a more precise classification for other diagramming systems as well. Their suggested labels of "Activity-on-Arrow," "Event-on-Node," and "Activity-on-Node" for the typical systems used respectively in CPM, PERT, and Tech. Report 9 are very descriptive. In the initial draft of this report a tentative decision was made to adopt these labels. However, for reasons explained in Chapter II, this decision was subsequently changed. Recently IBM has announced a project control system for the construction industry² based on a networking system which they label "precedence diagramming." The latter term refers to a diagramming system identical to that described in Tech. Report 9. The designation "precedence diagramming" will be adopted in this report to replace "circle-and-connecting-line diagramming."

The network lines in the precedence diagram are referred to in this report as "sequence lines" rather than "connecting lines." This is a minor change, but the new term is slightly more descriptive. The term "lag," a value associated with network lines, which was first introduced in Tech. Report 9, is retained in this report. Similarly, the associated term "network interaction limit," which also was originated in Tech. Report 9, is retained, although it will frequently be used in the abbreviated form "NIL."

Finally, the Phase I, Phase II, and Phase III designations used in Tech. Report 9 will be replaced by more descriptive terminology. Phase I covered network diagramming and will be referred to here by this term, used in the general sense of including all possible systems of networking rather than arrow diagramming only. Phase II covered the application of time estimates in the computation of scheduling and other time data. These calculations will be referred to in this report as the

¹ Joseph J. Moder and Cecil R. Phillips, "Project Management with CPM and PERT," Reinhold Publishing Corp., 1964.

² Application Programming Announcement, "IBM 1440 Project Control System for the Construction Industry," IBM Data Processing Division, February 28, 1964.

"basic time calculations." Phase III covered a technique that has been commonly labeled the "time-cost tradeoff" procedure, and this term will be used here. In addition, this and other techniques not discussed in detail in Tech. Report 9, such as resource allocation procedures, multi-project scheduling, cost control, etc., will be grouped here as "advanced critical path techniques." These are all potentially valuable cost-reduction tools that have already received considerable attention, but they are not yet widely applied in the field. They all involve extensions of applications based on network diagramming and basic time calculations. As these first two phases of critical path techniques are mastered and successfully implemented on a broad scale, the introduction of these advance techniques will offer additional opportunities for improved planning and cost reduction.

Development of This Report

The material in this report was not developed in the same sequence in which it is presented. A brief background of its evolution may provide some added insight for the reader's understanding.

Following publication of Tech. Report 9, the author maintained a continuing interest in developing more workable procedures for applying the potentially valuable time-cost tradeoff technique. Although the approach offered in that report permitted a manual application and allowed full judgment control at each computational step, it was extremely cumbersome to apply to even small networks. As the number of network activities increased, the point was soon reached at which the computational mechanics required a prohibitive amount of time and effort. It appeared evident that the procedural steps most at fault involved the development of data on two worksheets labeled "Precedence Matrix" and "Network Interaction Limit Determination." Some further study led to the conclusion that this data might be obtained more easily by means of an electrical analyzer employing a patchboard model of the job network. Such an analyzer was built and is described in Chapter VI.

Having designed an analyzer primarily for the purpose of simplifying time-cost tradeoffs, it became apparent that with some additions

the same device would be useful for network updating as well. In the case of time-cost tradeoffs, time changes are made intentionally, while in the case of updating, time changes generally occur unintentionally. In both cases it is necessary to determine the resulting effects, and the procedure is basically the same regardless of the reason for the change. Therefore the analyzer use was broadened to include updating as well as time-cost tradeoffs.

Further efforts led to the realization that procedures could be developed for working directly on the network diagram by methods analogous to the functioning of the electrical circuits of the analyzer. These manual methods would permit updating and time-cost tradeoff calculations for networks considerably larger than were practical under the procedures of Tech. Report 9. These procedures and the basic network mechanics on which they are based are presented in Chapters III and V. The network analyzer still offered a practical means of handling an intermediate range of networks extending from those that were too large for manual processing to those that were beyond the practical capacity of an economical analyzer and patchboard wiring.

Although these efforts extended the range of network size that may be successfully handled by non-computer methods, network size still remained an important limitation on both the basic time calculations and the various advanced critical path techniques. To attempt to make further inroads on this size barrier, some work was performed in the area of network breakdown. By using smaller subnetworks that still represent the essential relationships of the overall master network but that are within the range of non-computer procedures, it became possible to analyze practically all jobs by these methods. Network breakdown is discussed in Chapter VII. It was also noted that the capacity of the network analyzer could be doubled by eliminating certain data output information. The importance of maintaining this data up-to-date at all times was questioned. An evaluation of data requirements, described in Chapter IV, led to the conclusion that it is only essential to keep a limited amount of data up-to-date on a daily basis and that other data may be updated at less frequent intervals.

Further experience in the area of time-cost tradeoffs led to different procedures and viewpoints than those presented in Tech. Report 9. These include the placing of much less emphasis on the development of continuous cost slopes for each activity and greater appreciation for the value of the joint application of time-cost tradeoff and resource allocation procedures. The practicality of applying these techniques to subnetworks rather than the entire master network has also been considered. These new approaches are made possible by the work described in this report. Since they require a detailed presentation, they will be the subject matter of a report now in preparation.

Finally, throughout the period of these investigations, the use of precedence diagrams instead of the widely accepted arrow notation was continually questioned. The author has maintained his opinion that the precedence diagram is preferable. Since the diagramming system is the basis for all other critical path technique calculations, the decision to retain a "non-standard" system is discussed at some length in the next chapter.

CHAPTER II

NETWORK DIAGRAMMING

Function of the Diagram

All critical path techniques require that a job be broken down into its component activities and that the sequential relationships of these activities be stated. Only after this is done can data on time, cost, and other resource requirements be applied to provide further useful information. Therefore the project breakdown and statement of planning logic is a primary and basic step. It is generally accomplished by constructing a network diagram. This network diagram provides a paper model of the plan of action for performing the job. It also can provide, as will be discussed later, a working model on which it is possible to directly determine the effects of changes in planning logic or in performance times on all the activities that are influenced by the changes.

Although the job breakdown and statement of sequential relationships are absolute essentials to the application of critical path techniques, the use of a network diagram is not. An alternative is to merely prepare a list of job activities and for each activity to list those activities that must directly precede it. In effect, a network is established by such a listing even though it may never be presented in diagrammatic fashion. Proper procedures can be formulated for making calculations with time information and other data based on this listing and by either computer or non-computer methods. Information obtained from these calculations can be applied without reference to a diagram.

But even though a network diagram is not an absolute necessity for the application of critical path techniques, the methods presented in this report will require that one be used. It is felt that there is ample justification for this requirement. The job network is not only the most basic requirement in critical path work, but it also is the most

useful single result obtained from it. This is because it forces one to plan a job in detail, and it requires that the planning be set down on paper in specific terms. A network diagram is a logical vehicle for this planning on paper. It facilitates the orderly development of the plan. It clearly conveys the plan to all interested parties both as it is being developed and upon completion. It permits an overall view of the plan and provides a high level of detail for accurate evaluation. These reasons alone seem to justify construction of a network diagram.

However, another important reason for using a network diagram when non-computer methods are to be employed, is that it can furnish a worksheet for making calculations and interpreting results. In this report the diagram is to be used extensively for these purposes. This applies both to the basic time calculations, already developed, and to the new procedures proposed later in this report. In Tech. Report 9 the basic time calculations were made as a separate tabulation, but reference to the network diagram was necessary to obtain precedence relationships (see pages 26-29 and Figures 3, 5A, 5B, 5C, and 5D). Additional experience has indicated that it is generally simpler and faster to make these calculations directly on the diagram.

Changes in planning logic or in time data will require that previous results be updated. These changes may result from a systematic re-planning of the job as will occur in time-cost tradeoff and resource allocation applications, or they may occur unavoidably during the course of job performance. Later chapters in this report will discuss procedures for updating based on revisions by steps performed directly on the diagram. It is true that a separate listing of the results of such revisions may still be desirable. It is often advisable to maintain a record of the effects of each change that is made rather than to erase or otherwise revise a single set of figures. But even though a separate tabulation of results is maintained, the diagram serves as the worksheet on which the new data is developed.

Diagramming Systems

There are several possible systems for network diagramming. Often

a particular system has been specified not because it possesses inherent advantages but because of ignorance of the fact that there are alternatives. The two that are the most common by far are the activity-labeled arrow diagram and the event-labeled arrow diagram. In the Moder and Phillips terminology mentioned in Chapter I, these would be described as the Activity-on-Arrow and the Event-on-Node systems, respectively. The arrow diagram, used by both, is characterized by the representation of a job activity as an arrow. The fact that one activity must follow another is indicated by placing the arrow tail of the following activity at the arrow head of the preceding activity. The common point, or node, at which arrow heads and arrow tails meet is an event. It represents a point in time at which preceding activities have been completed and following ones may commence. The events are numbered. Each arrow connects two events and, therefore, can be designated by a dual number, the first number giving the preceding and the second number the following event. The only difference between these two systems results from a choice of emphasis on either the activities or on the events. In one system the activities are described, and the arrows are labeled accordingly. In the other system the events are described, and the nodes are labeled accordingly.

Historically, the Critical Path Method (CPM) has used the activity-labeled arrow diagram. In construction and maintenance operations where CPM originated, companies usually think of the steps necessary to accomplish a job in terms of the specific activities that must be performed. It might be said that these companies tend to be activity oriented. On the other hand, PERT has historically used the event-labeled arrow diagram. PERT was developed and adopted by governmental agencies to control complex programs where progress "milestones" were customarily established. These agencies have tended to be event oriented.

A third system of network diagramming was used in Tech. Report 9. Its characteristics were that activities are represented by single-numbered nodes and are connected by sequence lines to other nodes representing preceding or following activities. Like the activity-labeled arrow diagram this was an activity-oriented system. It was

called the "circle-and-connecting-line" system. The Moder and Phillips designation for this system was appropriately called the "Activity-on-Node" system.

In this report this system will be extended to include events as well as activities whenever it proves desirable to do so. Sometimes it is useful to show a key event and to develop data concerning it. Certain contract specifications require that important milestone events be shown on network diagrams. Sometimes the separate representation of an event common to a number of activities is a useful device for simplifying an otherwise intricate diagram. Events may also serve as interface connecting points in network breakdown or subnetwork development procedures. Fortunately the Activity-on-Node system lends itself quite simply to the representation of events as well as activities. An event can be considered to be equivalent to an activity of zero time duration and may then be labeled and treated just as an activity. This does lead to certain nomenclature difficulties in using the Moder and Phillips designations. When an event is substituted for an activity in this third system of network diagramming it becomes an "Event-on-Node" element. However, this designation has already been used to refer to an event-labeled arrow diagram which is basically a very different sort of diagram. Therefore, while the author was not satisfied with the designation used in Tech. Report 9 and while the "Activity-on-Node" designation appeared to be an acceptable improvement, the extension of this diagramming system in the present report requires another title.

In an Application Programming Announcement released February 28, 1964, IBM described a new Project Control System which is to become available during the third quarter of 1964. The announcement stated that "The project can be planned using a new technique called precedence diagramming in addition to the normal IJ arrow diagramming." An accompanying Application Program Bulletin¹ gave a brief description of precedence diagrams, comparing them to arrow diagrams and citing their

¹IBM Application Program Bulletin, "IBM 1440 Project Control System," p. 1, IBM Technical Publications Department.

basic advantages over arrow diagrams. Precedence diagramming appears to be the same as circle-and-connecting-line, or activity-on-node, diagramming. Because of the high standard and wide use of IBM equipment it is expected that this designation will be widely used. It therefore seemed logical to adopt the term "precedence diagramming" for the networking system which has already been described in Tech. Report 9 and extended in this report to include events as well as activities.

There is no indication in the IBM bulletin that they intended to include events in their precedence diagram. However, this is a simple matter, as explained above, and it provides a useful generalization to the scope of representation provided by the diagram. Just as "arrow diagramming" is used as a general term for either an activity-labeled or an event-labeled diagram, "precedence diagramming" can be used in the same way. There is a slight difference. An arrow diagram will generally be completely activity-labeled or completely event-labeled, but not mixed. A precedence diagram may be completely activity-labeled, or may have mixed activity-event labeling, but generally will not be completely event-labeled.

The various diagramming methods are illustrated in Figures 1 and 2.

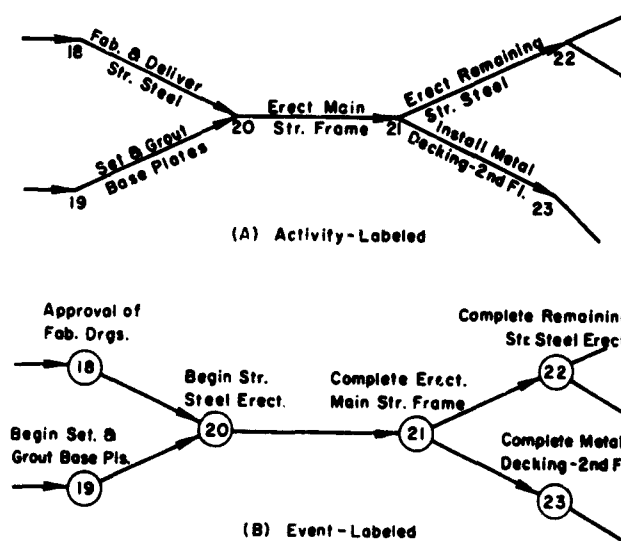


Figure 1. Diagramming Methods - Arrow Diagramming

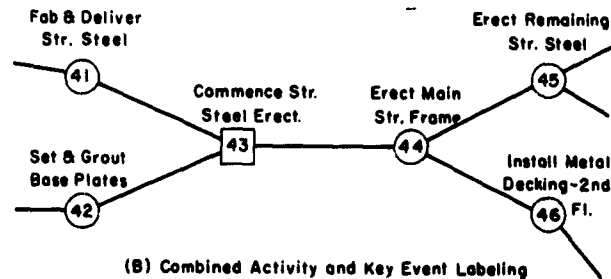
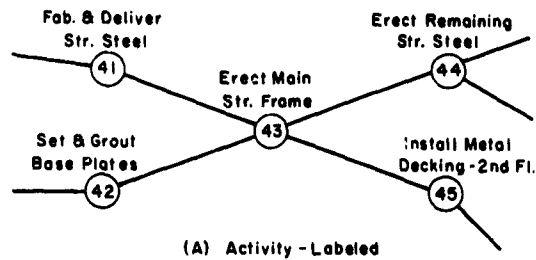


Figure 2. Diagramming Methods - Precedence Diagramming

Development of Precedence Diagramming

The origin of the precedence diagram as defined in the foregoing section is indefinite and undoubtedly pre-dates the introduction of critical path techniques. Considerable attention was given network flow theory before 1958. The author used the precedence diagram in his early scheduling studies, before he had knowledge of any of the critical path techniques, simply because he had been working with industrial engineering flow charts. This led him naturally to a precedence type diagram as contrasted to an arrow diagram.

Although Moder and Phillips in their recent text¹ have used arrow diagramming throughout their main presentation, they discuss the "Activity-on-Node System of Networking" in an appendix and make the following remarks:

The principal advantage of the activity-on-node system

¹ Joseph J. Moder and Cecil R. Phillips, "Project Management with CPM and PERT," Reinhold Publishing Corporation, 1964.

is its simplicity. The avoidance of dummy activities as special devices eliminates most of the problems of networking, especially for beginners.

The disadvantage of using activities-on-nodes as a networking system is primarily that it is non-standard. The other systems greatly overshadow it in general practice. The authors know of only one computer program for this system, as compared to more than 60 for the other systems.

They continue further:

The activity-on-node system has been supported as the best networking system by Professor John Fondahl of Stanford University. Based on his support, the Navy's Bureau of Yards and Docks has specified the use of this system in a number of construction contracts (in which it is called the "circle and connecting-arrow technique").

The authors have used the system in practice and have tested students on their ability to learn it as compared to the other systems. As a result of these comparisons it is the authors' opinion that the activity-on-node system is indeed superior in most respects to the other networking systems.

In another recent book,¹ three authors from Carnegie Institute of Technology have written a chapter entitled "Introduction to the Critical Path Method." This material presented by F. K. Levy, G. L. Thompson, and J. D. Wiest was developed in work that they performed under the support of the Office of Naval Research and the Bureau of Ships. They also use the same diagramming system as that used in Tech. Report 9 rather than arrow diagramming, and follow a description of their approach by the following comments:

The method described above by the present authors avoids the necessity (and complexity) of dummy jobs, is easier to program for a computer, and seems more straightforward in explanation and application.

Finally, with the adoption and promotion of precedence diagramming by IBM, the probability that it will become an accepted and more widely used alternative to arrow diagramming seems assured.

¹"Industrial Scheduling," edited by J. F. Muth and G. L. Thompson, Prentice-Hall Inc., 1963.

Factors Considered in Choice of Diagramming System - General

The construction of the network diagram is the most basic and important step in the application of the critical path techniques to be described in this report. Even though the mechanics of precedence diagramming have been presented in Tech. Report 9, the decision to continue to use this type of diagramming is important enough to require further analysis before proceeding to new topics. This decision was based on a careful consideration of the relative advantages of activity-labeled arrow diagrams, event-labeled arrow diagrams, and precedence diagramming. Factors involved in this comparison included the following:

- (1) Simplicity
- (2) Ease of Revision
- (3) Numbering System
- (4) Adaptability to Computers
- (5) Representation of Events
- (6) Time-scale Plotting
- (7) Current Practice

Each of these topics will be considered in detail below.

Simplicity

Of all of the listed factors, that of simplicity was considered to be of the greatest importance, and it had the most influence on the decision to use the precedence diagramming system. In order to construct a useful and realistic network diagram two kinds of skill are required. The first includes a thorough knowledge of the job, the problems involved in performing it economically, and the methods that may be employed. The second is a knowledge of the mechanics of network diagramming. The first skill requires a high degree of knowledge of construction practices and is not acquired easily. Generally the more closely one is associated with the work to be performed, the more one is capable of contributing here. The second skill may also require a high degree of special competency if network mechanics are complicated. If this is the case an expert may be trained within an organization or hired from a firm specializing in such work. Then the task of developing a diagram may be given to a team composed of one member who understands how

to draw a diagram properly and other members who understand the project that the diagram is to represent. However, this often leads to problems of communication and results that are not completely satisfactory. It also means that those who use the diagrams must call on their diagramming expert when it becomes necessary to make revisions.

As indicated earlier, a principal objective of this report is to permit the use of critical path techniques much more extensively in job planning and analysis. For small jobs, day-by-day job-site pre-plans, subnetworks for detailed breakdown and analysis of single activities from a master network, job planning by subcontractors, and a host of other applications, these techniques can offer far greater value than has resulted from the limited use that they now receive. To achieve this objective, insofar as network diagramming is concerned, the author believes that the best way is to make the mechanics of diagram construction so simple that those who understand the job are also able to diagram it properly. This solution recognizes that by far the more important of the two requisite skills is job knowledge, and that it is very desirable for more job personnel to be able to make and revise a diagram independently.

Precedence diagramming permits this simplification of diagramming mechanics. As compared with either of the two arrow systems (which utilize the same diagram except for labeling), precedence diagramming has two important simplifying advantages. First, it does not require the introduction of dummy activities in order to portray correct network relationships. Second, it does not complicate the plotting of activities by requiring that they be drawn in certain confining positions with respect to one another, i.e. that certain activity arrow heads must meet at a common point and that other activity arrow tails must originate at this same point.

As a very simple example consider a portion of a network diagram being developed to represent the construction of a foundation wall. Concrete forms are to be re-used on the second half of this work. The planner has indicated that these forms may be set after excavation has been completed and after they have been stripped from the earlier

placement. He now desires to show that after the forms are stripped from the location where they were first used, backfill may be placed there. In the case of the precedence diagram, he merely adds a node symbol for an activity in a convenient space, labels it, and draws the sequence line between it and the activity which it follows, as shown in Figure 3A. However, in the case of the arrow diagram, he finds that a dummy activity must also be introduced in order to avoid making the backfill operation for the east wall dependent on the excavation for the west wall to which it, of course, bears no relationship. Furthermore, in order to insert this dummy arrow, the planner must erase, or otherwise relocate, the arrow that he has previously placed on the diagram to represent the form stripping operation. This is shown on Figure 3B. The fact that the precedence diagram does not require a dummy activity illustrates the first of the advantages cited in the preceding paragraph. The fact that erasures or changes in the existing portion of the diagram are not required illustrates the second advantage. This latter point will be discussed in more detail in the next section.

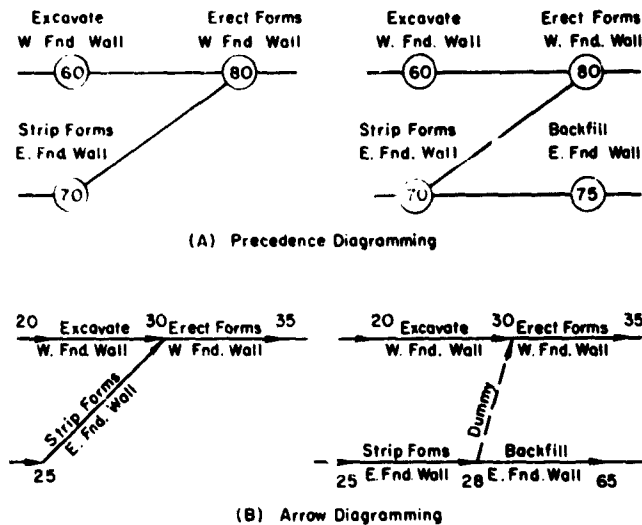


Figure 3. Diagram Development

As another simple example, consider the common case of a job with several overlapping activities. Figure 4 shows a portion of a network

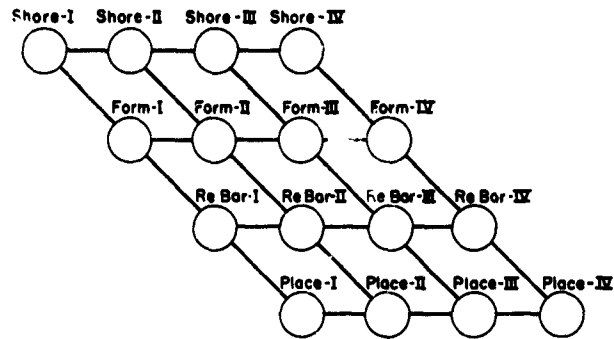


Figure 4. Overlapping Operations - Precedence Diagram

diagram for four spans of an elevated roadway. Each activity on each of the four spans is to be performed in series by a single crew moving from span to span. (No shoring or form reuse is indicated in this simple problem.) Drawing a precedence diagram to represent this work simply requires that sixteen activity nodes, four for each of four spans, be set down on paper, labeled, and connected with the appropriate sequence lines. This may be done quickly and with a minimum of effort or thought given to diagramming mechanics. On the other hand, in order to draw an arrow diagram, a total of twelve dummies must be introduced to show the proper relationships between the sixteen activities. The extra effort necessary to construct Figure 5 cannot be conveyed by just viewing the completed diagram. The reader should perform this exercise or a similar one independently, in order to appreciate the difference in the two systems.

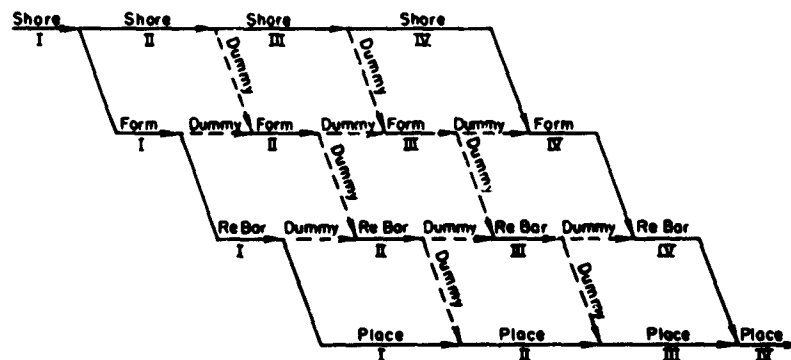


Figure 5. Overlapping Operations - Arrow Diagram

Besides the extra time and effort required to construct the diagram of Figure 5, greater skill is required to avoid errors in logic. If the person drawing the diagram understands the job requirements and the sequencing restraints imposed by job conditions, there is little chance that he will represent these incorrectly by precedence diagramming. There is a much greater chance that the job plan will be incorrectly represented by an arrow diagram. Figure 6 shows a common error for a portion of the diagram of Figure 5. Here dummy activities No. 17-20 and No. 20-23 have been shown because the need for them is quite apparent. However, this diagramming version still does not state the desired relationships properly. Although the intent of dummy activity No. 20-23 is to indicate that the reinforcing steel in Span II cannot be placed until that span has been formed, it actually states that the reinforcing steel in Span II cannot be placed until Event No. 20 has occurred. Event No. 20 does not occur until both the forming of Span II and the shoring of Span III have been completed. Therefore the placing of reinforcing steel in Span II is made dependent on the completion of the shoring of Span III, an activity to which it bears no relation. By similar reasoning, the placing of concrete in Span I has been made dependent upon the completion of forming of Span II and the shoring of Span III, even though the planner did not intend to state these restraints. These errors arise because of the omission of additional dummy activities which were properly included in Figure 5.

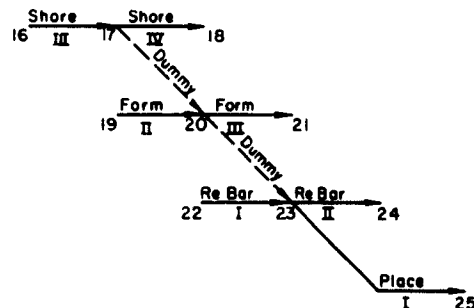


Figure 6. Illustration of Logic Error

In criticism of the precedence diagramming system, advocates of the arrow system point out that sequence lines are in reality the same as dummy activities and, furthermore, that for any given network the number of sequence lines will exceed the number of dummy activities that would be required in an arrow diagram. As an example, the precedence diagram of Figure 4 eliminates the twelve dummy activities required in Figure 5 but, by using sixteen sequence lines between nodes, has actually added back sixteen dummy activities. Technically this point is valid, but it is not a strong argument for diagramming simplicity. Plotting activity nodes and sequence lines is a straightforward procedure that can be explained and comprehended in a few minutes. The principal skill required is a knowledge of the job to be represented. Plotting activity arrows properly requires considerably more knowledge of diagramming technique in addition to a knowledge of the job to be represented. Even when this technique is mastered, it requires more time because the necessity of merging arrow heads and arrow tails at common points leads to frequent shifting and erasures as a diagram is expanded. A novice in network diagramming can sketch Figure 4 quickly and correctly. It will take him much longer to sketch Figure 5, and he will probably make some logic errors. A diagramming expert would sketch Figure 5 correctly, but it would require considerably more time than the corresponding precedence diagram.

These are simple examples. Another simple exercise was presented in Appendix H of the second edition of Tech. Report 9. It has been commented that this latter example was not typical since a small network of eighteen activities required the addition of nine dummy activities to the arrow diagram. In fact, however, this is not as large a percentage of dummies as in the example just cited where there are twelve dummy arrows for a 16-activity job. Nor is the percentage as large as required in some actual projects involving hundreds of activities. For these larger networks the advantages of precedence diagramming in achieving simplicity become even more pronounced.

Ease of Revision

Precedence diagrams are more easily revised than arrow diagrams. This is true for several reasons. If a new activity is recognized and is added to a precedence diagram, the node which represents it may be

entered with a minimum amount of effort and a maximum degree of flexibility. The use of sequence lines permits a great amount of freedom in selecting a favorable location without disturbing the existing diagram. However, there is relatively little flexibility in the addition of an arrow on an arrow diagram. Its ends must terminate at points representing previously plotted job events. Generally a new arrow cannot be added to an arrow diagram without further changes in the location of the previously existing arrows. For example, if two or more existing activities commence at the event-node where the head of the new arrow must terminate, it is often found that at least one of these activities is not dependent on the new activity. Therefore further shifting of network elements and the introduction of a dummy activity is required.

Seldom is a revision in job planning as simple as the mere addition of a single activity. Several activities may be added, others deleted, and a number of changes may be made in the sequential relationships. The flexibility of the precedence diagram permits even complicated changes to be made with little effort. Adding new activities and sequence lines and deleting existing activities and sequence lines is simple and does not affect the positioning of other activities. The inflexibility of the arrow diagram, on the other hand, generally demands a major revision of a sizeable section of the diagram. Not only is more drafting effort required, but also greater skill and knowledge of network mechanics.

Planning changes are inevitable during the course of field operations. The sequencing of activities will be modified either out of necessity or because an alternative approach becomes more favorable. Some activities will be broken down further in order that following activities can be started sooner or to avoid delays because others were not performed on time. It is very desirable that the network diagram be continually updated to reflect these changes and that the reasons for the changes be documented. By far the best place to do this updating is in the field office. The updating of a precedence diagram is within the capability of practically any of the field supervisory

personnel, and with a minimum amount of instruction, probability of errors, and degree of time and effort. This tends to encourage frequent updating that is so desirable if the maximum benefits are to be obtained from critical path techniques. The reverse is true of the arrow diagram. Because it requires more time, effort, and skill to update there is a reluctance to do so on the job and, sometimes, even a hesitancy to permit field office revisions to be made. In extreme cases, the field organization may forego desirable changes in field operations because they do not wish to change the diagram. More frequently they will make the changes in the field performance but will not bother to record their effects.

The work required to revise the activity-labeled arrow diagram is also greater because as the position of arrows is shifted it is necessary to remove and replace the labeling as well. The same revision on a precedence diagram might require shifting of unlabeled sequence lines, but, generally, the labeled nodes can be left undisturbed.

One further advantage of the precedence diagram from the standpoint of ease of revision is that the numbering of activities need not be changed. Even though new activities are added to the diagram, existing activities may retain their previous numbers. This statement assumes that the calculation procedures permit random numbering or, at least, do not require consecutive numbering. However, the change in position of an activity-arrow on an arrow diagram generally requires a change in at least one of the event-nodes at its terminal points. Therefore, the dual number of that activity is changed. If this number is used strictly for routine calculations this poses no problem. However, if the number has been used for identification purposes, then it is necessary to take steps to call the change to the attention of all users. For example, the number may have labeled activities for suppliers or subcontractors, or it may have been incorporated into a cost accounting system. The next section discusses numbering systems in more detail.

The example shown by Figures 3A and 3B, as discussed in the

preceding section, offers a simple illustration of the major points regarding revisions. To add a new activity to the precedence diagram requires:

- (1) Entering a new node symbol in a convenient location on the diagram.
- (2) Giving the new node a descriptive label and a number label.
- (3) Entering the necessary sequence lines to and from the new node.

To make the same revision on the arrow diagram requires:

- (1) Erasing and moving the existing arrow No. 25-30 to a new location.
- (2) Erasing and replacing the descriptive label of that arrow at its new location.
- (3) Entering the new arrow between the end of the relocated arrow and a previously located event node.
- (4) Giving the new arrow a descriptive label and numbering the event node between it and the relocated arrow.
- (5) Entering and labeling a dummy arrow No. 28-30.
- (6) Taking the necessary steps to see that the numerical designation of former activity No. 25-30 is changed to No. 25-28 in any records being used within the organization or in data distributed to others.
- (7) Possibly adding other dummies and repeating steps (1), (2), (5), and (6) for other activities; this if one or more of the activities originating at event-node No. 65 are not dependent upon the activity that has been added.

Even for as simple a case as this it should be obvious that the precedence diagram is much more easily and quickly revised.

Numbering System

The single numbering system of precedence diagramming has several

advantages over the dual numbering requirement of the arrow diagramming system. The most obvious is that it is more convenient to label, refer to, and index an activity with a single number than with dual, hyphenated numbers.

Another important advantage was pointed out in the preceding section. In precedence diagramming each activity has a unique number that can be permanently assigned to it. As long as random numbering, or at least non-consecutive numbering, is permissible, diagram revisions do not require changing the numbers assigned to an activity. As already shown, diagram revisions of arrow networks may require changing the originally assigned numbers.

Precedence diagramming avoids another problem that arises in the numbering of arrow diagrams and that also requires adding dummy activities. This situation arises when two or more activities have identical preceding and following activities and it is desirable to show these activities separately rather than lumping them together as a single, combined activity. An example is shown by the small portion of a network diagram presented in Figure 7. Here in preparation for placing

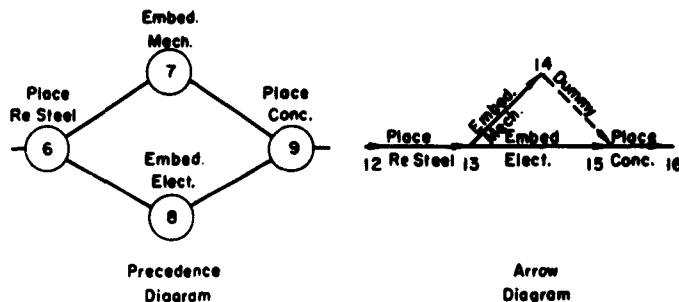


Figure 7. Illustration of Numbering Problem

concrete in a floor slab, reinforcing steel is laid, after which both embedded mechanical and embedded electrical items are installed. These latter two activities can proceed concurrently after steel installation and before concrete is placed. However, since two different subcontractors are involved, it is desirable to show these activities separately on the diagram and in any data output tabulations. A routine

1

application of precedence diagramming satisfies the desired requirements very simply. However, with arrow diagramming a dummy activity must be introduced. Otherwise, two activity arrows would have the same dual numbers. This would mean that when activities are designated by numbers, as is the case in communicating with an electronic computer, these two activities could not be distinguished from one another. The introduction of the dummy permits giving each activity a unique number and is the device required by arrow diagramming to solve this problem.

Another potential advantage of the single numbering system is that it lends itself more readily than dual numbering to use in conjunction with cost accounting systems. This, and the fact that a number once assigned will not have to be changed even if the job plan is changed, will permit a numbering scheme that can satisfy an overall system involving planning, scheduling, control and accounting functions. Single numbers drawn from certain reserved number series can also be used to indicate type of work, floor level in a multi-story building, work area, or other useful groupings.

The principal advantage claimed for the dual numbering of the arrow diagramming system is that it provides a means for determining which activities precede and follow a given activity without reference to the network diagram and by working with numbers alone. The preceding activities are those whose final number is the same as the first number of the given activity, and the following activities are those whose first number is the same as the final number of the given activity. Since numbers, rather than diagrams, are the language of the computer, the computer can use the dual numbers to establish precedence relationships. However, the same thing is accomplished in the case of the precedence diagram by listing the numbers (dual) of the sequence lines. In this respect the precedence diagramming system is a combination of single and dual numbering, combining the advantages of both. The activities have single, unique, permanent numbers with the resultant advantages. The sequence lines indicate sequencing in numerical terms and can be changed easily without adverse effect when diagrams are revised.

Adaptability to Computers

When the precedence diagramming system, as it is being called in this report, was presented in Tech. Report 9, no particular consideration was given to whether or not it was compatible with computer processing. That report was concerned only with manual methods based on reference to the network diagram. Critics of this system claimed that it would not lend itself to efficient computer processing. The basis for the claim seemed to have some substance. It was pointed out that the single numbering system for activities does not permit the computer to establish precedence relationships. Therefore, dual numbered sequence lines must be included as part of the input data. These sequence lines are, in reality, equivalent to dummy activities, but in quantity they far exceed the number of dummy activities required in a corresponding arrow diagram. Therefore they constitute excessive input data that must be handled, stored, and processed. The result would be less efficient computer processing.

Shortly after publication of Tech. Report 9, in early 1962, a computer program utilizing the Burroughs 220 computer was written at Stanford for the precedence diagramming system. It was successfully applied to actual projects in the building construction field. Later a greatly improved program was prepared for the IBM 7094 computer. This program provides both a printout and card output for various sorting and listing purposes. It also converts time data to calendar dates, allows random numbering of activities, provides resource schedules if desired, and offers extensive error checking. The number of input cards is approximately the same as that required by arrow diagramming programs. For this program one card must be furnished for every sequence line in the network. Activity information, including descriptions, durations, and any desired coding numbers, is included on these cards. Since the number of sequence lines in a precedence diagram is approximately equal to the combined number of actual activities and dummy activities in an arrow diagram, the net result is that about the same number of cards must be processed using either system.

The author believes that this computer program for the precedence diagramming system is at least equally efficient with existing ones for

arrow diagramming. However, the effort required to verify this for a project having a large network has not been made because it appears that the results would be of little more than academic interest in the case of computers such as the IBM 7094. To illustrate what is meant by this statement, consider two actual building construction projects processed by this program for which the following data was obtained:

	<u>Project A</u>	<u>Project B</u>
No. of Activities (no dummies)	425	716
Random Numbering	Some	Considerable
Input Time	0.08 min.	0.19 min.
Calculation Time	0.01 min.	0.05 min.
Output Time	0.16 min.	0.31 min.
Total Time	0.25 min.	0.55 min.

At a commercial service bureau rate of \$550 per hour for this computer, a project involving over seven hundred activities can be processed with about five dollars worth of computer time. Thus a slight advantage in one system over another represents a savings of only a matter of pennies as far as computer time is concerned. The other costs associated with getting information to and from the computer make these differences insignificant. More importantly, the other considerations such as relative simplicity in diagram construction are of so much greater importance in determining the more advantageous system than a few pennies cost in computer time, that this factor should have little weight in influencing a decision. It is worth repeating, however, that precedence diagramming permits efficient computer processing as well as easier non-computer procedures, a fact that has been challenged by some in the past.

Representation of Events

The event-labeled arrow diagram, characteristic of the PERT technique, is an event-oriented system. Each node on the diagram represents an event occurring at a specific point in time, and these events are labeled and are used for scheduling and control purposes. The activity-labeled arrow diagram, characteristic of the CPM technique, is an activity-oriented system, since it is the activities that are labeled.

However, the diagram itself is the same, and the events are shown by the nodes, even though they are not labeled.

The precedence diagram may be considered to be an activity-oriented system. The diagram is different than the diagram used in the other two systems. The principal differences are that the nodes usually represent activities and there is no representation of events. To those who are event-oriented this appears to be a serious shortcoming of the precedence diagram. Actually it need not be so.

This shortcoming is easily overcome because events can be represented on precedence diagrams by introducing them as activities of zero time duration. This approach is similar to the concept of the dummy activities used in arrow diagramming. However, there is an important difference. Dummy activities must be introduced of necessity in arrow diagrams in order to preserve the correct network logic. Event-nodes are introduced voluntarily on precedence diagrams only when it is desired to show a specific event. Figure 8 shows an example of an event, 'Completion of Work - Stage I Cofferdam Area.'

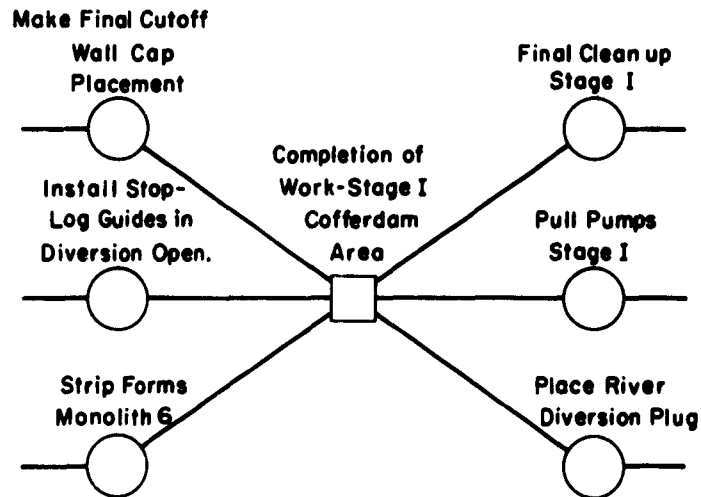


Figure 8. Precedence Diagram with Event Node

As indicated, it is sometimes very desirable to show certain key events associated with intermediate completion dates, installation inspections, seasonal shutdowns, and many other important project milestones. Sometimes this is a contract requirement. For example, a

recent NASA contract being administered by the U.S. Army Corps of Engineers at the Marshall Space Flight Center, Mississippi, required the construction contractor to include in his network diagram 99 specified milestone events but otherwise permitted activity labeling. A precedence diagram utilizing both activity and event nodes could be used effectively to satisfy such a requirement.

Event-nodes may also be used to improve the appearance of a precedence diagram. In the case illustrated in Figure 8, it is indicated that the three activities shown on the left side of the figure must be completed before the three activities shown on the right side may be started. If the event-node were omitted, there would be a total of nine criss-crossing sequence lines showing these relationships. Generally when one group of activities must all be completed before any of a following group may be started, the event marking this point in time is important enough to deserve special representation. But even if not, diagramming simplicity may suggest the use of an event-node. The use of event-nodes as interface points between subnetworks will be mentioned in a later chapter.

A point that is often overlooked by advocates of the event-labeled arrow diagram is that, despite the multitude of events shown and labeled on this type of diagram, an event may be missing that it is desirable to show and for which data should be developed. For example, in the portion of the event-labeled arrow diagram shown in Figure 9A, node No. 45 indicates a point in time when two other events have both occurred, the

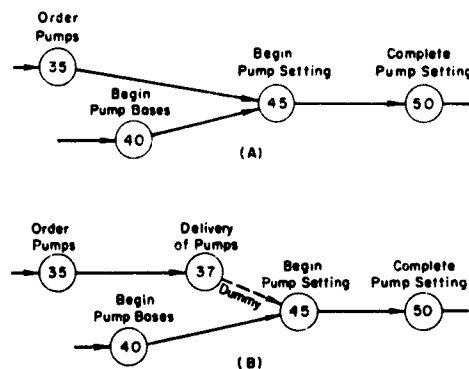


Figure 9. Additional Event Requirement for Event-Labeled Arrow Diagram

delivery of the pumps and the completion of the pump bases. These events are not shown separately. Even though the date of Event No. 45 might be established by the chain of activities through Event No. 40 and, therefore, would simply establish a maximum deadline for delivery of the pumps, an expeditor would prefer to see the promised delivery date of the pumps represented on the diagram. This is an important date and even though it may have free float originally, changed conditions may consume this free float. This would be brought to the attention of management more directly if this event were shown separately. This may be done, of course, by adding this event to the diagram as shown in Figure 9B and inserting a dummy between it and Event No. 45. This represents a third type of situation requiring the addition of dummy activities to arrow diagrams. To generalize the example that is presented here, whenever two or more activities terminate at a node on an arrow diagram, or two or more activities commence at such a node, that node represents the event signifying completion of all the activities and possible commencement of all the activities terminating and originating, respectively, at that node. It does not provide a separate representation of the completion or beginning of any single activity. In some cases, because of special significance, it may be desirable to show the completion or commencement of a single activity separately. This may be done by taking the special steps of adding this event to the diagram and adding a dummy activity. The point is, however, that the use of an arrow diagram, where every node represents an event, still does not assure that all important events will be shown. Events that are not shown and that are important may be added by taking special steps; this is equally true of the precedence diagram.

Along these same lines, diagrammers who use the event-labeled arrow diagrams often introduce numerous extra event nodes and accompanying dummies in order to show the desired labels on the diagram. This is the case even when there is a single chain of activities, so that the problem given in the previous illustration is not a factor. Although this is the simplest type of diagram, each node still represents both the completion of one activity and the commencement of

another. Usually a choice is made between one or the other of the possible descriptions when events are labeled. An example is shown in Figure 10A. However, it is not uncommon to see diagrams where the event node is repeated in order to show both labels, as is illustrated in Figure 10B. This seems to be an unnecessary proliferation of items in the network diagram that stems from a weakness of the event-labeled arrow diagram.

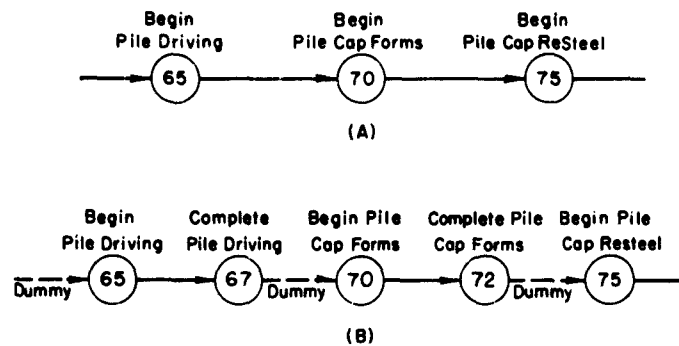


Figure 10. Double Labeling for Event-Labeled Arrow Diagram

In summary, it is agreed that it can be quite useful to represent certain events as well as activities on a network diagram. This can be accomplished simply under the precedence diagramming system.

Time-Scale Plotting

The original network diagram that is drawn as a job plan is developed is generally not plotted to a time scale. The usual practice is to obtain activity time estimates and perform the basic time calculations after a complete job plan has been represented in network diagram form. This original diagram is probably drawn on a rough worksheet with many revisions. After a satisfactory plan and schedule have finally been worked out, the network diagram is redrawn to give a neater and more orderly presentation to those who will use it. At this time the diagram may be plotted to a time scale. This provides a graphical portrayal of the scheduling relationships of activities as well as their sequencing relationships. It indicates clearly which activities are to be performed concurrently, and it shows the expected status of the job at any given date.

The disadvantage of the time-scale diagram, when produced by conventional drafting methods, is that the effort required to keep the diagram up to date is prohibitive. Whenever a single activity deviates from schedule beyond its free float range, other activities are affected and, often, extensive revisions are required to update the diagram.

In spite of this disadvantage, the time-scale diagram is still popular and useful. It can be read more easily by personnel that in the past have been accustomed to obtaining scheduling information from bar charts. Moreover, it offers a good basis for presentation even if there is no subsequent attempt to keep it up to date. The time relationships between activities will probably continue to be represented more logically by an out-of-date time-scaled diagram than by a diagram that is constructed without reference to scheduling data. It also is true that while it may be impractical to maintain a time-scale diagram for an entire project currently up to date, provision of a time-scale diagram for that portion of the entire network that is scheduled during a given, limited time span may furnish a practical solution to the updating problem. For example, such a diagram might be issued at periodic intervals (monthly, say) and might show all the activities to be performed in the next time-period or next two time-periods (30 or 60 days, say). With regard to updating scheduling data and time-scale diagrams, the recommendations of this report may be summarized as follows:

- (1) Scheduling data should be updated on a day-by-day basis for those activities in progress during the immediate time period ahead and for the project completion date. (Procedures are discussed in Chapter VII.)
- (2) Scheduling data for the remaining activities that commence later than the next time period in the future should be updated periodically, generally once each time period.
- (3) A network diagram of the entire project should be drawn to a time scale, but there should be no attempt to update it except for logic corrections involving

changes in sequencing, the addition of new activities, and the deletion of activities.

- (4) Periodically, generally once each time period, a revised time-scale diagram for that portion of the job to be commenced during the next one or two time periods ahead, should be drawn and issued if it is genuinely desired by those who would use it.

A time period for the purposes anticipated here would probably be one month, although it could be more or less depending on the nature of the job. The author does not feel that scheduling revisions to time scale diagrams by manual methods, more than one or two time periods in the future is worth the effort. The availability of economical means to replot and update network representations by automatic methods would, however, make the revision of time-scaled diagrams for the entire project practical.

Assuming that the construction of time-scaled diagrams is advantageous to some degree, the principal question here is whether one diagramming system is inherently superior to the other for this purpose. Critics of precedence diagramming point out that in this system an activity is represented by a node which, essentially being a point, does not lend itself to being plotted to a time scale. An arrow, on the other hand, is a linear type of symbol and can easily be plotted to a time scale. However, this is not a valid criticism because a time-scaled precedence diagram, simple to comprehend, can easily be constructed.

The date for the scheduling of an activity with which one is first concerned is the start date. Therefore, for a precedence diagram, it is logical to plot each activity node on the vertical date line of a time-scale chart which represents the earliest (or scheduled) start for that activity. This provides a satisfactory presentation of the job plan to a time scale. Figure 11 shows the simple pier job from Tech. Report 9 (Figure 2 of that report) plotted to a time scale using the activity-labeled arrow notation, and Figure 12 shows the same job plotted to a time scale using the precedence diagramming notation. For any given vertical date line an examination of either diagram will indicate which

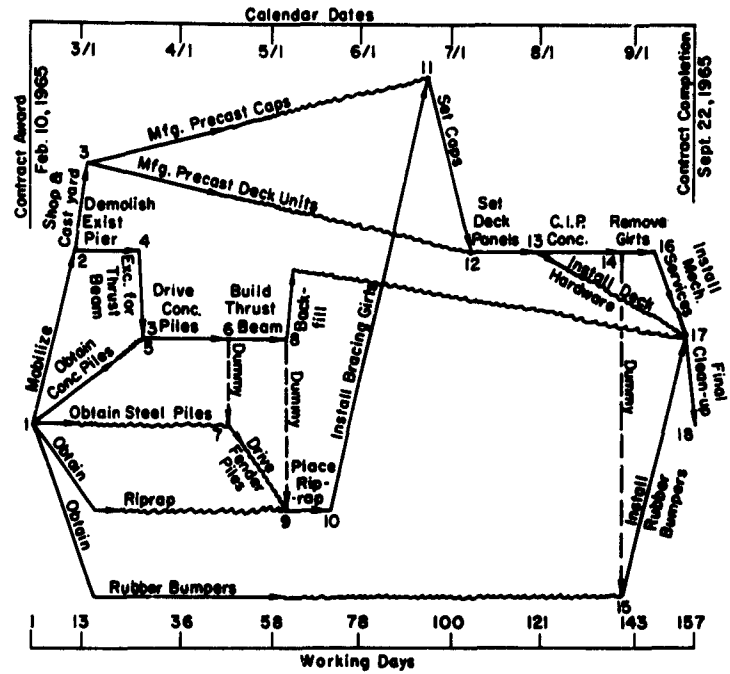


Figure 11. Time-Scaled Arrow Diagram

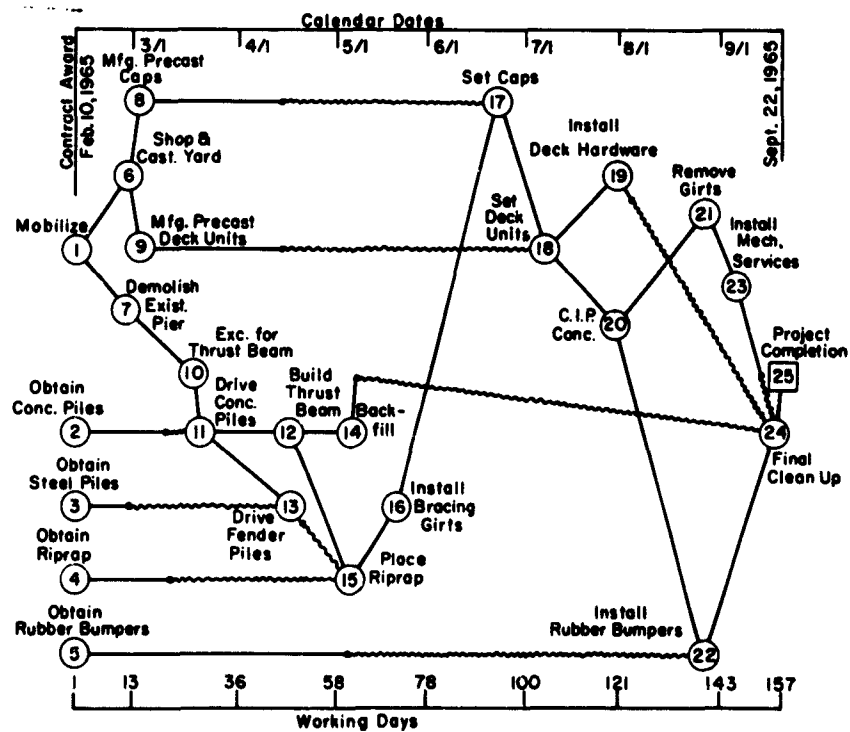


Figure 12. Time-Scaled Precedence Diagram

activities should have been commenced before that date, which activities are to be commenced next upon completion of those currently in progress, and the dates by which the current activities must be completed.

If free floats are desired, they may be indicated on the time-scaled arrow diagram, and lag times may be indicated on the time scaled precedence diagram. This optional feature is included in Figures 11 and 12. Then if a vertical date line passes through the free float portion of an arrow or through the lag portion of a sequence line (wiggling line), this indicates that the activity immediately to the left should have been completed. However, if the activity is still in progress there will generally be no detrimental effects as long as it is anticipated that the activities immediately to the right can still be commenced on the date shown.

The primary objective of time-scaled diagrams is to show the time relationships as well as the sequence relationships between the various activities. This objective can be accomplished with approximately the same effectiveness and ease by either arrow or precedence diagrams.

Current Practice

The strongest factor in favor of the use of arrow diagramming is its widespread current adoption. Early CPM and PERT work was based on arrow networks. The dissemination of knowledge concerning these techniques has been largely by management consulting firms and governmental agencies familiar with this early work. Many computer programs based on arrow diagramming have been written and are conveniently available. Owner agencies who have not been aware that any alternative exists or who are already deeply committed to their original system, have written specifications requiring the furnishing of an "arrow" diagram or even the use of a specific computer program.

It would have been easier to adopt the arrow diagramming system for the presentation of the non-computer procedures described in this report and in Tech. Report 9. However, precedence diagramming was chosen for reasons that seem more important to the objectives of these reports. Critical path techniques are still in their early stages of

development and implementation. It is one thing to convince the upper echelons of company management of their value. It is another thing to achieve their extensive application in a truly effective manner by the lower echelons of on-the-job project management. The former has been accomplished to a considerable extent both through presentations of the merits of the techniques that have led to voluntary adoption and through the brute-force means of specification requirements. But the gaining of widespread, effective application at the job level is a more difficult battle that has yet to be won. It is in this effort that simplicity will play a major role.

Although precedence diagramming has been compared with arrow diagramming on a number of points and found to be equally or more satisfactory, the two key factors given greatest weight in reaching a decision were simplicity versus current practice. The decision to retain the precedence diagramming system in this report was based primarily on the conclusion that working with a simpler system outweighs the benefits of adherence to a system that is already well-established by current practice.

Constructing the Rough Diagram

The network diagram should be drawn during original planning of the job. It is helpful in guiding the systematic development of this plan. It provides a means for keeping track of all sequencing decisions. It requires that the planners show all the activities whose completion is prerequisite to the beginning of a new activity, rather than solely the single activity which the planner believes will be the controlling prerequisite. By presenting the plan in a manner that may be easily and clearly conveyed to others, the network diagram permits those who help to develop the plan or who must review it, to judge if sequencing decisions are logical and if all prerequisite activities have been considered.

To begin this process of job planning and the development of a network model, the planners should start with a blank roll of paper to serve as their worksheet. Initial job activities are considered, are

represented by a node symbol near the left end of the worksheet, and are labeled. As each activity is entered on the worksheet, the planners ask themselves whether there are any earlier activities that must be performed before this activity and what activities may be performed upon completion of this activity. These preceding and following activities are entered on the worksheet, and the appropriate sequence lines are drawn between the node symbols. Then the same question is asked about each of the new activities. This procedure continues, with following activities generally being plotted to the right of preceding ones, until the final job activity has been entered on the diagram. In those cases where one activity can commence before a preceding one is entirely completed, a further breakdown is required. The final plan is carefully reviewed by those who developed it and by any others who can be helpful in evaluating it, and any desirable revisions are made.

As can well be imagined by visualizing the foregoing procedure, the original network diagram will probably provide an untidy and somewhat disorderly presentation of the job plan. Activities will have been entered as they came to mind rather than in systematic groupings; sequence lines may have been drawn free-hand, have been frequently crossed out and revised, and may run in circuitous paths or even from right to left as well as in the customary left to right direction; and the subdividing of overlapping activities and other necessary revisions will probably have required erasing or crossing out original node symbols and labels. The lack of tidiness of the original diagram is unimportant at this stage; in fact, there is probably a direct relationship between the benefit that has been derived from the use of a diagram as a planning aid and the number of changes. The important objectives of the rough diagram are to achieve a realistic and complete job plan and to record the decisions that have led to this plan.

Having fulfilled these important goals, the diagram will also provide the sequencing information necessary in making basic time calculations as well as computations for the advanced critical path techniques. It will also serve as a paper model for conveying the job plan to others. At some point it generally becomes advantageous to redraw the

diagram in order that it may serve these purposes more effectively. But the rough diagram may be satisfactory for some of these purposes. For example, the first scheduling computations may often be made from, or directly on, this diagram. Or if computer processing is to be used, the original computer input data may be entered on the coding sheets directly from it.

Before redrawing the diagram it is recommended that this first run of the basic time calculations be made in order to determine if an acceptable job plan has been produced. If these calculations are made manually on the diagram itself, only the "forward pass" involving determination of early start and early finish times is required. This will indicate the expected project duration. By tracing backwards from the final activity along sequence lines connecting equal early start and early finish times to the first project activities, the critical activities can be determined without calculating late finish and start times and total floats. If the project duration is satisfactory and if there are no obvious reasons for changing the sequencing of the critical activities, the diagram may be redrawn with reasonable assurance that the effort is not premature. However, if the resulting project duration is unsatisfactory, then major replanning of the job with corresponding changes in the diagram may be necessary. Or, having determined the critical activities, a more concentrated examination of these activities may indicate possibilities for overlapping them; this requires changes in the breakdown of activities and in the diagram. Only after an acceptable job plan has been achieved and the logic and resulting project duration have been reviewed and approved, should redrawing of the diagram be considered.

Figure 13 shows the rough diagram for the network from which Figure 12 was developed. Besides sequence numbers, which are discussed in the next section, estimated duration times were entered for each activity. Then early start and early finish times were calculated. Finally, the critical path was traced across the diagram starting at the final activity and working to the job beginning. In this example the node symbol was divided into four sectors to receive the four bits

of data that were to be used or calculated.

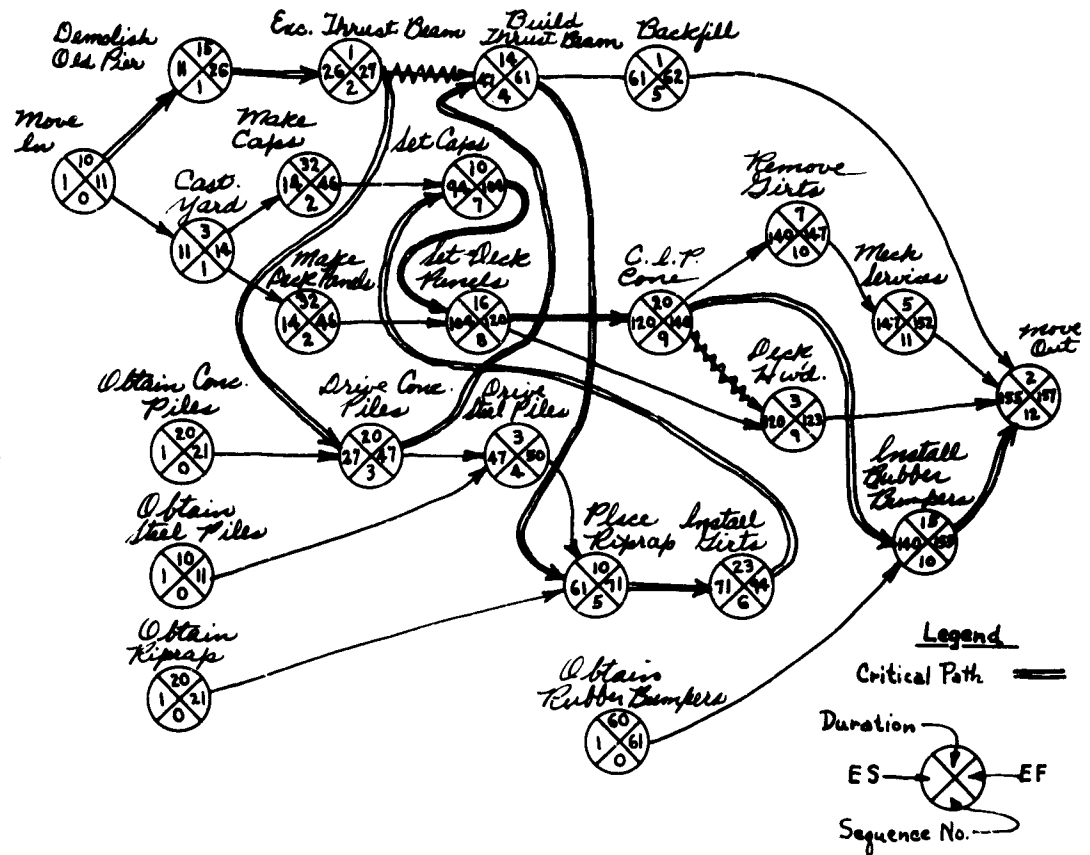


Figure 13. Initial Rough Network Diagram

Redrawing the Diagram

The principal purpose of redrawing the original network diagram is to produce a neater, better organized presentation of the job plan from which information may be obtained more easily by those who will use it. Where further calculations are to be made directly on the diagram (or on a print made from it), an orderly arrangement of the activities will permit such work to be done more rapidly. And, of course, where the diagram is to be furnished to others than the original planners, appearance may be a factor demanding redrawing of the rough worksheet version.

One important method of achieving a clearer presentation is to

group related activities on the diagram. For example, in the case of a multi-story building project, the activities involved in the construction of a single floor may be grouped together. The resulting pattern may then be repeated in neat and logical order for the other typical floors. It may also be logical to group the general contractor's structural operations on each floor separately from the subcontractors' items of finish work. Carrying this still a step further, certain closely related activities may be grouped in single horizontal chains. These might include a series of related concrete placements or the series of activities of one single subcontractor. There are an infinite number of diagramming arrangements for every job plan, and, therefore, there is an opportunity to display considerable skill in developing a logical presentation with a minimum degree of complexity.

One way to avoid a complex appearance is to keep sequence lines as short and direct as possible. However, the use of broken lines instead of straight ones is preferable when this permits the elimination of confusing intersections or avoids intersections at poor angles. As explained earlier in this chapter, the introduction of an event node can sometimes simplify a portion of a diagram.

A basic requirement for the logical presentation of the job plan which may be violated on the rough diagram, is that the flow of work should proceed from left to right across the diagram. Two formal methods for satisfying this requirement are the use of either a sequence-step diagram or a time-scale diagram.

Sequence-Step Diagrams

The sequence-step approach to redrawing a network diagram in a more orderly form was used in Tech. Report 9 (see pages 22-23 and Figures 1, 2, and 3 of that report). In this approach the nodes on the rough diagram are assigned temporary sequence numbers. These numbers indicate the total number of preceding nodes in the longest chain between the beginning of the project and the node being numbered. After this is accomplished, equally spaced vertical guide lines are laid out across the sheet on which the diagram is to be redrawn. Since the

number of guide lines is determined by the sequence number of the final activity, a suitable scale can be selected in advance to produce the desired size of diagram. Having numbered the guide lines from left to right, activity-nodes (and event-nodes) are plotted on the guide line with their corresponding sequence number. The nodes should be located vertically along the guide lines to achieve logical groupings and favorable positions with respect to other node symbols. The result of this procedure is that each activity is located to the right of all its preceding activities but as far to the left as is consistent with that limitation. Finally the activities are given permanent numbers by starting at the top of the first sequence guide line, running down that line of activities, moving to the top of the second guide line, running down that line, and proceeding in this fashion to the extreme right end of the diagram. Figures 13 and 14 (duplicating Figures 1 and 2 of Tech. Report 9) show, respectively, the assignment of sequence numbers on the rough diagram and the resulting redrawn sequence-step diagram.

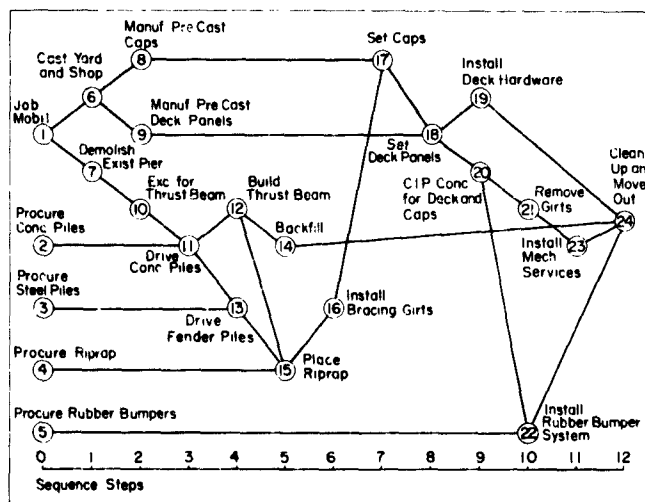


Figure 14. Sequence Step Network Diagram

The sequence-step diagram provides a good layout for performing calculations on the diagram. The systematic left-to-right, top-to-bottom pattern described above for final numbering of the diagram may also be used for these calculations and for the orderly scanning of values entered on the diagram. Such calculations and scanning are required by some of the procedures for updating, time-cost tradeoffs, and resource allocation. The sequence-step diagram also provides an effective means for arranging activities on a final diagram if it is not desirable to assign them permanent numbers or to perform initial scheduling calculations on the rough diagram. For example, for a large and complex network a planner might decide to perform the basic time calculations using a computer program that does not permit random numbering and, in this case, he might find it advisable to first redraw the rough network before assigning permanent node numbers. In this case the sequence-step diagram may serve as an intermediate network model during the stage of replanning the job. After all scheduling calculations and logic revisions necessary to achieve an acceptable job plan have been completed, a time-scale diagram might be drawn. In other cases, the sequence-step diagram may serve as the final diagram of the master plan for the entire job. If desired, it can be supplemented by a periodically issued time-scale diagram showing the portion of the over-all network, possibly with more detail, that covers the work to be performed in the next time period.

The principal defect of the sequence-step diagram is that even though activities and events are shown in an orderly and proper sequential relationship, running from left to right across the diagram, they are not necessarily shown in their proper time relationships. For example, the activities that are plotted on a given vertical sequence line are not necessarily concurrent but, rather, may be performed in entirely different time periods that may not even overlap. When new activities are added to a diagram, the sequence-step diagram may be slightly more difficult to revise than other types of diagrams. Generally, however, new activities can be squeezed between existing sequence lines or, if extensive revisions are made, the original

diagram may be cut vertically and a strip showing the new activities may be inserted, as an alternative to redrawing the entire diagram. The fact that the sequence-step diagram does not need revision when time data and corresponding schedule changes occur more than compensates for the much less frequently occurring need to make changes for the addition of new activities.

Time-Scale Diagrams

It is usually practical to make at least the first set of scheduling calculations directly on, or from, the original rough network diagram. Often the results of these basic time calculations lead to revisions in the planning, and one or more additional sets of scheduling calculations may also be made on the rough diagram. If the diagram is to be redrawn after these initial calculations, it can be plotted directly on a time scale. In a sense the time-scale diagram is merely an expanded case of the sequence-step diagram where date lines replace sequence lines and early start times replace sequence numbers.

The time-scale diagram has the disadvantage, noted earlier in this chapter, that it generally requires a prohibitive amount of effort to keep it up to date. A practical compromise is to plot the network, after any initial replanning has been completed, to a time scale but not to attempt to update it for scheduling changes as the job progresses. At the beginning of the job, then, the activities will be presented in both the correct sequential and time relationships. If scheduling changes occur as the job progresses but the diagram is not updated, the sequence relationships will still be correctly shown and the time relationships, though not entirely correct, will probably still be represented more accurately than if the sequence-step had been used. It would, of course, be advisable to remove the time-scale labeling when it ceases to be valid.

As a worksheet for making calculations or for assigning permanent numbers to activities, the time-scale diagram may be used in the same manner as described for the sequence-step diagram. It has the slight disadvantage of grouping fewer activities on the same vertical line and,

therefore, requiring more horizontal shifting as one works across the diagram. This disadvantage is counterbalanced by the more accurate representation of time relationships between activities. The ability to scan down a vertical line through the diagram and tell which activities are being performed concurrently often permits problem areas to be recognized. For example, even where more formal resource allocations procedures are not employed, the time-scale diagram may indicate periods where there are excessive requirements for certain key resources. Then rescheduling on an informal, judgment basis may effectively remove these more obvious resource peaks.

Figure 12, referred to previously, shows a time-scale diagram for the simple pier job shown plotted on a sequence-step basis in Figure 14. A comparison of these two diagrams indicates relatively little difference in the time relationships as portrayed by the two methods. In a larger network it is more probable that some of the time relationships between activities will receive greater distortion by the sequence-step method. However, often this distortion is subject to a simple correction. An activity that is plotted on a sequence line too far to the left to be in correct relationship to other activities from a time standpoint may arbitrarily be shifted to a sequence line further to the right if this can be done without affecting the position of other nodes. For example, Activity No. 14 on Figure 14 could be shifted from one to six sequence lines to the right without affecting the rest of the diagram. In this case it is not desirable to shift it, but in cases where such a shift would place the activity-node in a more accurate position from a time-relation viewpoint it may be made.

To summarize this discussion of the choice of network diagramming methods, it should again be emphasized that the construction and use of the network diagram is the most important application of all the critical path techniques. To encourage this application on a broader base of job-level management, the precedence diagram was chosen because of its greater simplicity. This simplicity should not be lost by imposing tight, formal rules for diagram plotting. When a rough diagram is to be redrawn for presentation purposes or when a diagram is to be redrawn following

major updating revisions, it is probable that one man in an organization who clearly understands these more formal approaches can oversee the task of preparing the most appropriate type of network diagram. But in the construction of the original rough diagram, in job-level revisions to an existing diagram, and in sketching detailed subnetworks for analysis of portions of a larger job, it is not wise to place great emphasis on formal rules for diagramming arrangement. As long as the logic of the diagram is correct and it is presented in a manner that can be understood clearly, the primary objectives of diagramming will be accomplished.

CHAPTER III

BASIC NETWORK ELEMENTS AND MECHANICS

General

Having chosen the precedence diagramming system and having developed a network diagram which is a realistic paper model of the plan of action for performing a given project, time data is next introduced. Using estimated time values for each activity's duration and the sequencing relations shown by the network diagram, the project duration and the corresponding detailed scheduling for every activity can be calculated. Additional valuable information regarding the criticality or amount of scheduling leeway of each activity may also be computed. These calculations are simple and routine, and have been discussed in Tech. Report 9. They include the determination of early start, early finish, late start, late finish, total float, and free float. The term interfering float, to represent the time difference between the total float and the free float of an activity, was also used in Tech. Report 9 and is relatively common elsewhere. In the previous report the calculations were performed and entered on a separate tabulation sheet. It is worth noting here that an alternative is to perform all work directly on the network diagram. Except for this observation, there is no need to discuss these computational procedures again.

Although the basic time calculations furnish very valuable information for scheduling and control purposes, the potential of the critical path techniques is realized only to a very limited extent if their application is terminated at this point. These techniques have been properly described as "dynamic" ones, but this is only so when updating procedures are included. Because time estimates and scheduled start dates are inevitably changed during the course of job performance, updating is required to revise the former schedules. Sometimes this updating must include sequence changes and the addition or

deletion of activities as well. Updating is also the basis for the application of advanced critical path techniques such as time-cost trade-offs, resource allocation, and multi-project scheduling. While these procedures attempt to improve the pattern and magnitude of the expenditure of resources, whether they be men, material and equipment, or time and money, they are all based on making planned changes in activity durations, scheduled dates, or sequential relationships. They involve systematic replanning requiring frequent updating and the analysis of the results produced by a proposed change. So, except for a rudimentary application of the critical path techniques, updating plays a basic role.

The principal reason that electronic computers are generally considered essential in the application of the critical path techniques to all but the simplest networks, and for any application of the advanced procedures, is that frequent updating is needed and the computer can perform this updating efficiently. If the range of practical application of non-computer methods is to be extended or if these methods are to be used to effectively supplement computer processing in a mixed application, the problem of efficient updating by non-computer means is a major one. Updating by computer is accomplished by a complete recomputation of all output data using the revised input data. If a complete recomputation is essential, non-computer methods will continue to be a poor substitute for computer processing in the great majority of cases. The success of non-computer updating appears to rest, in the author's opinion, in the following three approaches:

- (1) Increased selectivity in data requirements through recognition that only certain data must be currently updated and that other data need only be updated occasionally or not at all.
- (2) The development of efficient methods to revise only the data affected by a given change rather than using the more indirect approach of a complete recomputation of all data.
- (3) The development of methods that permit revising data only in a portion of the overall network without

suffering important losses from the lack of revised data for the entire network.

Chapter IV will discuss the first of these three approaches; Chapters V and VI the second, and Chapter VII the third. All methods are based on basic network mechanics. The purpose of this chapter is, then, to examine network elements and mechanics in more detail than was presented in Tech. Report 9 in order that a suitable foundation for the following four chapters will be established. Of the two network elements, nodes and lines, the former will be considered first.

Network Nodes

The nodes of the precedence diagram are the elements that receive major emphasis and descriptive labeling. Nodes are generally represented by circles, probably because a circular symbol is convenient to draw. However, other symbols may be used for various reasons. For example, if nodes represent events as well as activities, it may be desirable to distinguish an event-node by representing it by a square instead of a circle. Or, if calculations are being performed directly on the network diagram and there are six items of data to be associated with each activity, a hexagonal node with six interior triangles in which to record data might be useful. Except on rough diagrams, nodes are usually drawn with templates. Other possibilities include the use of a rubber stamp or of punch-out symbols having adhesive backings.

A majority if not all of the nodes of a precedence diagram will represent activities. The designation "activity" is used in a broad sense to refer to any time-consuming requirement in the execution of a project. In most cases these activities involve the physical performance of job-site tasks. However, they may involve off-site items as well, such as the manufacture of materials by suppliers, the approval of drawings by an architect, or the procurement of a necessary permit. They may also involve time-consuming requirements which are non-physical, such as lead times and restraint periods. For example, if certain conditions make it advantageous to postpone a particular operation until 60 days after the beginning of the project, an artificial activity representing the passage of 60 days may be inserted between

the beginning of the project and the activity to be restrained. An event differs from an activity in that it does not fulfil the time-consuming requirement but, rather, represents an instant in time. However, it is useful to consider the event as a special case of an activity where the time requirement has been reduced to zero and to treat the event-node in the same manner as an activity-node insofar as network mechanics are concerned. A diagram consisting of only event-nodes similar to the event-labeled arrow diagram, has no meaning with precedence diagramming, since there would be no time-consuming elements in the network.

Every node in the precedence diagram is labeled. Usually each node has two identifications; one is a descriptive title which briefly defines the activity or event and the other is a symbolic title, such as a number or letter, which is more convenient for computational purposes. If calculations are made directly on the network diagram, only the descriptive title is needed, since the symbolic title is not used for computational purposes in this case. However, if the calculations are made on separate forms or if a record of results is maintained as a separate tabulation, the symbolic title serves as a convenient abbreviation. If manual methods are being used, the symbolic title may consist of letters, or a combination of numbers and letters, as well as of numbers alone. Sometimes this is helpful, as, for example, when a single operation must be broken down into multiple activities because other activities may commence at intermediate stages of its completion. Here it may be advantageous to assign a single number to the overall operation and letter suffixes to the component parts, such as Activities 12A, 12B and 12C. When computations are to be performed by a computer, symbolic labeling is essential and generally must be entirely numerical.

Theoretically, the activities represented by the nodes of a precedence diagram (or by the arrows of an arrow diagram) should cover all work that is required for the performance of the project. Actually a complete coverage is rarely practical. To attempt to show every minor, time-consuming task would result in an unwieldy diagram requiring

that too much input data be furnished and producing too much output data to be absorbed effectively. For example, it is very important for the diagram to include activities involving the procurement of materials having a substantial delivery time, such as piles, structural steel, or major items of installed equipment. However, it is not realistic to show the procurement of the many off-the-shelf items of hardware and other expendable supplies which may be obtained on fairly short notice. Nonetheless their procurement is also time-consuming and requires manpower for unloading and handling on the job. Even in the case of major items of materials, such as reinforcing steel, the furnishing of shop drawings, fabrication, and delivery of the first major shipment are usually included on the diagram, but the corresponding steps for continuing shipments are rarely shown. Nor is the omission of activities from the diagram confined only to cases involving procurement of supplies. In most cases there are many man-hours of work on minor tasks that are never shown. Therefore, in viewing a network diagram for a project, the fact should be kept in mind that there are missing activity-nodes because it is impractical to show a 100 per cent coverage. It follows that the activities shown seldom account for 100 per cent of the resources required to perform the job.

It may be wise to give more detail and achieve closer to a 100 per cent coverage of the work to be performed for a limited portion of the project. For example, if a portion of the overall network representing work to be performed in the immediate future can be isolated without destroying any vital network relationships, it would be both possible and practical to add more detail and obtain a more complete representation of this segment. This approach will be discussed in Chapter VII.

Another conflict between theory and practicality involves the scope of an activity represented by any single node. From a theoretical viewpoint this scope has precise boundaries. It includes only that which cannot be performed until the preceding activities are completed and which must be performed before the following activities can commence. In a practical sense it is seldom possible to define activities with

such distinct limitations. Most activities, no matter how finely they are broken down, still can overlap to some degree. This overlapping is contrary to the fundamental assumption underlying networking and is an obstacle to producing a truly realistic paper model of the project. Since the validity of the results obtained from all the critical path techniques is dependent on the degree of reality of the network diagram, this is a basic problem. Generally a reasonably realistic network model that provides useful results can be constructed with an ordinary amount of effort, so that this problem is not serious and is generally ignored. However, by remaining aware that the boundaries of activities often are not precise it is sometimes possible to make more intelligent applications of the critical path techniques.

As an example, consider the case of three activities from a network diagram that are shown in series and are labeled, respectively, A, B, and C. The diagram states that Activity A must be completed before Activity B can be commenced and that Activity C cannot start until Activity B is completed. A duration time for Activity B that is furnished by an estimator is presumably consistent with this definition of the scope of Activity B. It is probable, however, that there is some preparatory work in connection with Activity B that could, if necessary, be performed prior to the completion of Activity A. For example, certain materials might be moved to the work site and laid out for assembly; some work might be pre-assembled; jigs, scaffolds and material handling devices might be prepared; and so on. Furthermore, it is probable that there is some finish-up work for Activity B that could, if necessary, be performed even after Activity C has commenced. The obvious comment at this point is that Activity B should be broken down further into at least three activities. Without doubt, if this preparatory and finish-up work can be clearly defined, has rigid sequencing restrictions of its own, and is significant, the further breakdown is a solution. In many cases, however, an attempt to subdivide the activity would require more effort than is justified. The greater complexity of the resulting diagram and the additional input and output data would be more detrimental than helpful. Nonetheless, it is important to recognize this imperfection

in the definition of Activity B because:

- (1) If the activity proves to be critical, the effort required to break it down further may be justified. Even if this is not done, the activity duration may be re-estimated and revised downward to include the performance of just that portion of the activity that must fall between the completion of A and the beginning of C. This may permit a decrease in project duration without an increase in cost.
- (2) If the activity proves to be critical, some work that would normally not be considered to be preparatory or finish-up work might be converted to it by incurring increased costs. For example, by performing the work under less favorable conditions or by taking extra preparatory steps, the critical portion of the activity might be expedited. This is a form of time-cost tradeoff that should not be overlooked.
- (3) Whether an activity is critical or non-critical, the possibility of performing a part of it outside the time boundaries established by the basic time calculations offers an additional opportunity to employ resources that might otherwise be idle or laid off and, at the same time may reduce the need for resources at peak points. This is an effective means of smoothing resource requirements.

In summary, decisions on the degree of work breakdown and on the scope of the activity represented by each network node require good judgment and some amount of compromise. The intelligent use of the network diagram requires an understanding of the approximations that go into the practical definition of activities.

Network Lines

The other basic elements in the network diagram besides the nodes, are the network lines. Each of these connects a single preceding node

to a single following node. They indicate sequencing restrictions and are called, in this report, "sequence lines." These sequencing restrictions may result from physical or logical necessity, or they may be arbitrarily introduced. An example of a physical restriction is one that states that concrete cannot be placed until after the formwork for the placement has been erected. An example of an arbitrary, or decision, restriction is one that states that formwork for placement B will be erected after these forms are stripped from placement A, since a decision has been made that only one set of forms will be built.

Both types of sequencing restrictions appear in most network diagrams, because there are always physical restrictions that must be shown and generally decision restrictions that are so obviously necessary that the planner includes them even on the original diagram. However, good arguments can be presented for showing only physical restrictions on the original diagram and avoiding all decision restrictions at that stage of job planning. Then, by the use of a methodical resource allocation procedure, activities can be rescheduled and decision restrictions introduced as necessary to transform the plan into one that is realistic.

Although the practicality of constructing a diagram where 100% of the sequencing restrictions represent conditions of absolute physical necessity can be questioned, this general approach does have merit. It offers opportunity to replace arbitrary decisions with ones based on known resource availabilities and known effects on project duration. Many cases may be cited where decision restrictions that the planner considered to be obvious were actually unnecessary or incorrect and detrimental to most economical performance. The procedure of initially avoiding decision restrictions and introducing them later based on data analysis will become more useful as resource allocation methods are improved and simplified.

In the absence of the ideal approach where decision restrictions are initially avoided, some of these sequencing restrictions will be included in the original job plan. After the critical paths have been determined, it is advisable to examine each sequencing restriction in

the chains that make up these paths. When a restriction results from an arbitrary decision, the basis for this decision should be reevaluated. There is a possibility that an economical means of reducing project duration can be achieved by adding resources and performing certain activities in parallel rather than in series. As an aid in suggesting such possibilities for reevaluation those network lines representing decision restrictions might be differentiated from those for physical restrictions. For example, as a diagram is developed, each decision restriction might be shown as a dashed line, with a solid line reserved for physical restrictions.

When a network diagram is redrawn, it is customary for later activities to be shown to the right of earlier activities. In this case it is understood that the flow of work is from left to right; thus arrowheads are not required on sequence lines. However, on rough diagrams or on redrawn diagrams where this left-to-right convention has been violated, sequence lines should be shown as arrows indicating the following activity at the head of the arrow. It is important that all who use the diagram be aware that it is a precedence diagram and not an arrow diagram, even though arrows are used. The event-labeled arrow diagram and the precedence diagram are very similar in overall appearance even though they are quite different structurally.

Lag

For every sequence line in the precedence diagram there is an associated value which will be referred to as the "lag" of the sequence line. This term was originated and introduced in Tech. Report 9, and it is used with the same meaning here. Lag values are not required for the basic time calculations commonly performed in critical path applications. However, they will be the basis for the mechanics to be presented in this report for updating basic time calculations. These mechanics, in turn, are important in the non-computer procedures for the advanced critical path techniques where updating of data plays an essential role.

The lag value of a sequence line may be defined as the difference,

stated in number of time units, between the start time of the activity or event represented by the following node and the finish time of the activity or event represented by the preceding node. The time values used should be consistent, and there are several possibilities. These are early start and finish times; late start and finish times; or scheduled start and finish times, which may differ from either of the preceding values. Ordinarily, scheduled time values, which are usually but not always the same as the early time values are used to determine lag values. Where the simple term "lag" is used, it will be determined by the currently scheduled time values. Lag values based on early time values will be designated as "early-lag." Or if it is necessary to use lag values based on late start and finish times, the term "late-lag" will be used.

A lag value must either be positive or zero. If a following activity commences (or a following event occurs) later than the completion of a preceding activity (or occurrence of a preceding event), the lag of the sequence line connecting the corresponding nodes will be positive. If the following activity commences immediately upon completion of the preceding activity, the lag of the sequence line connecting them will be zero. A negative lag value indicates that a following activity could start before a preceding one was completed; this violates one of the fundamental assumptions of network diagramming. Since such apparent errors unavoidably occur to a limited extent in practice, they should be acknowledged and utilized as suggested earlier in this chapter. At the same time it must be recognized that the procedures cannot recognize these imperfections in the routine mechanics of the system.

There is a close relationship between lag values and free float values. If an activity is followed directly by only one other activity, then its free float is equal, by definition, to the early-lag of the sequence line between it and the following activity. When an activity is followed by two or more activities then the least of the lag values of all the sequence lines between this activity and the following ones is equal to its free float. In Chapter II it was pointed out that it is possible to regard each sequence line as a dummy activity on an

arrow diagram. In this case the term early-lag would become the same as the term free float for each of these sequence dummies. Since lag values will be calculated and updated in the procedures developed in this report, it will prove convenient to use these values to obtain and update free float data rather than to consider free float calculations as a separate procedure.

Date Convention

Before proceeding with discussions that involve time data and scheduling dates, rules should be adopted for interpreting the meaning of a given date. By one convention an early start on Day 10 may be interpreted to mean the beginning of the tenth day; again by another it may mean the end of the tenth day which is equivalent to the beginning of the eleventh day. The convention in Tech. Report 9 was to interpret each date as being at the end of the corresponding day. In this system the first activity commenced on Day 0, which was interpreted to mean the end of Day Zero or the start of the first day. If the first activity required six days for performance, then it was finished on Day 6 ($0 + 6 = 6$) which meant, quite logically, the end of the sixth day. A following activity could then start on Day 6, which really means the beginning of the seventh day. This convention provides finish dates that require no translation, but the start dates require the addition of one day to obtain the value used in normal communication. This convention has the advantage that the finish date of the final activity gives directly the day on which final completion of the project is anticipated.

In much of the work to be covered by this report, procedures are oriented to start dates rather than finish dates. Thus it seemed appropriate to adopt the opposite convention, which provides start dates that require no manipulation. This convention interprets each date as the beginning of the corresponding day or, in the case of finish dates, as the end of the preceding day. Here the first activity will begin on Day 1. If its performance requires six days, it will be finished on Day 7 ($1 + 6 = 7$). This means the beginning of

the seventh day and must be interpreted to indicate the end of the sixth day. A following activity will then start on Day 7 which means, quite logically, the beginning of that day. It will be necessary to subtract one day from the finish date of the final activity in order to state the end of the day on which project completion is expected. If the finish time of the final activity is calculated to be Day 85, then project completion is expected at the end of the 84th day.

Neither of these approaches is perfect since some translation is required in either case. It is possible to add an extra step to the approach adopted in this report that will eliminate the need for any translation; this is to reduce each finish date automatically by one day. If an activity starts on Day 1 and requires six days for performance, it will finish on Day 6 ($1 + 6 - 1 = 6$). In stating the start time of the following activity, the one day must be added back to give Day 7. If this is done all start dates will represent the beginning of the corresponding day and all finish dates will represent the end of the corresponding day. It is simple to make such an adjustment by computer programming, and it would be possible, of course, in non-computer procedures as well. However the extra effort doesn't seem justified. Therefore in this report all dates will be interpreted as the beginning of the corresponding day for start times and as the beginning of the following day for finish times.

Transmission of Scheduling Changes

Whenever the scheduled start or finish date of one activity is changed, the scheduling of other activities may be affected. Such changes occur if the field performance of an activity requires more or less time than was estimated or when a time estimate is intentionally revised as a result of better data or a different plan for performance. These changes occur also, even when activity durations remain unchanged, if the scheduled start or finish time of an activity is set for a date different than the earliest possible date or is rescheduled from the date previously established. The resulting effects can be determined by making the changes in the data and recalculating the entire

schedule for the overall project. This is the simplest procedure in the case of computer processing. However, an analysis and understanding of the manner in which changes are transmitted through the network will permit a more direct approach that is more workable for non-computer processing.

The effects of changes are transmitted through a precedence network by sequence lines having zero lag values. These effects will not be transmitted by sequence lines having positive lags. However, the changes may cause certain positive lags to be reduced and eventually to become zero, at which time the effects of the change will be transmitted by the corresponding sequence lines also. The transmission of changes by zero-lag sequence lines is easy to understand. As already explained a zero-lag sequence line represents the case where the start time of the following activity is the same as the finish time of the preceding activity, or, stated differently, the following activity commences as soon as the preceding activity is completed. If a change delays the completion of the preceding activity, then the following activity cannot start until this later date. Hence the change is transmitted to the following activity. If the change results in completion of the preceding activity at an earlier date, then the following activity can start on this earlier date. The one exception to this latter statement occurs when the following activity is at the terminal end of another zero-lag sequence line. This case will be considered later in this chapter.

As an illustrative example, consider the simple network diagram shown in Figure 15. In this figure, data giving the activity number, duration, early start time, and early finish time is shown inside each activity-node symbol. Positive lag values are shown adjacent to the sequence lines for which they are computed; double lines indicate sequence lines having zero lag. Now suppose that the performance time for Activity 2 is increased from 10 days to 11 days. Its finish date will be changed to Day 15 (the end of the fourteenth day). This change will be transmitted by zero-lag sequence lines to Activities 4, 5, 7 and 8. Both the start and finish dates of each of those activities will

move forward one day as shown by Figure 16. Note that the change was not transmitted to Activity 6 but that the lag value of the sequence line to this activity was decreased by one day. Note also that the transmission of the change terminated at Activity 4 although it did reach the following activity via a different path.

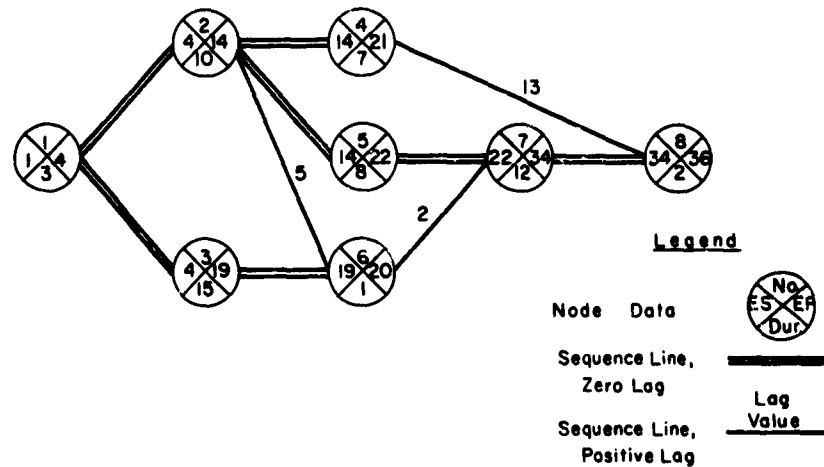


Figure 15. Illustrative Example - Original Job Plan

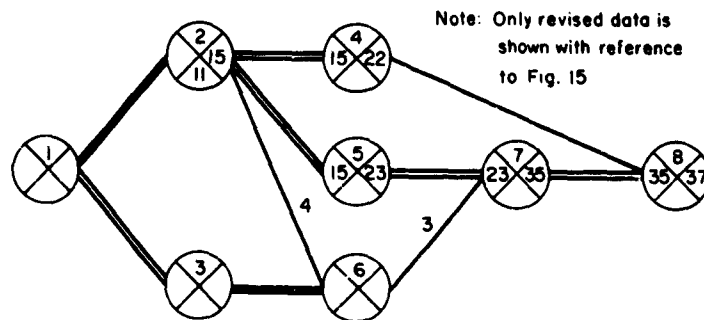


Figure 16. Effect of Duration Increase of Activity No. 2

Determination of Critical Path

The conventional manner for determining critical activities is to compute total floats and note which activities have a total float of zero. This requires calculation of early start and finish times (the forward pass through the network), of late start and finish times (the

backward pass), and subtraction of early finish times from corresponding late finish times to obtain total floats. When procedures are used in which lag values are calculated and zero-lag sequence lines are shown clearly on the network diagram, it is often convenient to determine critical activities directly from these sequence lines instead of computing late start and finish times and total floats.

If the duration of a critical activity is increased, the overall project duration will be increased by an equal amount. This characteristic provides a basis for determining critical activities from zero-lag sequence lines. The project duration is determined by the finish time of the final activity. Since that finish time is increased if the duration of any activity connected to the final activity through chains of zero-lag sequence lines is increased, all such activities must be critical and no others can be critical. Therefore the critical activities may be determined by starting at the final project activity and tracing back along all zero-lag sequence lines towards the beginning of the project. The paths traced in this manner will be critical paths and all activities on these paths will be critical activities. For example, by commencing at Activity 8 on Figure 15, this procedure will indicate that the critical activities in this network are Activities 8, 7, 5, 2, and 1.

There must be at least one critical path from the beginning of a project network to the end of the network, since some chain of activities connecting the beginning and end of the project determines project duration. This is true when all activities commence at their earliest start times and continues to hold true, even though some non-critical activities are rescheduled to start at later dates, as long as all critical activities start at their earliest start times. However, at times there is good reason to reschedule a critical activity to start at a later time than its earliest possible start date despite the fact that project duration is increased by this decision. In this case, the critical activity preceding the rescheduled activity is no longer critical, and there is no critical path across the entire network. Of course a critical path across the entire network can be developed if

an artificial restraint activity is introduced between the beginning of the project and the rescheduled critical activity. It must be assigned a duration that will produce the desired start time of the rescheduled activity. This procedure is generally unnecessary and becomes undesirable when there is extensive rescheduling, as there may be in studies to improve resource allocations. However, this procedure is useful where automatic updating methods are based on a recomputation of the basic time calculations for the entire project. Here, subsequent updating would convert the rescheduled activity back to its former start date in the absence of the artificial restraint. Some computer programs avoid this problem by providing for the input of scheduled dates.

As an illustration of the points discussed in the preceding paragraph, suppose that Activity 5 of the network shown in Figure 15 were rescheduled to start on the 16th day, a Monday, because it was undesirable to commence it near the end of a week. The resulting data revisions are shown in Figure 17. The only critical activities, as determined by backtracking along zero-lag sequence lines, are Numbers 8, 7, and 5. If an artificial restraint activity, Activity 9, were introduced, as shown by the dashed symbol and sequence line, it would be critical also and there would be a critical path across the entire network. Otherwise there are no critical activities in the first portion of the job and there is no critical path across the entire network.

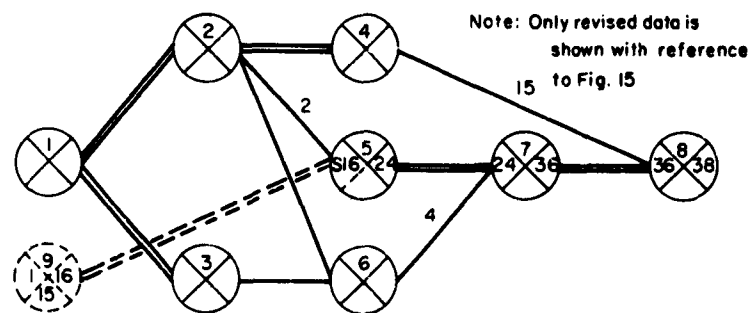


Figure 17. Effect of Rescheduling Activity No. 5

It is recommended that instead of using an artificial restraint activity when an activity is to be rescheduled later than its earliest start time, the newly scheduled start time be stated and be preceded by the letter "S." This will be a reminder that this date has been specifically scheduled. Then if a later routine updating affects this date, there will be a warning that the results should be investigated since further rescheduling may prove necessary.

It is desirable to maintain a record showing the reasons for all scheduled dates. Otherwise it may be difficult to recall why an activity was rescheduled for a certain date. Moreover, these reasons should be frequently reviewed because it is possible that the conditions upon which they are based will change or cease to exist.

Determining the critical path by backtracking along zero-lag sequence lines is a useful procedure when the calculation of late start and finish times and total floats is not required for other purposes. This may be the case, for example, when the original rough diagram has just been completed. Early start and finish times must be calculated to determine project duration. If this duration is not satisfactory, the critical activities must be replanned. The use of lag values will permit this replanning and the accompanying updating without a complete recomputation of all data (for further detail, see Chapter V). If lag values are computed for use in updating they will be available to determine critical activities. Until a satisfactory plan is finally achieved there is little need for data on late start and finish times and total floats. Moreover, Chapter IV will suggest that even in those cases where late start and finish times and total floats are required, they need not necessarily be kept continuously updated. Therefore an alternative method of determining critical activities based on the use of data that should be kept continuously updated has additional merit.

Lag Changes

It has already been noted that when the start or finish time of one activity in a network is changed, additional changes may be produced

not only in the start and finish times of other activities but in the lag values of certain sequence lines as well. Since the lag of a sequence line is determined by the difference between the start time of the following activity and the finish time of the preceding activity, it follows that lag must change if one of these two values change and the other does not. If neither of the two values change or if both change equally, then the lag remains unchanged.

As an illustrative example, consider again the revised network diagram shown in Figure 16. The change in the duration of Activity 2 has caused the lag of Sequence Line 2-6 to be decreased, since the start time of its following activity remains unchanged but the finish time of its preceding activity was moved to a later date. The same change caused the lag of Sequence Line 6-7 to increase since the start time of its following activity increased to a later date but the finish time of its preceding activity remained unchanged. Sequence Line 4-8 is one of five whose lag is unchanged due to the fact that both the start time of the following activity and the finish time of the preceding activity increased to later dates. Sequence Line 3-6 is one of three whose lag is unchanged because neither the start time of the following activity nor the finish time of the preceding activity was changed.

The fact that lag values may either increase or decrease as a result of activity duration or scheduling changes means that zero lags may become positive and that positive lags may become zero. This is important because zero-lag sequence lines serve as the vehicles for transmitting changes through the network. If new zero-lag lines are added or existing ones are changed to positive-lag lines, a different pattern for the transmission of changes results.

Network Interaction Limit (NIL)

Suppose that a certain change of one time unit in either activity duration or in the scheduled start or finish time of an activity results in a decrease of the lag value of one or more of the network sequence lines. Then, if successive changes are made in the same direction in

increments of one time unit each, the decreasing lag values will approach and may eventually reach zero. For a specific change, the number of time units that first causes one of the positive lags to be reduced to zero is defined as the Network Interaction Limit for that change. This term was originated and introduced in Tech. Report 9. Since it will be used frequently in this report, the abbreviation "NIL" will be employed. The NIL is important because it establishes the magnitude of a change that will first produce a significant change in the network relationships for the entire project. With the introduction of a new zero-lag sequence line, an additional critical path, or sub-path, may or may not be formed. This can be determined by visual examination if zero-lag sequence lines are marked on the diagram. However, whether it becomes critical or not, a new path has been opened for the transmission of the effects of any further changes. Therefore the network will function differently than before, and this fact must be taken into consideration.

Referring again to the network of Figure 15, it has already been noted that as the duration of Activity 2 is increased, the lag of Sequence Line 2-6 decreases. If the duration of Activity 2 is extended to 15 days, the lag of Sequence Line 2-6 becomes zero. The corresponding revision in network data are shown on Figure 18. In this case the

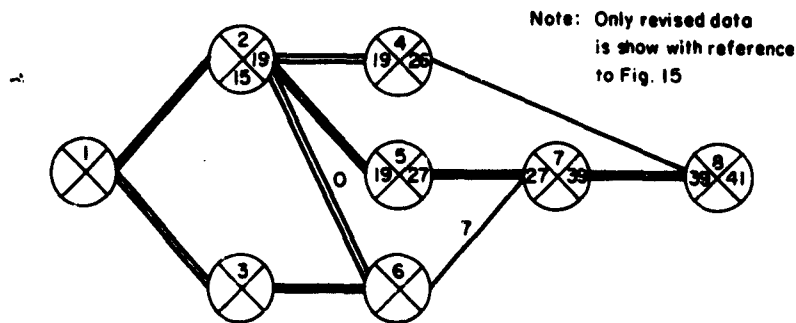


Figure 18. Effect of Duration Increase Equal to NIL

NIL associated with an increase in the duration of Activity 2 from an initial value of 10 days, is 5 days. Thus, if a duration increase of 5

days occurs, the first new zero-lag sequence line is added to the network. For this example no new critical path has been produced. But the new zero-lag path alters the transmission of the effects of any further increase in the duration of Activity 2 in the following ways:

- (1) The scheduling of Activity 6, in addition to that for Activities 4, 5, 7, and 8, will now be affected.
- (2) The lag of Sequence Line 2-6, which has been decreasing, will now remain at zero.
- (3) The lag of Sequence Line 6-7, which has been increasing, will now remain fixed at a value of 7 days.
- (4) The lag of Sequence Line 3-6, which has been fixed and equal to zero, will become positive and start to increase.
- (5) There is no NIL for any further increase in duration of Activity 2, since no positive lag value is being reduced by such a change. Therefore, regardless of the magnitude of the increase in duration, the network will continue to function in the same manner.

As a further illustration of the determination and effect of a network interaction limit, suppose that the duration of Activity 2 in Figure 15 were decreased rather than increased from its original value of 10 days. This change causes the lag value of Sequence Line 2-6 to increase and the lag value of Sequence Line 6-7 to decrease. Since the latter has a positive value of only 2 days, the NIL for this change is 2 days. When the duration of Activity 2 has been reduced by 2 days to a total of 8 days, a new zero-lag sequence line is introduced and creates a second critical path through the project network. The revised data for this change is shown on Figure 19. Note that a second path between Activities 1 and 7, by way of Activities 3 and 6, has now become critical. Every activity in the network except Activity 4 has now become critical.

As noted, the definition of network interaction limit has been based on the change that causes a positive lag to become zero. A change that causes a zero lag to become positive will also affect the manner in which the network functions. This latter form of change is equally

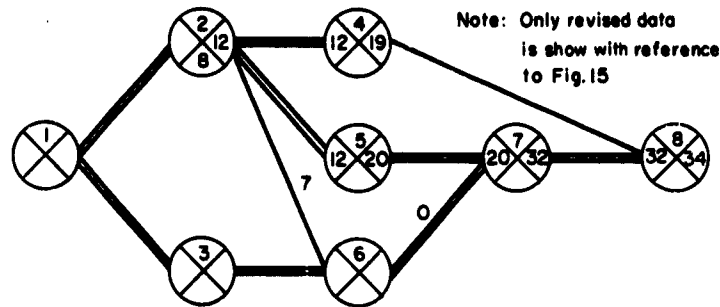


Figure 19. Effect of Duration Decrease Equal to NIL

important. An understanding of its handling and of the reason why it is not included in the definition of the network interaction limit is essential. Two possible situations involving zero-lag to positive-lag changes can occur. In one case the change from zero to positive lag results when the start time of the following activity of the sequence line shifts to a later date. This change is associated with increasing durations or with the rescheduling of a start or finish to a later date. In the second case, the finish time of the preceding activity of the sequence line shifts to an earlier date. This change is associated with decreasing durations or the rescheduling of a start or finish date to an earlier date.

An example of the first case results if the duration of Activity 2 on Figure 18 is increased beyond 15 days. As already noted, any such increase will cause the lag of Sequence Line 3-6 to change from zero to a positive value. While such a change affects the manner in which the network will function when the duration or scheduling of certain other activities is varied, it does not affect the transmission of changes produced by a continued increase in the duration of Activity 2. Therefore it is not considered in determining the NIL for the specific change involving an increase in the duration of Activity 2. In summary of this case, then: (1) the changes in all lag values, including those from zero to positive, are noted and recorded; (2) these changes will affect the manner in which the network transmits certain other changes

that may occur in the future; but (3) these changes are not considered in determining the NIL for the specific change since they do not affect the manner in which the network transmits that change.

There is no need to be concerned nor to attempt to watch for zero-to-positive lag conversions. The first case, just described, does not affect the NIL, and it is taken care of by the routine updating that accompanies all such changes. The second case makes itself known automatically by the introduction of an apparently negative lag value. This case is described in the following section.

Interpretation of Apparent Negative Lag

A sequence line cannot have a negative lag since, according to basic network theory, a following activity cannot commence until a preceding activity has been completed. However, certain scheduling changes appear, at first, to produce negative lags. As an example, suppose that Activity 2 on Figure 19 is reduced in duration from 8 days to 6 days. This change is apparently transmitted through zero-lag sequence lines to Activity 7, whose early start time should be reduced accordingly. However this change is not transmitted to Activity 6 because there is no zero-lag path from Activity 2 to Activity 6. Therefore the early finish time of Activity 6 is unchanged. These results would indicate a decreasing lag for Sequence Line 6-7. But this sequence line already has a zero value and any decrease would require a negative value. Since this is not allowable, a closer examination is necessary. This examination reveals that Activity 7 cannot move to an earlier start time since it is restrained by the completion of Activity 6 as well as Activity 5. But since Activity 5 is free to move to earlier start and finish dates, the lag of Sequence Line 5-7 is converted from a zero value to a positive value by this change. The effects of the change in Activity 2 are not transmitted to Activity 7 but, rather, terminate at Activity 5.

This situation arises only when activities are shortened in duration or when scheduling times are shifted to earlier dates. Also, it can only arise, when there are two or more zero-lag sequence lines

terminating in a common following activity. The rule for handling this case may be stated more generally as follows:

A scheduling change cannot be transmitted to an activity if it would cause the lag of any sequence line terminating at that activity to become negative. Instead, the lag of the sequence line through which the scheduling change would otherwise be transmitted must be made positive.

The application of this rule to the change of Activity 2's duration from 8 to 6 days indicates that it is necessary to recognize that Sequence Line 5-7 is converted from a zero to a positive value. This sequence line is changed on the network diagram to indicate that it is no longer a zero-lag line. Then the network values are updated in the routine manner as shown on Figure 20. This change does not affect the NIL for reducing the duration of Activity 2 below 8 days. In this particular case there is no NIL since this change, although it increases the lag of three sequence lines in the network, does not decrease the lag of any sequence line. But in any case the recognition of the zero-to-positive lag conversion does not affect the NIL for the proposed change.

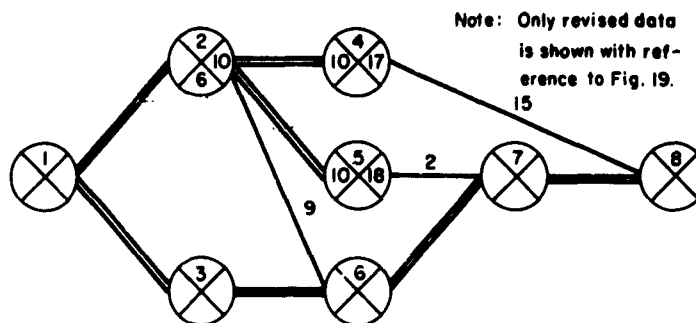


Figure 20. Effect of Further Duration Decrease of Activity No. 2

Concurrent Changes

Up to this point the concepts of the transmission of changes by zero-lag sequence lines and of network interaction limits have been described in relation to changes involving a single activity. However, these concepts are useful also when a change involves two or more

activities. In order to treat them concurrently it is necessary that the change in each of the activities be of the same magnitude and in the same direction and that the effect of changing any one of the activities separately would not be transmitted to any of the other activities in the group. The fact that changes in two or more of the activities, if considered separately, would affect a common activity outside of the group does not prohibit considering the changes concurrently.

As an example of concurrent changes, consider a proposal to expedite the duration of the project shown in Figure 19 by shortening critical Activities 3 and 5. Both of these activities are to be shortened in duration by the same amount, and the effect of shortening either activity separately is not transmitted to the other activity. Therefore, no error will be introduced if they are treated concurrently. The zero-lag sequence lines indicate that shortening Activities 3 and 5 will affect the scheduling of Activities 6, 7, and 8. Further examination indicates that the positive lags of Sequence Lines 2-6 and 4-8 will be reduced. The smaller of these two positive lag values is 7 days, and this becomes the NIL for the concurrent change of the two activities. As long as these two activities are shortened together by seven days or less, the network will transmit the corresponding scheduling changes to Activities 6, 7, and 8.

Suppose that for the network of Figure 19 a plan involving the shortening of Activities 3 and 7 had been adopted instead of shortening Activities 3 and 5. In this case it is not convenient to treat these changes concurrently since any variation in Activity 3 is conveyed to the other, Activity 7. The effect of each day of the proposed change, if considered concurrently, is to shift the scheduling of some activities, such as Activity 6, by one day and to shift the scheduling of other activities, such as Activity 8, by two days. Moreover, the rate of change of lag values would vary. For each day of concurrent change the lag of Sequence Line 2-6 would decrease by one day while the lag of Sequence Line 4-8 would decrease by two days. It is simpler to determine the effects of changes of multiple activities which are connected to one another by chains of zero-lag sequence lines by changing each

activity separately rather than concurrently.

When multiple, independent activities are to be changed in the same direction but by varying amounts, all the activities may be changed concurrently by the number of days of the activity with the least change. Then that activity can be dropped and the remaining activities changed concurrently by the remaining number of days of the next smaller change. This procedure is repeated until the activity with the greatest magnitude of change has been processed. Network interaction limits must receive proper consideration in making each change, of course. Further applications of concurrent changes will be discussed in Chapter V in connection with updating calculations.

Determination of Floats

The calculation of lags for the sequence lines of the precedence diagram provides data from which free floats can quickly be obtained, and it offers alternative procedures for calculating total floats. The relationships between lag values and free floats have been discussed earlier in this chapter. Free floats may be determined in most cases without even comparing multiple lag values by keeping in mind two rules. The first is that if only one sequence line originates at an activity, its lag is the activity's free float. The second rule is that if any sequence line originating at an activity has a lag of zero, the activity's free float is zero. In the few remaining cases where multiple sequence lines originate at an activity and all have positive lags, the determination of the activity's free float requires comparison of these lags to pick the least value; but even this is quite simple.

The conventional method for computing the total float of an activity is to subtract its earliest finish date from its latest finish date. This requires that late start and finish dates be calculated first. One alternative method is to compute the total floats of each activity from the total floats of following activities and the lags of the sequence lines between them. The total float of an activity having only one following activity is equal to the total float of the following activity

plus the lag of the sequence line connecting them. This case is illustrated in Figure 21 where the total float of Activity C and lag

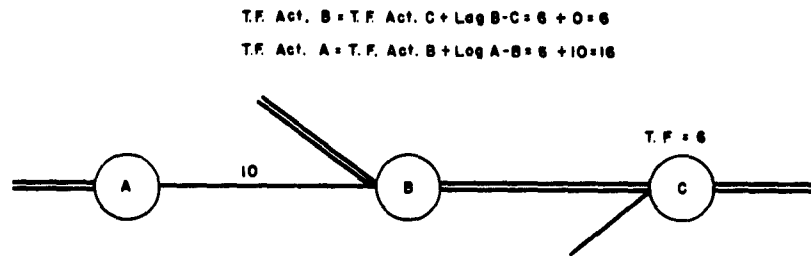


Figure 21. Computation of Total Floats - Single Following Activity

values are known. The total float of Activity B, and then of Activity A, can be simply calculated as indicated. The total float of an activity having multiple following activities is equal to the least of the values obtained by adding the total float of each following activity to the lag of the sequence line to it. This case is illustrated in Figure 22 where the total floats of Activities B, C, and D and the

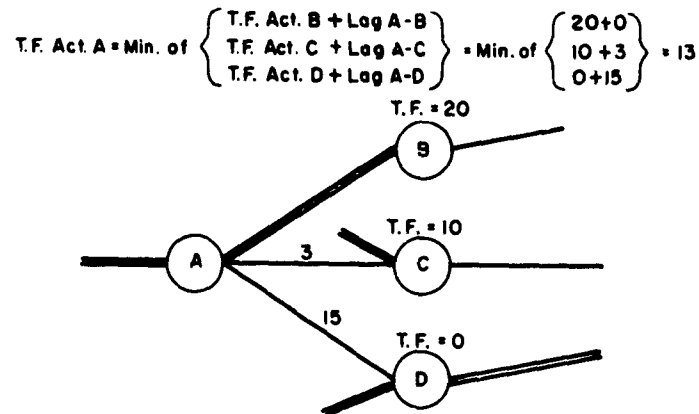


Figure 22. Computation of Total Floats - Multiple Following Activities

lag values are known. The total float of Activity A is determined as indicated. These total float calculations assume a knowledge of the total floats of following activities. Since the total float of the final project activity is always zero, this activity may serve as a starting point. By backtracking across the network, total floats may be calculated directly for each activity without the necessity for calculating late start and finish times. Where a complete set of late start and finish data is desired for other purposes, such as resource allocation analysis, the conventional approach may be better. Where late start and finish data is just a means for obtaining total float data, the alternative approach may be better.

Of greater importance is a method for determining the total float of a specific activity within a network without calculating either a complete set of total float data or a complete set of late start and finish data. By definition, the total float of an activity is equal to the number of time units that its completion date can be extended before project duration is affected. Therefore total float of a specific activity, can be determined by finding the number of days that its duration must be increased to make it critical. To do this, the NIL for an increase in the activity's duration is determined. If the sequence line that is converted to a zero-lag value creates a new critical path, the NIL is equal to the total float of the activity. If not, the lag values must be updated and the NIL for an added increase in duration of the activity is determined. This procedure is continued until a new critical path is produced. The cumulative change, which is equal to the sum of the NILs up to this point, gives the total float of the activity. Generally only one or two steps are required for this determination. For example, what is the total float of Activity 2 in Figure 20? If its duration were increased, the NIL would be 2 days, as determined by the decreasing lag of Sequence Line 5-7. If this lag becomes zero, a new critical path will be created and Activity 2 will become critical. Therefore, its total float is 2 days.

The ability to determine the total float of any given individual activity is important because it eliminates the necessity for continuously

maintaining a completely updated set of total float data. In turn, this will make updating by non-computer methods more practical, since less data must be updated.

Applications of Network Mechanics

An understanding of the basic network elements and mechanics, discussed in this chapter, permits the development of new procedures for more rapid and practical non-computer applications. Further aspects of network mechanics involving the introduction of new activities or the deletion of current ones in existing networks will be discussed in connection with schedule updating procedures in Chapter V. The breakdown of complex networks into simpler subnets is also an application of network mechanics and will be discussed in Chapter VII.

CHAPTER IV
AN ANALYSIS OF DATA ELEMENTS AND APPLICATIONS

Evaluation of Scheduling Data

At the beginning of Chapter III it was pointed out that the updating of data was the major factor in establishing the range of practicality of non-computer procedures. In order to extend this range, three approaches were suggested there for modifying and increasing the efficiency of non-computer updating methods. The first of these proposed approaches, increased selectivity in setting data requirements, forms the subject matter of this chapter.

Updating of data in both computer and non-computer applications of the critical path techniques has generally been too infrequent to realize the full benefits of these techniques. In the case of the non-computer applications the chief reason for the reluctance to update is the amount of computational effort which is required. Of the several ways for reducing the amount of effort necessary to keep a network up to date, one rather obvious method is to reduce the amount of data to be maintained. In computer processing the amount of data computed and printed out, over and above that which is absolutely essential, has a relatively small effect on the total effort and expense, of making a rerun. This is not true of non-computer processing. Here there is almost a direct relationship between the amount of data and the effort required to update it. Therefore, it becomes much more important to be selective in the information that is demanded and to carefully evaluate the necessity of keeping each of its various groupings current.

In order to arrive at conclusions on this matter of increased selectivity in data requirements, each data grouping will first be examined individually to evaluate its importance and, in some cases, to consider alternative updating methods. These groupings include: duration, early start time, early finish time, late start time, late

finish time, free float, total float, lag, and criticality. Next, each of the major critical path applications will be examined in order to determine the importance of the various data groupings for that application. These applications include: original planning on rough diagram, master plan presentation on redrawn diagram, field schedules, job control, and replanning techniques. Finally, recommendations will be made as to which data groupings should be updated as a matter of routine, which may be updated on an individual basis as needed, and which need not be maintained.

Data Elements

Durations

Activity durations serve not only as output information on the various schedules and diagrams but also are the basic input data on which all other calculations depend. Therefore they must be kept current just as the diagram logic must be kept up to date. Updating activity durations simply involves changing the previously estimated duration whenever the need arises. This need may result from actual field performance in a different length of time than estimated, from a re-estimate of time requirements based on a more accurate knowledge of the conditions to be encountered, or from a re-estimate of time requirements based on intentional changes in the manner of performance. These latter changes often result from time-cost tradeoff studies or from employing idle resources.

Early Start Times

Early start times for every activity are computed as part of the procedure for determining project duration and critical activities. They are also required in order to determine lag values which, in turn, are necessary in the updating procedures to be discussed in the second half of this chapter. Therefore it is clearly essential that they be calculated as part of the original data. The next question is whether or not they must be kept currently up to date. It will be found that lag values and the criticality of activities can be updated without actually updating early start data. However, field schedules and job

control require currently updated values of scheduled start or scheduled finish times. Note that it is not essential to maintain both start and finish time data since knowledge of either one and of activity duration permits a simple determination of the other. This report places more emphasis on start time data and proposes that only the scheduled start times be kept updated on a day-by-day basis. Wherever practical, scheduled start times are the early start times. When an activity is to be scheduled to start later than its earliest start time an artificial restraint activity can be introduced, as discussed in Chapter III, which will make the scheduled start equal to the earliest start. In the absence of such restraints, a scheduled start later than the early start time will be preceded by an "S." As suggested in Chapter 3, a record of the reasons for setting scheduled start dates later than early starts should be maintained. In view of the importance of early start times it is clear that they should not only be calculated as part of the original data but that they should be routinely kept current.

Early Finish Times

Early finish times for each activity must be computed in the process of determining project duration, critical activities, and lag values. However, these other items can be updated without the necessity for updating early finish times. Therefore, unless early finish times are important themselves, there is no reason for keeping them up to date. The facts that it is not essential to update both early start times and early finish times, that by not updating one of these data groups the total effort involved is reduced by one-half, and that this report has chosen to emphasize early start times rather than early finish times, results in the recommendation that early finish times not be updated.

The early finish date of the final project activity has special importance. This gives the project duration. It is recommended that an event-node representing project completion be introduced at the end of each network. Since an event has zero time duration, the early start time of this event, which is to be kept updated, will give the currently estimated project completion date. The same is true for any

other activities whose completion dates are of special significance. An event node may be introduced to represent these important "milestones," and the up-dated early start times for these events will indicate their current scheduling.

Late Start Times

Data on late time and total float are closely related. The purpose of calculating late times may be to use them with early times in order to determine corresponding total floats. Conversely, if total float is determined by some other method, late times may easily be obtained by adding total floats and corresponding early times. Although it may be a convenience to calculate and maintain both late times and total floats, it is far from essential to do so. Therefore, when non-computer methods are being employed, it is suggested that only one of these sets be maintained. Which one is chosen will depend on which is most directly useful and on which may be kept up to date with the least effort. If formal resource allocation procedures are to be used, late time data is often required and should be maintained rather than total float data. Moreover, there are certain shortcuts in updating late times that favor their selection. These will be explained in the next section.

It has been indicated that late times need not be calculated at all if they are not to be kept current, and if total float can be determined by an alternative method. However, if the decision is made to calculate and maintain late times, the conventional "backward pass" calculations are generally the easiest procedure for obtaining them. They require computation of both late start and late finish times. Just as in the case of early start and finish times, if one of these data sets is kept updated and if activity durations are kept current, the remaining data can be easily obtained if needed. Therefore, even though both late start and late finish times are calculated at first, only one of these sets need be maintained. It is assumed here that for procedures that require late time data, late finish times are generally more useful. However, there are exceptions that may make it preferable to maintain late start times instead. For example, resource allocation

procedures may make direct use of late start times rather than late finish times.

In summary, in situations where total float data is more advantageous than late time data and can be obtained by alternative methods, late start times need not be calculated at all. When it is preferable to develop late time data, late start times will be calculated but generally will not be updated unless a particular application places more emphasis on them.

Late Finish Times

The difference between the earliest start time of an activity and its latest finish time establishes a total time range over which an activity may be rescheduled and its duration varied without affecting project length. The latest finish time serves as a limit for the shifting of the activity's completion date, and resource allocation procedures attempt to observe this limit. Therefore late finish time data often is required in resource allocation procedures and should be maintained up to date. As mentioned in the preceding section, some of these procedures may utilize late start time data instead.

Total float data may also be used to establish scheduling limits and to indicate criticality. If either total float or late time is available, the other may be calculated easily if needed (assuming that early start times and activity durations are to be maintained updated). Therefore, both sets are not essential. In the preceding section it was indicated that certain shortcuts favored maintaining late time data. These will be explained here.

First, it is assumed that late finish times for each activity have been calculated by the conventional "backward pass" routine and also that the corresponding project completion date has been noted. It is further assumed that the changes to be considered here include both changes in activity durations and the rescheduling of activities to start and finish at different dates than originally scheduled. The effects of these changes may be divided into two patterns depending on whether the activity changed is non-critical or critical. Considering the non-critical case first, it may be stated that only late finish

times of activities that precede it are affected by a change in a non-critical activity. If we assume that the network diagram is plotted in such a manner that the time flow is from left to right (e.g. time-scale or sequence-step diagramming), then only activities to the left of the changed activity can be affected. This is because late finish times are calculated by a backward pass and, therefore, the change can only be transmitted to the left. The effect of a change in a critical activity is different, since the change causes the final project completion date to shift. In this case the late finish times of all activities to the right (or not preceding) the changed activity will be altered by the same amount and in the same direction as the final project completion date. This is because this shift in project completion is transmitted by the backward pass calculations to all of these activities. However, this shift is not transmitted through the changed activity and, therefore, some of the late finish times of activities to the left of the changed activity will be affected while others will be unaffected.

The updating "shortcut" is based on the fact that the late finish times of later activities follows a known pattern. For changes in non-critical activities, they are unaffected. For changes in critical activities, they follow the project completion date shifts. Fortunately, in the majority of situations it is the data for activities following the changed activity that should be updated. For example, in the routine updating required by changes in field operations as the job progresses, the effects on earlier activities are unimportant since these activities have already been performed. In resource allocation work, which is a principal application of late time data, most procedures involve working from left to right across the network diagram. Again, it is only important to update late finish times of later activities than the one being rescheduled, since the earlier activities have already been rescheduled. Of course, after a complete resource allocation rescheduling run has been made from left to right across the network, it is necessary to recalculate all late time data before another run is made, or if this data is needed for any other subsequent use.

In summary, if either late time data or total float data is needed, it is not essential to maintain both. For applications where only the effects of changes on activities concurrent with, or later than, the changed activity are important, there is a definite advantage in maintaining late time data.

Free Floats

The method of determining free floats from lag values has been discussed in Chapter III. Other updating procedures of this report require that lag values be kept updated at all times. Each time a lag is changed, it is a simple manner to observe whether or not this lag did determine, or will now determine, the free float of the preceding activity. If so, the free float may be changed, and updating is a relatively simple procedure. Free float values are useful for field schedules, job control, and replanning procedures. Therefore, it is recommended that free float data be calculated and kept current at all times.

Total Floats

The calculation of a complete set of total float data at an early stage of job planning serves the purpose of giving the planner an overall awareness of the amount of scheduling leeway available for each and every activity. This data may be calculated by the conventional approach of first calculating early finish times and late finish times and then subtracting the former from the latter. If late finish data is not wanted for other purposes, the alternative method described in Chapter III, utilizing lag values and total floats of following activities, may prove more efficient.

The validity of total float data has a short life. Any change in an activity duration or in scheduling will generally produce a number of different total floats. Therefore, to be useful, total floats must be updated frequently. Since extensive changes are often involved, considerable effort is required to update the entire data set. Since only a knowledge of the current value of total float of the activity being changed is sufficient for most purposes, it becomes more practical to update total floats on an individual basis.

A method for determining total floats of individual activities was

presented in Chapter III. It involved calculating the cumulative network interaction limits in making the activity become critical. This approach is doubly useful since the calculations not only give an updated total float but also reveal which other activities will be affected and by how much. Whenever a change in activity duration or scheduling is proposed which will extend the completion time of the activity beyond its free float limit, other activities are shifted. Theoretically, project completion is not delayed because these other activities also have float. Practically, any time that activities scheduled for certain dates and involving other operations are shifted, delays leading to later project completion or other undesirable effects are likely to occur. Therefore when a change is proposed in an activity, the planner is not only interested in its total float but also in the other activities affected. Even a complete, updated set of total float data does not provide this information.

This report recommends that total float data be computed during the early job planning stage and that it be recomputed occasionally to provide a general awareness of the degree of criticality of the various activities. It recommends that instead of maintaining this data at all times, total floats be computed for individual activities as needed.

Lags

The lag values of network sequence lines are basic data for the updating procedures of this report. Therefore it is essential that they be kept up to date themselves. Start and finish times calculated or established at the time when the original network is completed are used to compute lag values. As changes occur, these lags may be updated by merely determining if the preceding and following activities of their respective sequence lines are affected. It is not necessary to update start and finish dates for this purpose.

Criticality

Critical activities should be known at all times. This is not difficult. The original critical path may be traced out on the network diagram by backtracking from the final project activity along zero-lag

sequence lines. Since lag values are kept current, lag changes from positive to zero or from zero to positive can be noted on the diagram. It will become obvious whenever a new critical path is created or an existing one eliminated. These paths in turn indicate which activities are critical and permit this knowledge to be kept up to date.

In the section on Total Floats it was recommended that a complete set of total float data be computed occasionally in order to give the planner an awareness of the relative criticality of activities but that no attempt be made to keep this data updated at all times. A possible compromise in this matter is to maintain an updated record of just those activities that are critical or near-critical. All activities that are within a certain number of days of critical can be considered to be "sub-critical." The specific number of days chosen will probably vary with overall project duration and other factors, and it is best to make this a judgment decision for each case. The existence of sub-critical paths can be determined by noting all sequence lines that terminate at critical activities and that have positive lags less than the limit set for sub-criticality. Then, the sub-critical paths can be traced from the terminals of these sequence lines towards the beginning of the project along zero-lag sequence lines. In some cases, the lag value that determines sub-critical paths is less than the sub-criticality limit and there are other positive-lag sequence lines terminating in this sub-critical path whose incremental lag value still produces a total float within the sub-critical limit. These sequence lines can establish additional sub-critical paths that may be traced. By following this procedure all subcritical paths and corresponding activities can be determined quite easily. These sub-critical paths can be updated as changes take place with less effort than is required to update all total floats.

Data Requirements and Applications - General

As a result of the foregoing analysis of the various data sets, certain recommendations have been made. Before summing up these recommendations, the major applications of this data will also be examined.

One objective of this examination is to relate the recommendations for data requirements to specific applications. A second is to discuss the manner in which this data should be recorded and made available for each application.

There is no single system for handling data that is satisfactory for every job or to every organization. Although recognizing this fact, this report makes certain recommendations concerning the treatment of data as well as about data requirements. It will suggest that some data be entered directly on the network diagram, that some be maintained on separate tabulations under the control of project management, and that others be distributed to the lower echelons of field forces including subcontractors and suppliers. These suggestions should be modified or ignored if by so doing the requirements of the user are better satisfied.

Original Planning on Rough Diagram

An initial rough diagram is constructed as the plan for performing the project is developed. As soon as the planner feels that the diagram is a realistic representation of a reasonable method for performing the work, estimates of the duration of each activity should be made. These estimated times should be entered directly on the diagram at each node. Next, in order to determine project duration, early start times and early finish times for each activity must be calculated. These "forward pass" calculations may be made directly on the diagram. If the resulting project duration is satisfactory, the next step may be to redraw the diagram for a more orderly presentation of the master plan for the project. If a time-scaled diagram is to be used, the early start data is already available. If a sequence-step diagram is proposed, sequence numbers may be calculated and entered directly on the rough diagram.

Should the initially calculated project duration prove to be unsatisfactory, revisions should be made on the rough diagram before preparing a new one. These revisions are often major and may involve changes in logic as well as revisions to durations. However, they cannot be made until the critical path and critical activities are

known. It is generally not worth while to compute late starts, late finishes, and total floats to determine critical activities. An alternative is to compute lag values and to backtrack along zero-lag sequence lines to determine critical activities. This method has the advantage that the lag values may then be used for updating required by subsequent revisions. However, this method of updating, to be discussed in greater detail in the next chapter, is less efficient on rough diagrams than on orderly, left-to-right sequenced diagrams. Therefore, computations of all lag values at this stage is not generally recommended. Rather, the critical paths are determined by backtracking from the final network node along sequence lines where the early finish time of the preceding activity is the same as the early start time of the following. These, of course, are zero-lag lines, but the procedure does not require computation of all lag values or even of all zero-lags. If at the conclusion of replanning on the rough diagram several revisions have been made and some updating performed, it is suggested that early start and finish times be recalculated as a check. This may be done on a fresh print of the rough diagram or by new entries, possibly in a different color, on the same print.

In summary, the data required for rough diagrams includes activity durations, early start times, early finish times, and determination of criticality. All data is originally entered directly on the diagram and is updated directly on it. These recommendations are shown on Figure 25 at the end of this chapter.

Master Plan Presentation on Redrawn Diagram

After initial efforts have produced a job plan that appears to be reasonable and that results in an overall project duration that is acceptable, this plan and the accompanying data should be arranged in a more orderly form. A major step in accomplishing this objective is to redraw the network diagram. This permits improved appearance of the diagram, grouping of related activities in a more logical manner, and showing the flow of work from left to right across the network. The time-scaled diagram and the sequence-step diagram offer two systematic approaches. However, a diagram arranged with good judgment in a less

formal manner may still be just as satisfactory. If a time-scale diagram is used, it has already been suggested that the scale be deleted after the initial presentation rather than to spend the effort to keep the diagram updated to the time scale. The redrawn diagram does not necessarily reflect the final job plan even at the time of the beginning of the project. It simply shows a reasonable plan that may be subject to further improvement by the application of such techniques as time-cost tradeoffs and resource allocation. It also furnishes an improved worksheet on which replanning and corresponding updating procedures may be performed. Although this report goes further than Tech. Report 9 in suggesting that many calculations be performed on the diagram instead of on separate tabulation sheets, the recording of data on separate sheets is recommended at this point. The diagram will serve as a better worksheet if it is uncluttered and if data that may possibly require frequent change is not entered on it. Therefore this report recommends that the redrawn diagram show only the activities and events with descriptive and numerical labels, the sequence lines, the activity durations, and the critical path. It should also indicate those sequence lines that have zero lag.

It is recommended that two separate records be established for data that is to be maintained and updated as changes take place. One is of "Node Data" and the other of "Sequence Line Data." The Node Data sheets list each activity or event by number; show duration, early start time (or scheduled start time), and free float; and provide space for updating these figures. The Sequence Line Data sheets list each sequence line in the network, show lag values, and provide space for updating these figures. For easy reference the sequence lines can be listed in order of the preceding node number and, where more than one line originates at a node, in order of the following node number. In some applications, including those requiring the use of the network analyzer to be described in Chapter VI, it may be preferable to give the sequence lines a single number on the data sheet. Figures 23A and 23B show sample Node Data Sheets and Figure 24 a sample Sequence Line Data Sheet.

Node Data														
Change No.														
Act. No.		0	1	3	4	5	9	10	11	13				
1	Dur	10												
	ES	1												
	FF	0												
2	Dur	20												
	ES	1												
	FF	6	0											
3	Dur	10												
	ES	1												
	FF	36	30					35						
4	Dur	20												
	ES	1												
	FF	40	34					39	25					
5	Dur	60												
	ES	1												
	FF	79	73											
6	Dur	3												
	ES	11			530									
	FF	0												
7	Dur	15	9	5										
	ES	11												
	FF	0												
8	Dur	32												
	ES	14			33									
	FF	44	42		23		28		14					
9	Dur	32												
	ES	14			33									
	FF	58	52		33		38		24					
10	Dur	1												
	ES	26	20	16										
	FF	0		4			9							
11	Dur	20												
	ES	27	21					26						
	FF	0												
12	Dur	14												
	ES	47	41					46	932	29				
	FF	0												
13	Dur	3												
	ES	47	41					46						
	FF	11												
14	Dur	1												
	ES	61	55					60	46	43				
	FF	93								96				
15	Dur	10												
	ES	61	55					60	46					
	FF	0												

Figure 23A. Node Data - Sheet 1

Node Data												Sheet	2
												Act.	16-30
Change No.	0	1	6	7	8	9	10	11					
Act. No.													
16	Dur	23											
	ES	71	65			70	56						
	FF	0											
17	Dur	10											
	ES	94	88			93	79						
	FF	0											
18	Dur	16											
	ES	104	98			103	89						
	FF	0											
19	Dur	3											
	ES	120	114			119	105						
	FF	32											
20	Dur	20											
	ES	120	114			119	105						
	FF	0											
21	Dur	7											
	ES	140	134			139	125						
	FF	0											
22	Dur	15		12	0								
	ES	140	134										
	FF	0											
23	Dur	5											
	ES	147	141			146	132						
	FF	3	0										
24	Dur	2											
	ES	155	149	146		151	137						
	FF	0											
25	Dur	0											
	ES	157	151	148		153	139						
	FF	0											
26	Dur				5								
	ES				21								
	FF				-5	0							
27	Dur						25						
	ES						89						
	FF						12						
28	Dur						3						
	ES						26						
	FF						0						
29	Dur						17						
	ES						29						
	FF						0						
30	Dur						5						
	ES						46						
	FF						100	86					

Figure 23B. Node Data - Sheet 2

Sequence Line Data														Sheet <u>1</u>
														Act. <u>1-45</u>
Sequence Line			Log Revisions											
No.	P. Act.	F. Act.	0	1	3	4	5	6	7	8	9	10	11	13
1	1	6	0			19								
2		7	0											
3	2	11	6	0						Delete				
4	3	13	36	30							35	Delete		
5	4	15	40	34							39		25	
6	5	22	79	73				Delete						
7	6	8	0											
8		9	0											
9	7	10	0											
10	8	17	48	42		23					28		14	
11	9	18	58	52		33					38		24	
12	10	11	0		4						9	Delete		
13	11	12	0									Delete		
14		13	0									Delete		
15	12	14	0											
16		15	0											3
17	13	15	11									Delete		
18	14	24	93										96	
19	15	16	0											
20	16	17	0											
21	17	18	0											
22	18	19	0											
23		20	0											
24	19	24	32					29						
25	20	21	0											
26		22	0							Delete				
27	21	23	0											
28	22	24	0							Delete				
29	23	24	3					0						
30	24	25	0											
31	2	26								0				
32	26	11								-5	0	Delete		
33		28										0		
34	27	30										12		
35	10	28										9		
36	28	12										17	3	0
37		29										0		
38	29	15										14	0	
39		30										0		
40	30	24										100	86	
41														
42														
43														
44														
45														

Figure 24. Sequence Line Data - Sheet 1

At the time that the master plan is redrawn and the key data is tabulated, it is usually helpful to calculate total floats. It has already been recommended that no attempt be made to maintain a complete set of updated total float data but, rather, that total floats for individual activities be calculated as needed. However, at this point and possibly at later points during project performance, it is generally worth the effort to calculate a complete set of total float data in order to gain an awareness of the relative degree of criticality of the various activities. If late start and finish data will be useful for other purposes such as resource allocation studies, they may also be calculated and can be used to obtain total floats. Otherwise the alternative method for obtaining total floats described in Chapter III may be used. It is suggested that these calculations be made on an extra print of the redrawn network.

A summary of the suggested data to be shown on the redrawn diagram of the master plan and on the separate records is given in Figure 25.

Time-Cost Tradeoffs

Although an acceptable plan for job performance should be developed during the rough diagram stage, completion of the redrawn diagram does not signal a termination of planning efforts. Additional replanning aimed at further improvements in the job plan should continue up to the time that the job actually begins and on throughout the course of the job. One of the replanning techniques discussed in Tech. Report 9 and to be covered in further detail in a later report is that of time-cost tradeoffs. It requires intentional revisions to activity duration times. To determine the appropriate revisions requires a knowledge of the critical activities and calculation of network interaction limits. Therefore lag values must be kept updated at all times. The essential results of the revisions are reflected satisfactorily for most scheduling purposes by updating early start times and free floats in addition to the data already mentioned.

The handling of data for this application may be summed up without further examination of tradeoff mechanics. The revised activity durations are changed both on the diagram and on the Node Data sheets.

Early start time changes are determined and changed on the Node Data sheets. Lag values are updated, changed on the Sequence Line Data sheets, and where a change involves a zero-lag value, the results are shown on the diagram. Any resulting changes in the critical paths are also shown there. Finally, free float values are updated as necessary and the changes made on the Node Data sheets. These data handling recommendations are also shown on Figure 25.

Resource Allocation

Another replanning technique aimed at achieving further improvements in the job schedule is resource allocation. Although the principal applications of this technique are generally made before the job starts, continued and more detailed efforts to decrease and smooth out resource requirements should continue during job performance. Resource allocation changes most frequently take the form of rescheduling activity start dates. Sometimes activity durations are also revised when fewer or more units of certain resources are available so that performance time is affected. The data requirements and data handling for this application are the same as for time-cost tradeoffs with two exceptions. One is that rescheduled start times are treated as early start times, revised values are entered on the Node Data sheets, and any necessary updating of other data is performed. The second is that late start and/or late finish time data may be desirable in this work, depending on the procedure used.

Figure 25 summarizes the data handling recommendations for this application.

Field Schedules

The data requirements discussed so far are those needed by the planners in developing a job plan and in attempting improvements to make it even better. In order for this information to have effect, it must be distributed to those who perform the work. As changes occur, earlier information must be updated. The principal point here concerns how much data should be distributed. On the one hand it is desirable to issue all information that can be genuinely helpful to those that have use for

it. On the other, the effort involved increases faster than the original amount of data issued, since all of it must be updated as changes occur. Moreover, some data may be improperly interpreted or applied unless those who use it are required to discuss it with the person responsible for the overall planning effort.

A particular data user needs scheduling information involving his own work. For example, a craft superintendent or a subcontractor is concerned only with those activities for which he has the responsibility. He needs to know the estimated duration of these activities and the scheduled start dates. If the activity has free float, he should also know this figure, since this permits him to take advantage of the extra time to revise his method of performance, his crew size, or his scheduled start date.

This report does not recommend issuing total float data to the lower levels of job management or to subcontractors and suppliers, since it can be misleading to those who are involved in only a portion of the overall activities of a project. It can only be properly interpreted by those who are responsible for the master planning and who have access to the facts concerning all activities. It is true theoretically that using total float time in excess of free float (referred to as "interfering float") does not extend project duration. Rather it uses float time that is shared by other activities. This delays them and causes rescheduling to later dates. However, once field schedules have been issued and commitments made that activities may start on certain dates, postponements can be as serious as if the activity were critical. But any single subcontractor, for example, is not in the position to judge this importance. Moreover, if resource allocation has been considered in earlier planning, rescheduling because of extensions into the interfering float range may seriously affect the resource balances that have been achieved. Again, only the person responsible for overall planning can judge the effects properly.

Finally, there is the question whether or not field schedules issued to others should indicate which activities are critical and which are not. This report recommends that criticality be shown on field

schedules. The fact that an activity is labeled critical stresses its importance to lower-level managers. On the other hand, if an activity has no free float or if the free float has been consumed, and if it still is not labeled critical, the lower-level manager is given notice that some float time may still exist. He is thereby encouraged to bring any proposal for the advantageous use of extra time to top project management. It may prove that his proposal justifies rescheduling other activities. Or it may prove that the activity is, in a practical sense, critical because the change will produce undesirable effects on others or on resource schedules. The point is that if the lower-level manager knows that an activity is non-critical, he is encouraged to make such proposals which, in turn, may be properly evaluated by the planner.

Copies of the network diagram may be issued with field schedules. If these diagrams simply show planning logic and are free of other data except for activity labels, they may be used in conjunction with updated field schedules with a minimum degree of updating. While the master diagram maintained by top project management is updated frequently, it will only be possible to issue updated field schedules periodically. Field schedules subsequent to the initial one need only show changes and may vary in the extent of their coverage. For example, a monthly schedule may indicate changes for the remainder of the job while weekly schedules show modifications that will affect activities to be started within the coming month. Updated time-scale diagrams showing activities only for that portion of the network in progress during the next period (e.g. next 30 days), or a corresponding portion of a bar chart, may be issued periodically if the effort appears warranted. It is very doubtful that updated time-scale diagrams for the remainder of the project can be justified.

In summary, field schedules are as simple as is possible. They cover only those activities that fall within the responsibility of the person to whom the schedule is issued, and show activity durations, early (or scheduled) start dates, free float times, and whether or not activities are critical. Updated schedules show only revisions in this data. Start dates should be converted from working day dates to calendar dates.

Job Control

Both logic and schedule changes are inevitable during project performance. Logic changes include those in the sequencing of activities, deletion or addition of activities, and their combination or further breakdown. Schedule changes include variations in start dates and activity durations. Good job control requires reporting of all such changes as soon as possible after they occur and updating the network diagram and other data to determine the effects of the changes and to evaluate their importance. It is also good practice to maintain a record of changes in the order in which they occur along with the effects that they produce. This may be of value in the equitable settlement of disputes and in determining liability for delays. Job control also involves rendering decisions on proposed changes that will affect activities in addition to the one changed or that will affect the achievement of key deadlines.

The handling of schedule changes is essentially the same as discussed already for time-cost tradeoff and resource allocation. The handling of logic changes involves alterations to the network diagram in addition to handling resulting schedule changes. Rendering decisions on proposed changes can best be accomplished by making the change tentatively, updating all data affected, and then judging the merits of the proposal.

Summary and Recommendations

Figure 25 sums up the data requirements for the six applications of critical path techniques that have been discussed in this chapter. There are five sets of data that are required in most of these and should be maintained and updated. It is recommended that three of these, activity durations, early start times, and free float times, be recorded and updated on Node Data sheets as shown in Figure 23. Although not essential, it is also convenient to record and update activity durations on the master network. The fourth data set, lag values, should be recorded on Sequence Line Data sheets as shown in Figure 24. Zero-lag sequence lines are also marked on the master network for use in updating. The fifth set, criticality of activities, is shown

Summary of Data Requirements									
Application	Duration	Early Start (Scheduled Start)	Early Finish	Late Start	Late Finish	Free Float	Total Float	Log	Criticality
Original Planning on Rough Diagram	E-D / R-D	C-D / U-D	C-D / U-D						C-D / U-D
Master Plan Presentation on Redrawn Diagram	D-T	T		0	0	C-T	C	C-T-D*	D
Time-Cost Tradeoffs	R-D-T	U-T				U-T		U-T-D*	U-D
Resource Allocation	R-D-T	R-T / U-T		0-J	0-J	U-T		U-T-D*	U-D
Field Schedules	S-P	S-P				S-P			S-P
Job Control	R-D-T	R-T / U-T				U-T	I	U-T-D*	U-D
C - Calculated D - Shown on Diagram E - Estimated I - Individually Updated as Needed O - Optional Calculation P - Periodically Updated R - Revised as Necessary S - Schedules for Specific Users T - Tabulated U - Updated After Each Change * - Zero-Logs Only									

Figure 25. Summary of Data Requirements

directly on the master network by tracing critical paths back from the final activity along zero-lag sequence lines.

Having certain data is optional or is only required for specific procedures. Late start or finish times may be useful for resource allocation work. A set of total floats, calculated at the beginning of the job and possibly at occasional later dates, causes an awareness of the relative criticality of the activities. Near-critical activities having total floats values less than a specified amount, may be calculated and maintained, although this is not indicated on Figure 25. Total float values may also be individually updated as needed by the procedures discussed at the end of Chapter III.

When data is updated frequently by procedures that transmit changes through zero-lag sequence lines, an occasional recomputation of all the data by conventional methods will furnish an independent check on computational accuracy. Where non-computer methods are used to supplement computer processing, periodic computer runs will produce a complete and independent set of data to furnish this check.

CHAPTER V

UPDATING PROCEDURES

General

Three approaches were proposed at the beginning of Chapter III for extending the range of practicality of non-computer methods. The second of these involves updating procedures. It was pointed out that the conventional approach to updating is complete recomputation of all data and that, in the case of computer processing, this is the simplest approach. It was also indicated that in the case of manual processing, procedures that require only recomputing data affected by the change can considerably reduce the amount of effort involved for updating. The purpose of this chapter is to develop and describe such procedures.

Three basic types of change that make updating necessary are:

- (1) Changes in the duration of an activity
- (2) Rescheduling activities
- (3) Changes in the project plan.

Changes in activity durations are to be expected because estimated durations will, at best, be only reasonably close to actual performance. Durations may be changed intentionally either to expedite the job or to take advantage of available extra time. Rescheduling of activities often results from replanning efforts, such as leveling of resource demands, directed towards improving job performance and reducing costs. Rescheduling may also be required when field forces fail to start an activity on the scheduled date. Changes in the job plan also are almost certain to occur because of changed conditions, unforeseen opportunities or difficulties, or the development of better and more detailed plans based on additional knowledge of the job. These changes can include additions of new or deletions of existing activities, alterations in sequencing of activities, or the modification of an entire section of the network involving multiple activities,

and a combination of these types of changes.

The procedures of this chapter are based on using a print of the network diagram as a worksheet and transmitting the effects of changes through zero-lag sequence lines. It is desirable to clearly mark the zero-lag lines on the worksheet in order that the paths formed by them may always be apparent. Since, during the updating procedures, zero-lag sequence lines may become positive lag lines, a marking method should be used that permits easy alteration. One possibility is to double each zero-lag line by adding a parallel line in pencil which can be deleted by erasing. Also for these procedures it is necessary to temporarily mark activities and sequence lines affected by a change. This must be done before the time at which the corresponding values are actually updated on either the diagram or on a separate tabulation. One effective method for accomplishing this is to mount the print of the network diagram on a lightweight steel sheet, so that small, inexpensive magnets can serve as markers. It has been found that three different colors (or shapes) of marking magnets are helpful; one for activities, and one each for decreasing and increasing lag values. When the marking procedure is completed, the magnets may be removed as each corresponding change is recorded. There are, of course, other ways to indicate changes than with magnets.

Duration Changes

The first step in updating is to determine and temporarily mark those activities that are affected by the change. Only the finish time of the activity whose duration is changed will be affected; however, both the start and finish times of all following activities which are connected to the changed activity by paths of zero-lag sequence lines will be affected. In order to avoid missing any activity that should be marked, it is recommended that they be scanned in a systematic top-to-bottom, left-to-right pattern. After having marked the activity whose duration has been changed, the next step is to mark those activities, if any, to the right of it that are connected to it by zero-lag sequence lines. Then, proceed to the first marked activity to the right and repeat the procedure. If more than one activity on the same

vertical line is marked, process the top one first and work down. After all activities on that line are treated, proceed to next closest marked activity to the right and repeat. The procedure is completed when there are no remaining marked activities to the right. It has been assumed that the diagram is plotted in a left-to-right time sequence.

Figure 26A presents an example of the procedures to be described in this section. It refers to the simple pier job of Chapter II.

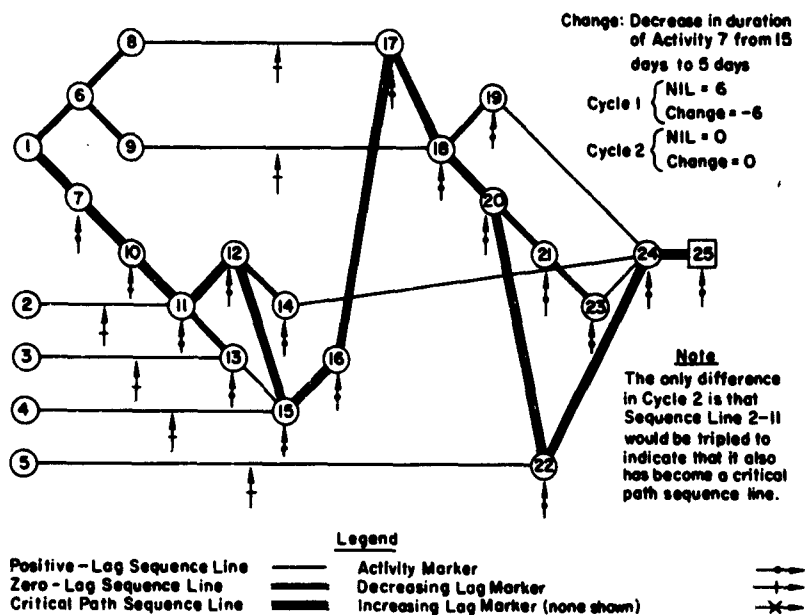


Figure 26A. Updating for Duration Change - Cycle 1
(Cycle 2 similar)

This diagram is plotted to a sequence-step scale, and descriptive activity labels have been omitted for simplicity. The initial data for this project is shown on the Node Data sheets of Figures 23A and 23B and on the Sequence Line Data sheet of Figure 24. It will be assumed that, at the outset of this job, a different method was conceived for demolishing the existing pier and that this method reduces the duration of Activity 7 from 15 to 5 working days. The first step in updating to reflect the effects of this change is to place activity markers on each affected activity as explained in the preceding paragraph. Usually these markers are placed directly on the node symbols but here they

have been placed just beneath the corresponding node symbol on Figure 26A to present a more legible diagram.

As the scheduled times of the marked activities change, certain lag values are subject to change also. Should any positive lag change to zero or any zero lag become positive, a new transmission pattern for the effects of further change is created. Therefore, before scheduling values may actually be updated, the network interaction limit must be determined. This may be accomplished by marking those sequence lines that have decreasing lag values and then, by checking, find which has the minimum value. In the case involving a decrease in duration, a decreasing lag will result when the start time of the following activity of any sequence line is shifted to an earlier date while the finish time of the preceding activity of the same sequence line is unaffected. In order to mark sequence lines having decreasing-lag values, another systematic run is made following the same top-to-bottom, left-to-right pattern. A different colored or shaped marker is used to designate decreasing lags. In this pass the sequence lines terminating at each marked activity are examined to determine whether or not they originate at unmarked activities. If so, they will have decreasing lags and should be marked accordingly. If they originate at a marked activity, their lags will not change and they should not be marked. The forward pass across the network starts, in the case of a decreasing duration, with the first marked activity to the right of the activity whose duration is changed. The start time of the changed activity does not shift and, therefore, the sequence lines terminating at its node are not considered. If the change involved an increasing duration, the forward pass would commence with the changed activity rather than with the first marked activity to its right. The procedure is continued until the last marked activity to the right of the diagram has been processed. Then the lag values of all the marked sequence lines are checked on the Sequence Line Data sheets to determine which has the least value. This least value establishes the NIL. Should the NIL be zero (which can only occur in the case of a decreasing duration), this is a special case. As discussed in Chapter III, it indicates the necessity for changing a zero-lag to a positive value. This

point will be illustrated in the example below. After a zero to positive lag change is made, the entire procedure of marking activities and decreasing-lag sequence lines should be repeated, taking the new positive-lag line into consideration.

The normal result at the end of the first decreasing-lag processing run is the determination of a positive NIL. Then a third systematic pass through the network is required to mark those sequence lines with increasing lag values. In the case of a duration decrease, increasing lag values result when the finish time of the preceding activity of a sequence line is shifted to an earlier date while the start time of the following activity of the same sequence line is unaffected. Therefore, on this run, the sequence lines departing from each marked activity are examined, and those that terminate in unmarked activities are temporarily marked to indicate that they have increasing lags. Again, a different colored or shaped marker is used. In this case, the forward pass across the network starts with the activity whose duration is changed (since its finish time shifts) and proceeds to the next activity to the right and so on in the same pattern, until there are no other marked activities to the right. With this step completed, all activities affected by the duration change, all sequence lines with decreasing lag values, and all sequence lines with increasing lag values have been marked, and the NIL has been determined. For the purposes of this discussion the case of a change involving a decreasing duration has been used, but if the duration had increased instead, the procedures for determining decreasing and increasing lag values would merely be reversed.

Returning to Figure 26A, the decreasing lag sequence lines have also been marked. This is accomplished by examining each marked activity to determine whether or not any sequence lines terminating in them originate at unmarked activities. In the case of Figure 26A, six lines with decreasing lags are found. A check of their lag values on the Sequence Line Data sheet indicates Sequence Line 2-11 has the least lag: namely 6 days. This establishes a positive NIL so that another pass must be made to mark increasing lag sequence lines. An examination of

each marked activity indicates that there are no sequence lines originating at those activities that terminate in unmarked activities. Therefore, there are no increasing lag sequence lines.

If the duration change is less than or equal to the NIL, the updating can be accomplished in a single cycle. Otherwise, additional cycles are required. At the end of each cycle, tabulated (or diagram) data is updated as necessary. First, each marker representing a decreasing lag value is removed as that lag value is changed in the Sequence Line Data record. Next, each marker representing an increasing lag value is removed as these lag values are revised in the same record. In both cases, as lag values are changed, a check should be made to determine whether or not these changes cause the free float of the preceding activity of the corresponding sequence line to change. If so, a new value is entered in the Node Data record. Finally, each activity marker is removed as the early start time of that activity is revised in the Node Data record. The early start time of the activity whose duration is changed is not revised, but a revised duration for this activity should be entered in the Node Data record.

If the duration change is greater than the NIL, updating will require more than one cycle of markings. At the end of the first cycle, revisions are made in the tabulated data in the amount of the NIL. Then the entire procedure is repeated for one or more additional cycles until the cumulative revisions are equal to the duration change and the final incremental revisions are equal to or less than the NIL for the final cycle. Generally one or two cycles suffice.

The case of a change requiring multiple cycles is illustrated by the example presented in Figures 26A and 26B. Since the NIL of 6 days is less than the duration change of 10 days, revisions on the Sequence Line Data sheet, Figure 24, and the Node Data sheets, Figures 23A and 23B, are made in the amount of 6 days, and the procedure is then repeated. Before placing markers on the diagram, the diagram is also updated to show Sequence Line 2-11 as a zero-lag line and a new critical path line as well. Markers for Cycle 2 will be exactly the same as for Cycle 1 since the new zero-lag line does not affect the transmission of

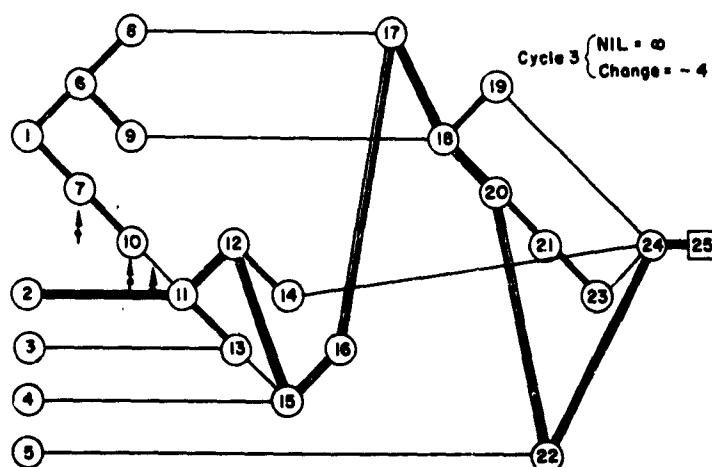


Figure 26B. Updating for Duration Change - Cycle 3

changes in this instance. The results indicate a decreasing lag for Sequence Line 2-11 which already has a zero value. This is the case discussed in Chapter III, and an investigation reveals that the lag of Sequence Line 10-11 must be made positive, thereby eliminating the critical path between Activities 1 and 11. Cycle 2, resulting in a NIL of zero, produces no changes to be recorded on the tabulation sheets. Cycle 3 markers are shown in Figure 26B. Since no decreasing lags are produced, the NIL is infinite. Then the remaining four days of the 10 day duration change can be completed. Revisions for this final increment are shown in the tabulation records of Figures 23A, 23B, and 24.

If two or more activities have changed in duration, and the record is to be updated to reflect these multiple changes, the effects produced by each activity can be determined separately. However, as long as the activities do not directly affect one another (i.e. they are not connected by any path of zero-lag sequence lines), they may be treated concurrently. To accomplish this and at the same time make certain that there is no direct interaction between the activities, commence with the activity having the earliest start date and apply markers to the affected activities. Next, select the activity whose duration has

changed, which has the next earliest start time. If this activity has no marker on it, then it may be handled concurrently. In this case, place additional markers on all activities that it affects and that do not already have markers. Repeat this procedure until markers have been placed for each independent activity whose duration has been changed. Having placed markers on all affected activities, markers for changing lag values may be placed as previously described, the NIL may be determined, and the updating procedure may continue just as in the case of a single activity with changed duration. If one of the activities with changed duration is found to be already marked as a result of the application of markers for an earlier changed activity, then it cannot be handled concurrently with the others. Instead a separate updating run is required for it. Care must be exercised when making revision entries on the tabulation records to see that none of the early start times of the activities of changed duration are revised, even though these activities have markers. Also, in placing markers for changing lags, it must be remembered that these same start times have not changed, in spite of their markers. A solution to this problem is to use a fourth color or shape of marker to designate the activities with changed duration.

Sometimes it is desirable to handle two or more activities concurrently when the changes in their durations are not equal, but are in the same direction. This may also be done if care is exercised. Updating can be performed for all the independent activities concurrently, but only in the amount of the least duration change. At that point those markers that were present solely because of the activity with the least duration change can be removed from the diagram. Care must be exercised in this procedure to avoid removing markers that are also required by one of the activities having a larger duration change. Having removed markers resulting solely from the activity of shortest duration change, the updating procedure is repeated for the increment of time between the shortest and the next shortest duration change. The overall procedure is further repeated as many times as necessary.

In summary, this section has discussed updating of data when

activity duration changes occur. The simplest case involves a duration change of a single activity in an amount less than the network interaction limit. However, larger changes can be handled also. Moreover, duration changes of several activities can be treated concurrently in most cases; this is true even where different activities have unequal duration changes, as long as they are in the same direction.

Rescheduling Changes

In the case of duration changes of activities, the finish time of the changed activity shifts to a new date but the start time remains the same. In the case of rescheduling changes, both start and finish times are shifted to new dates. Such changes are frequently introduced by rescheduling procedures aimed at improving resource requirements. Insofar as the effect of the change in finish time is concerned, it is the same as though it were caused by a change in duration, and the updating may be accomplished by the procedures described in the preceding section. The effect of the change in start time is to change the lag values of any sequence lines terminating at the rescheduled activity. These values may be updated accordingly in the Sequence Line Data record. If the new lag values, in turn, cause new free float values for any of the preceding activities, these free floats should be updated in the Node Data record.

As an illustration of a rescheduling change, assume that Activity No. 6 of the pier network is to be scheduled to start on the 30th working day instead of the 11th. A resource allocation analysis has indicated that this change is desirable to provide continuity of employment for carpenters. As a result of the change, they may proceed with installation of bracing girts as soon as casting yard work is completed. Figure 27 shows the markers placed to indicate affected activities, decreasing lag sequence lines, and increasing lag sequence lines. Since the resulting NIL, determined by Sequence Line 8-17, is greater than the proposed change, the adjustments may be accomplished in a single cycle. Note that the start time of Activity 6 is updated on the Node Data sheet of Figure 23A and that the start time is preceded by the prefix "S" to indicate a scheduled start time different

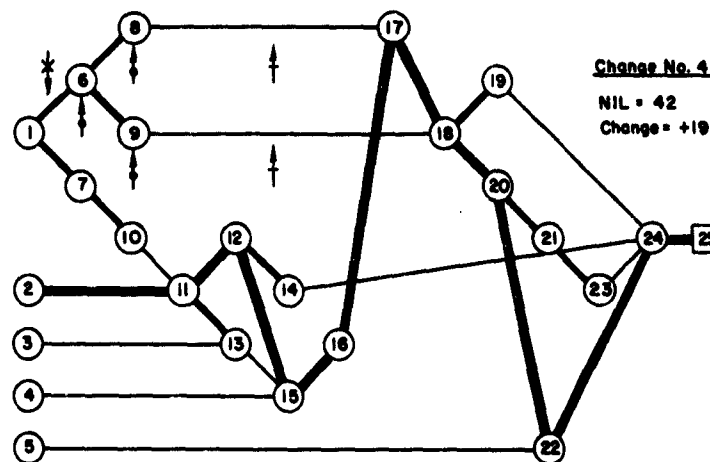


Figure 27. Updating for Scheduling Change

than the earliest start time. Note also that Sequence Line 1-6 is updated on the Sequence Line Data sheet shown in Figure 24. This case differs from that with a duration change; in the latter, Sequence Line 1-6 would not have been affected since the start time of Activity No. 6 would not have shifted.

Logic Changes

While duration and rescheduling changes provide the most common reasons for updating data, the planning logic for the project may sometimes be changed also. One of the important advantages of precedence diagramming, as discussed in Chapter II, is that it permits such revisions to be made on the diagram with relative ease. Data may be updated during the process of diagram revision by methods similar to those already discussed.

The simplest change in the job plan to process is the deletion of an activity. If the activity to be deleted has free float, it can be removed, along with the sequence line preceding and following it, without having any effect on the scheduling of other activities. However, if the lag of one of the deleted sequence lines determined the free float of a preceding activity, then this free float value would have to be corrected. Also, it is necessary to add new sequence lines if the existing sequential relationships preceding and following

activities are lost by the removal of the sequence lines to and from the deleted activity. If new sequence lines are introduced, their lags are computed in the usual manner. If their lags result in new free float values for preceding activities, then those free floats must also be updated.

If the activity to be deleted does not have free float, then it must have at least one zero-lag sequence line extending from it to a following activity. Before it can be deleted, its duration must be reduced until all these zero-lag values become positive. The corresponding changes in data are determined as described in the section on duration changes. When this is accomplished, the activity can be deleted in the manner just explained. It is possible that the duration of the activity to be deleted may be reduced all the way to zero and a zero-lag sequence line between it and a following activity will still exist. In this case the zero-duration activity may be deleted and the sequence lines leading to and from it may also be removed. In their places, new sequence lines from each of the preceding activities to each of the following activities should be added, unless they prove to be redundant. Lag values can be computed for these new lines in the usual manner. A check should also be made to determine whether or not the free floats of any of the preceding activities should be updated.

As an illustration of the deletion of activities both in the case where they have free float and in the case where they do not, assume that a change order is issued for the pier job which deletes the procurement and installation of rubber bumpers, Activities 5 and 22, from the contract. Activity 5, as shown on Figure 27, has free float, and it and Sequence Line 5-22 can simply be deleted without further change. The corresponding updating of the tabulation sheets, Figures 23A and 24, is shown under Change No. 5. Since Activity No. 22 does not have free float (in fact, it is a critical activity), its duration must be reduced until the lag of Sequence Line 22-24 changes from zero to positive or the duration of Activity No. 22 becomes zero. Then it can be deleted. Figure 28 shows markers placed for

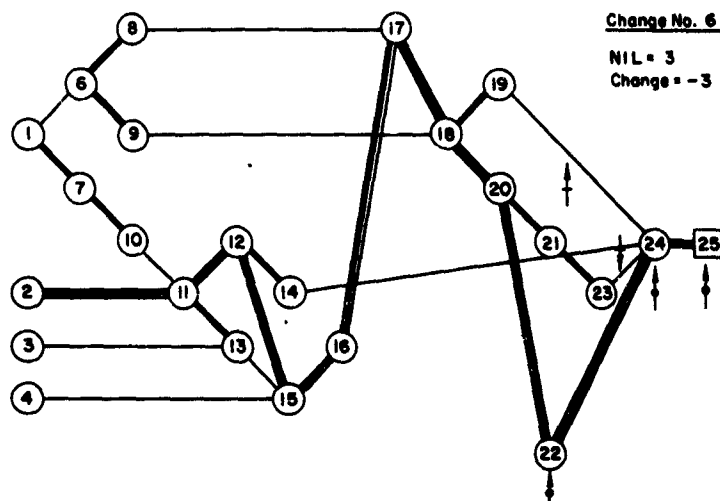


Figure 28. Updating for Deletion of An Activity

this duration change. The NIL, determined by the decreasing lag of Sequence Line 23-24, is 3 days. Any further reduction in the duration of Activity 22 will cause the lag of Sequence Line 22-24 to become positive and its free float also to become positive. Therefore, after reducing the duration of Activity 22 by 3 days, it and Sequence Lines 20-22 and 22-24 may be deleted from the diagram. Corresponding changes on the tabulation sheets are shown as Changes No. 6 and 7 on Figures 23B and 24.

If an activity is to be added to the project plan, it is entered on the diagram and sequence lines are drawn between it and the activities which it must follow and precede. The early start time of the new activity is determined in the usual manner from the early finish times of the preceding activities. Adding its estimated duration, its early finish time is established. If the resulting lag values of the following sequence lines are either zero or positive, the only remaining steps are to enter the new data in the tabulation records and to revise the free floats of preceding activities if necessary. New data includes (1) the new sequence lines and their lag values in the Sequence Line

Data record and (2) the new activity, its duration, early start time, and free float in the Node Data record. However, if the lag value of the sequence lines to any of the following activities proves to be negative, those activities must be rescheduled to a new start date which will increase the lags to a zero value. This shifting is accomplished as discussed in the preceding section on rescheduling changes. If more than one of the following activities needs to be shifted, these changes can be made concurrently. Having completed the addition of the new activity, it will sometimes be necessary to remove old sequence lines that are no longer necessary or valid. This may be due to redundancy or because the new relationship eliminates the need for the former restraints. If such sequence lines have positive lags, the only precaution in their removal is to check whether or not the free floats of any preceding activities are changed. However, should they have zero-lag values, the early start time of the following activities must be recalculated and those activities shifted to earlier dates if necessary. These rescheduling changes are also handled as previously described.

To illustrate the addition of an activity to a network, assume that at the beginning of the pier contract an extra work order is issued instructing the contractor to perform a pile loading test on one of the concrete piles before driving the remainder. The new activity is shown as Activity No. 26 on Figure 29. Change No. 8, shown only on the tabulation sheets, involves adding this new activity and the two adjacent sequence lines after having deleted Sequence Line 2-11. In this case, the lag value of the new following sequence line, No. 26-11, is negative rather than zero or positive. Therefore, this change also requires the rescheduling of Activity No. 11 by the amount necessary to convert the lag of Sequence Line 26-11 from a negative to a zero value. The necessary rescheduling of Activity No. 11 is shown on Figure 29 and also as Change No. 9 on the tabulation sheets. This change also probably affects the improved resource scheduling achieved by the rescheduling of Activity No. 6 and might require another rescheduling of that activity.

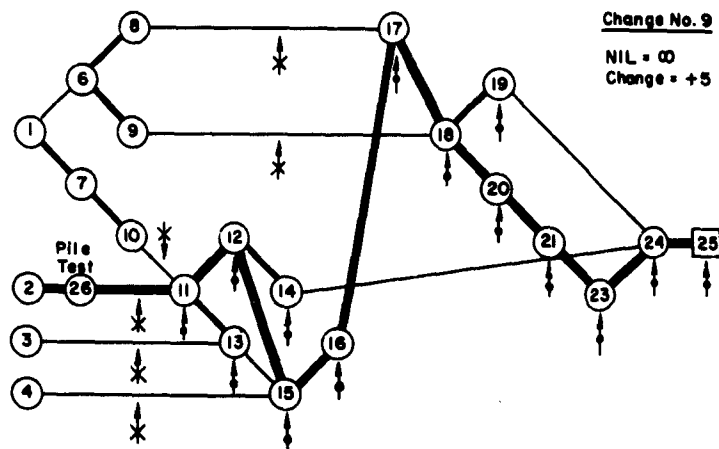


Figure 29. Updating for Addition of An Activity

A change in sequencing of consecutive activities or the modification of an entire segment of the network usually involves both the addition and the deletion of activities. These changes can be made in a single operation rather than a number of separate ones if due care is exercised. All activities and sequence lines to be removed from the diagram are deleted. The remaining activities which preceded a deleted sequence line and the remaining activities which followed a deleted sequence line are temporarily marked to indicate this fact. Next, all new activities and sequence lines are added to the diagram. If any activities that have not been marked now precede new sequence lines, they are labeled accordingly. If any activities that have not been marked now follow new sequence lines, they are also labeled. The start and finish times of all new activities and lag values for all new sequence lines are calculated in the usual manner. Some of these lag values may be negative, indicating the necessity to reschedule the following activities until the negative lags change to zero. Before this rescheduling is performed, the markers designating preceding activities may be removed as soon as the free floats of these activities are checked and revised. The markers for following activities that do not require rescheduling may also be removed. The remaining markers then indicate those activities that must be rescheduled. If

they follow negative-lag sequence lines, they must be shifted to later dates. If they follow only positive-lag sequence lines, they may be shifted to earlier dates. This rescheduling is accomplished as already described, with as many activities shifted concurrently as is possible. As activities are rescheduled, the remaining markers are eliminated. The entire revised section of the diagram should be carefully reviewed at the completion of this logic updating procedure to be certain that no sequential relationships have been overlooked and omitted on the new version of the job plan. Any redundant lines should be removed.

To illustrate a change involving the deletion of several activities and the addition of several new ones and to show how this change may be handled in a single operation, consider the following case: On the eighth working day of the pier contract the owner issues a change order eliminating the steel fender-pile system and substituting a number of 7-pile dolphins in its place. The owner agrees to accept the steel piles already on order. The owner also points out to the contractor that Activity No. 12, Build Thrust Beam, can proceed after the concrete piles under it have been driven instead of delaying until all concrete piles have been driven, as shown on the existing network diagram. It is also pointed out that driving of the dolphin piles can proceed concurrently with riprap placing since the two operations will not affect one another.

Figure 30A indicates the deletion of the activities and sequence lines eliminated by this change. Activities preceding and following deleted sequence lines have been marked accordingly. The deleted activities are also indicated on the Node Data sheet of Figure 23A under Change No. 10. The five deleted sequence lines are indicated on the Sequence Line Data sheet, Figure 24. Figure 30B illustrates the second step which involves adding the new activities and sequence lines to the diagram. One more activity, No. 24, is marked to indicate that it follows a new sequence line. The new activities are added to the Node Data tabulation as shown in Figure 23B. Activity No. 27 has a scheduled start on the ninth day since the order for timber piles could not be placed until the day following receipt of the change order.

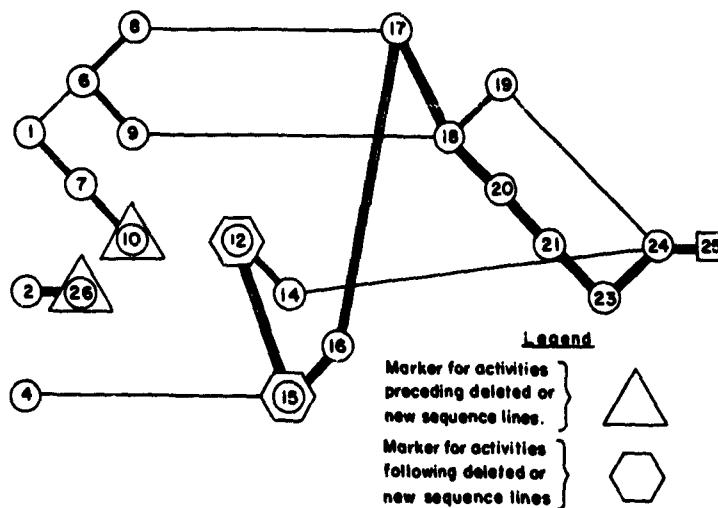


Figure 30A. Updating Involving Multiple Changes - Step 1

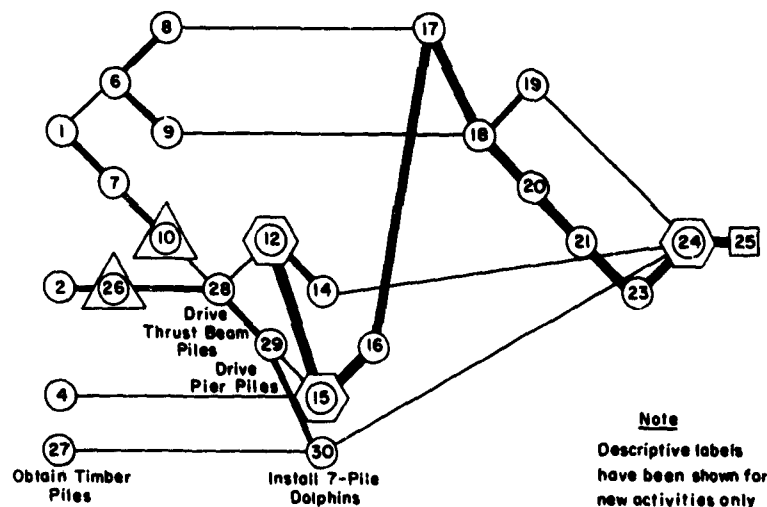


Figure 30B. Updating Involving Multiple Changes - Step 2

Other early start dates are calculated from current data and the diagram relationships. The new sequence lines are entered in the Sequence Line Data tabulation shown on Figure 24, and their lag values are calculated. Those lines having zero lags are marked accordingly on the diagram.

Next, the markers designating activities preceding deleted or new sequence lines may be removed as the free floats of these activities are corrected, if necessary. Neither the free float of Activity No. 10 nor No. 26 needs to be corrected in this case. The marker for Activity No. 24, which follows a new sequence line, may also be removed since this activity does not require rescheduling. However, Activity No. 12 must be rescheduled to start on the 29th day and Activity No. 15 for the 46th day. These two cannot be rescheduled concurrently, since Activity No. 15 is connected to Activity No. 12 by a zero-lag sequence line. However, the rescheduling of Activity No. 12, which must be performed first, eliminates the necessity for rescheduling Activity No. 15.

The rescheduling procedure is the same as that covered in the preceding section. The reader may make these changes as an exercise to test his understanding of the procedure. Here again is a case where the entire change cannot be accomplished in a single cycle. Intermediate results are shown on the tabulation sheets in Figures 23A, 23B, and 24. Change No. 12, not shown, simply involves recognizing that the lag of Sequence Line No. 12-15 has been converted from a zero-lag to a positive-lag value. The final updated diagram is shown in Figure 30C.

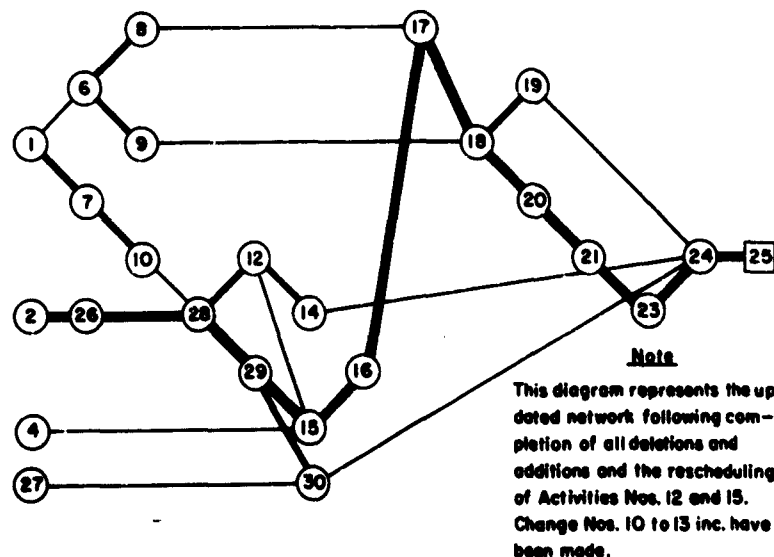


Figure 30C. Updating Involving Multiple Changes - Step 3

Activity No. 6, which was rescheduled earlier to improve resource requirements, has not been re-examined to determine if further rescheduling would be advantageous because of the most recent changes.

Summary

The schedule updating and diagram revision procedures presented in this chapter are probably more difficult to describe in detail than they are to perform. For networks in the general size category of 100 activities or less, these procedures are usually satisfactory. Many actual jobs fall into this category. However, these basic methods may be extended to networks of considerably larger size by means that will be discussed in the next two chapters.

CHAPTER VI

THE DEVELOPMENT AND APPLICATION OF A NETWORK ANALYZER

Analyzer Development - General

As network size increases, the procedures described in Chapter V for updating data become more laborious. This is particularly true when changes are only proposed and the decision to implement them is contingent upon the resulting network interaction limit. In order to determine the NIL, it is required that all the activities that would be affected by the proposed change be determined and temporarily marked. This is time-consuming but is justified if the change is actually made since each marked activity must be updated. However, if the change is not made, the activities are not updated and the markers are simply removed. A more nearly ideal approach would be to determine directly which sequence lines have decreasing lags (and thereby determine the NIL) without having to find which individual activities are affected by the change. Such an approach would be still more satisfactory if these results could be obtained quickly and easily and if it were possible to extend the size of networks which could be updated practically by these means. This chapter will describe an electrical device, referred to as a "network analyzer," which has been developed in an attempt to simplify updating procedures.

The first network analyzer was designed and built to improve the procedures of "Phase III" of Tech. Report 9, which are referred to in this report as time-cost tradeoff procedures. One of the most cumbersome steps in the time-cost tradeoff procedure was to determine the other activities affected by a duration change of a selected activity. This was accomplished by using a triangular precedence matrix (see pages 44-47 of Tech. Report 9). A major disadvantage of this approach was that the matrix took one row and one column for each and every activity of the job. Even with a network of only a 100 activities, the matrix size became prohibitive. Moreover the matrix entries and

updating steps required too much effort and care.

It was found that an electrical network could represent the network diagram and provide a substitute for the precedence matrix at the same time. By energizing a terminal representing the activity whose duration has been changed, the other affected activities could be determined by testing the terminals representing them and noting whether or not they also were energized.

The second cumbersome procedure of the time-cost tradeoff technique was in determining network interaction limits (see pages 43-49 of that report). Having listed every sequence line and its lag value, two columns of entries were required from information provided by the precedence matrix. The first column corresponded to the preceding activities of the sequence lines, and entries were made opposite those preceding activities which were affected by the proposed change. The second column corresponded to the following activities of the sequence lines; entries also were made opposite those following activities that were affected. These entries were especially time consuming since the following activities were not in numerical order and considerable searching was required. Finally, the entries were examined to determine which sequence lines had decreasing and which had increasing lags. An electrical network has been developed that will produce this output information when the terminal representing the activity whose duration is changed is energized, as described above. In this case the state of energization of pairs of terminal points representing network sequence lines must be observed to determine decreasing or increasing lags.

Two models of a network analyzer were built. These in turn have led to a number of proposed improvements for a third.

It soon became evident that these analyzers would be useful for updating purposes as well as for time-cost tradeoff applications. Further work also indicated that the transmission of changes could be accomplished directly on the network diagram by manual methods in a manner similar to their electrical transmission by the network analyzer. This approach was found to be practical for smaller networks and simpler

than the precedence matrix procedures of Tech. Report 9. The results of this work have already been presented in Chapter V. For small networks they furnish a more practical approach than the network analyzer. Nor is the analyzer suitable for very large networks. The amount of patchcord wiring, the cost of the equipment, and the recording of output information all establish certain practical limits for its size. However, in an intermediate range of network sizes, say from 100 to 250 activities, the use of a network analyzer appears to be advantageous. This also happens to be a size range into which a large number of jobs fall.

Electrical Network

An electrical model of a network diagram may be easily constructed using patchcord wiring. Terminal points represent activity nodes, and the patchcord wires sequence lines. The updating procedures discussed in Chapter V were based on the principle that changes in one activity were conveyed to other activities through zero-lag sequence lines. This principle can be translated to the electrical network by introducing input voltage at the terminal point which represents the changed activity. Each zero-lag sequence line is represented by closed-circuit wiring while positive-lag lines are represented by open-circuits. An important requirement is that the flow of current be uni-directional. When a change in one activity is conveyed to other activities by zero-lag sequence lines using the methods of Chapter V, this change is conveyed in but one direction. If early, or scheduled, start and finish data is involved, the change travels only toward later activities. If late start and finish data is involved, a backward pass conveys changes only to earlier activities. In an electrical sense this uni-directional flow can be accomplished by placing a diode, properly oriented, in the wiring that represents each sequence line.

From this simple arrangement it is very easy to obtain the required information for updating. If a potential is applied to the terminal representing the changed activity, a check of the terminals representing the other activities will indicate which are energized and, hence, affected by the change. A check of the two terminals representing the

the preceding and following activities of each sequence line will indicate lag changes. If neither, or if both, terminals are energized there is no change. In the case of duration reduction or in the case of a rescheduling to an earlier date, if only the terminal representing the preceding activity is energized, this indicates an increasing lag. In the case of an increased duration, or a rescheduling to a later date, it indicates a decreasing lag. Of course, if only the terminal representing the following activity is energized, the reverse conclusions may be drawn. To obtain output information concerning affected activities and lag changes, it is important to lay out the circuitry in such a manner that the energization of the appropriate terminals may be checked easily and rapidly.

A network analyzer, then, will consist of both temporary and permanent circuitry. The temporary portion is created by patchcord wiring and will represent a specific network diagram. The permanent circuitry will be designed to permit input data to be introduced and output information to be obtained in an orderly and simple manner. A more versatile and economical unit will result if the permanent and temporary circuitry are located on separate, connecting units that allow a single permanent unit to be used with any one of a number of temporary units. This permits use of the same basic unit to analyze as many different project networks as desired without having to rewire the networks each time they are studied or updated. This can be accomplished by the use of removable patchboards on which the network diagram is represented by temporary patchcord wiring, and which may be preserved for the duration of that project.

The title "network analyzer" has been used in referring to the device described in this chapter. This device is not a computer. Just as in the manual analysis presented in Chapter V, the results obtained from the network analyzer only indicate changes in values. No adding or subtracting is performed by the analyzer; this must be done separately. It is conceivable that the analyzer could be modified to perform computations. However, this would add considerably to its cost and defeat the objective of producing a simple and economical substitute for more

sophisticated procedures based on computer processing.

Construction and Application of Network Analyzers

Model I

The first model of a network analyzer utilized a discarded computer patchboard and was assembled in less than a day's time at a home workbench. It was intended to serve only as a prototype for further development and had a capacity of only 25 activities. However, this size was ample to permit performance and comparison of the time-cost tradeoff calculations for the 18-activity network used as an illustrative example in the discussion of Phase III of Tech. Report 9. It also could perform similar calculations for the 25-activity pier network for which a diagram was presented in Tech. Report 9, and in Chapters II and V of this report. This model did not incorporate the feature of a removable patchboard for the temporary wiring. Model I is shown in Figure 31.

The power source for Model I was a 6-volt battery enclosed in the box shown. Power input to activities whose durations were to be changed was accomplished by patchcord jumpers connecting the power source to numbered holes representing these activities. In the case shown in Figure 31, a change involving simultaneous decreases in the durations of three activities was being checked. Patchboard holes adjacent to those serving as input receptacles were used for output purposes to determine the other activities affected by the change. This was accomplished by inserting a probe in those holes which, if the activity terminal were energized would cause a light on the control box to be illuminated. Actually this output information is useful only for updating purposes after the NIL has been determined. At the time that Model I was built, the only objective was to obtain the NIL in a simpler manner than was done in Tech. Report 9. At this time, updating had not been considered as an important function of the device. Therefore these output points are not shown by separate labeling.

The main upper-right portion of the patchboard was used for the temporary network wiring. This was accomplished using standard

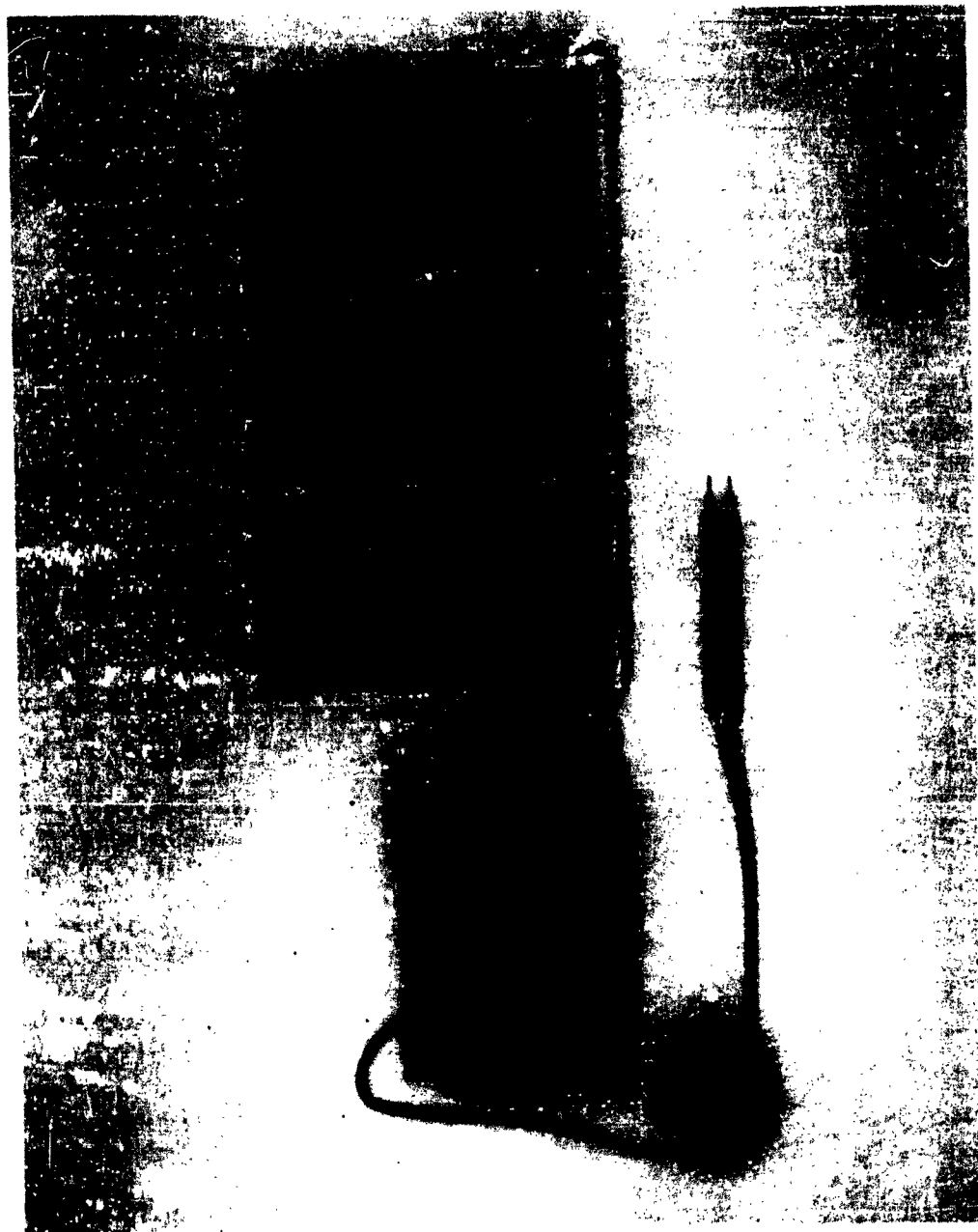


Figure 31. Network Analyzer Model I - Front View

commercial patchcords having built-in diodes. (These are not shown in Figure 31). The permanent circuitry on the back side of the board was laid out in triangular matrix form similar to the precedence matrix of Tech. Report 9, as shown in Figure 32. This space-consuming arrangement proved to be an unnecessary requirement. Patchcord wiring at the bottom of the front of the board routed the activity output to appropriate locations for checking sequence line lag changes. A row of double output holes provided capacity for checking simultaneously the preceding and following activities for as many as 50 sequence lines. This checking was accomplished by the double-pronged probe shown in Figure 31. Output results were indicated by the two lights on the control box. If neither or both were illuminated when the probe was inserted in the two holes representing a given sequence line, the lag of that sequence line was unaffected by the change being made. If only one light were illuminated, a change in lag was indicated. The direction of the change depended on which of the two lights was illuminated and on whether or not the activity durations were being increased or decreased (or whether the activities were being rescheduled to an earlier or later date).

This simple device, although its capacity was intentionally small, proved effective as an aid in making planning changes associated with time-cost tradeoffs. It could be used as an aid in updating scheduling data as well. Its efficiency can be demonstrated by performing the network interaction limit calculations shown on Figure 13 of Tech. Report 9. These can be accomplished rapidly and much more easily with this simple network analyzer. All in all, Model I successfully achieved its purpose.

Model II

A second network analyzer was subsequently built to try out various improvements as well as to achieve a larger capacity. An exterior view of this analyzer is shown in Figure 33 and the interior showing the removable patchboards and the receiver for them appears in Figure 34. This model could have handled a network of 120 activities, but, in order to experiment with a second function to be described below, its capacity was halved to 60 activities.

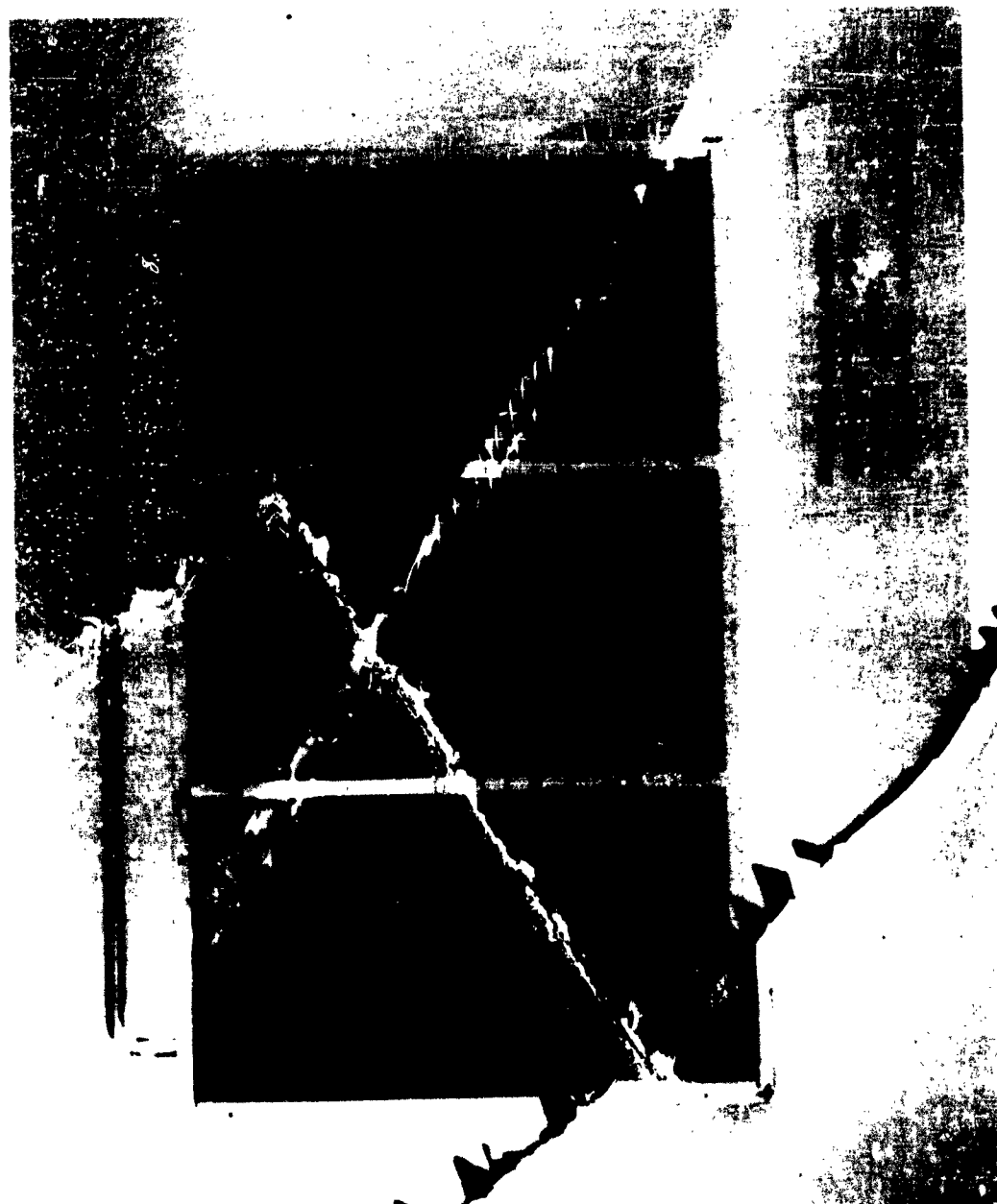


Figure 32. Network Anal.
Rear Model I - Rear View



Figure 33. Network Analyzer Model II - Exterior View



Figure 34. Network Analyzer Model II - Interior View and Removable Patchboards

Model II operates from a 110 volt A.C. power source. The wiring which represents the network diagram is done on removable patchboards, so that the analyzer can serve a number of different projects simply by changing pre-wired boards. Patchboards can be retained throughout the duration of a project and are available to update or replan the job whenever it becomes desirable to do so. The fixed portion of the analyzer was designed to permit combinations of additional identical units to double, triple, or even further extend the capacity as far as other limiting factors would permit.

Input to Model II is accomplished by slide switches on the front vertical panel, there being a separate switch for each network node. Switches corresponding to those activities whose durations are varied or that are rescheduled are moved to the "on" position. Toggle switches on the top of the horizontal panel are connected in series with the sequence lines corresponding to their numbers. When turned "on" they close the circuit of that line and cause it to function as a zero-lag sequence line. In the "off" position, the corresponding sequence line is an open circuit and therefore functions as a positive lag sequence line.

Output from Model II is by means of neon indicator lamps. Pairs of these lamps represent the preceding and following activities, respectively, for each sequence line in the network. These lamps are mounted in the bottom section of the rear vertical panel, with different colors distinguishing preceding from following activities. Lamps at the top of the panel represent each activity in the network, rather than each sequence line, and can be used to identify those activities affected by a given change. This array of lamps permits all effects of a specific change to be observed at one time. In contrast, on Model I, a large number of output terminals were tested individually or in pairs in order to obtain the same information from a single set of output indicators.

It has already been pointed out that the original intent of the network analyzers was to determine the network interaction limit more simply and thereby make the time-cost tradeoff procedure of Tech.

Report 9 easier to apply. However, it became apparent that the network analyzer was capable of providing data for updating schedules and that in practice this function would be at least equally important. As originally conceived, the network analyzer would provide information for updating early start and finish dates and free floats. It later was decided to experiment with modifications that would provide the additional information required for updating late start and finish dates and total floats. This required that duplicate network circuits be wired on the removable patchboards. It also called for duplicate permanent circuits and arrays of output lamps. Thus, to furnish this additional updating information, the capacity of the analyzer was halved.

In the discussion in Chapter V of procedures for updating scheduling data based on the transmission of changes by zero-lag sequence lines, only early (or scheduled) time data was utilized and transmission only in the forward direction was considered. This treatment followed the conclusions of Chapter IV that it was less essential to update late start and finish data and total float times frequently.

If it is desired to update late start and finish data, as will be assumed in the discussion of the Model II analyzer, then a procedure similar to that used for updating early data may be employed. Late start and finish data are used to determine late lags (instead of the early lags already discussed) and to determine network interaction limits based on changes of these lag values. Changes in the scheduling of late start and finish data are transmitted by zero late lag sequence lines in the reverse, backward direction.

From the standpoint of the construction on a network analyzer, two distinct circuits are desirable if the late time data is also to be updated. There are several reasons for the separate circuits. First, the sequence line diodes must be oriented in the opposite direction to assure the proper uni-directional current flow. Second, since the computations for late zero-lags and for early zero-lags are performed independently, a given sequence line may have a zero value in the "early" circuit and a positive value in the "late" circuit,

or vice versa. Therefore, if these zero-lags are to be introduced by the positioning of a two-pole toggle switch, two separate sets of switches and their accompanying circuits are needed. Finally, total float changes for each activity are indicated by a double set of lamps. One indicates changes in the early start time and the other in late start time (or early finish time and late finish time) for that activity. In a similar manner to that of the sequence line indicators, if both or neither of these lamps are illuminated, there is no change in the total float value. But if only one lamp is illuminated this is evidence of a change in total float, with the direction of the change depending on which of the two lamps is energized and on the direction of the change in activity duration or scheduling which is responsible. Since it is necessary to observe two output indicators simultaneously, one in the early and one in the late circuit, it is necessary to have two separate circuits.

The introduction of the late time circuit also requires a modification of the input circuit from the front panel slide switches. Insofar as the early time circuit is concerned, it makes no difference whether the activity whose duration or scheduling is varied is critical or non-critical. In one case the change will be transmitted forward all the way to the final project activity while in the other it will not be transmitted that far. But in both cases the mechanics are the same. Therefore, the corresponding input slide switches need merely be moved to the "on" position to introduce a voltage potential to the terminal of the changed activity. However, in the late-time circuit the mechanics of changes produced by modifying a non-critical activity are different from those produced by a change in a critical activity. This fact requires an additional position on the input switches. If the changed activity is non-critical, the input voltage is applied to the terminal of the corresponding activity, just as in the case of the early time circuit, and the flow is backward through the circuit from that point. However, if the changed activity is critical, the late finish time of the final network activity is changed. Therefore the input voltage must be applied to the terminal of the final activity instead

of that of the changed activity. Moreover, if a duration change is involved, the activity changed will not transmit the effects of the change to the preceding activities. Therefore, the second position of the input switch, for critical activities, not only energizes the final activity but at the same time opens the circuit through the changed activity. The wiring is such that the effect on the early time circuit is the same for either position of the switch.

Materials cost of the Model II analyzer amounted to slightly less than \$500. This includes the patchboard and receiver, all neon indicator lamps, all switches, two complete sets of diodes for early and late circuits, power rectifier, safety devices, and the chassis. Additional patchboards are priced at \$16.50 each. Patchcords are standard and cost from eight to ten cents each, depending on length. The principal cost, however, proved to be the labor of assembly and wiring, which for the construction of a single unit ran about three times that of materials. It is true that these labor costs included the developmental work associated with building a single, new model, but this was counterbalanced to some extent by the service of a technician with superior skill and speed.

Model III (Proposed)

Model III has not been built. If this is done, experience with the first two models suggests a number of modifications. These will be described in more detail as follows:

- (1) Capacity. As indicated earlier in this chapter, the size of network that can be handled advantageously with a network analyzer probably falls in a range between 100 to 250 activities. Below this range the manual methods discussed in Chapter V are satisfactory. At higher levels the network should be subdivided into smaller subnetworks or electronic computer methods applied. Therefore it is proposed that Model III have a capacity to handle networks of approximately 250 activities. Since a unit with half this capacity would probably not compete with the use of manual methods,

there seems to be little advantage to constructing a number of linked smaller units.

In general it has been determined that the number of sequence lines in a network diagram will seldom exceed one and a half times the number of activities. In building Model II, for example, capacity was provided for 100 sequence lines to match the 60-activity capacity. The provision of approximately 50% more sequence line capacity than activity capacity is recommended for Model III. Patchboards and receivers capable of providing this capacity are available as standard units.

- (2) Late-time Circuit. The additional effort and expense of providing and using the late-time circuit appears to be greater than its value. This conclusion was reached in Chapter IV in evaluating the need to maintain various data sets when employing manual methods of updating. Similar reasoning appears applicable when working with the network analyzer. In summary, it was decided that total floats could be calculated for specific activities as needed. Although late start or finish times may be useful in resource allocation procedures, principles have been discussed that permit the intelligent use of this data even though it is less frequently updated.

It is proposed that the late-time circuit be omitted from Model III, since this would about halve both the cost and size of the unit. It also cuts in half the number of patchboards and patchcords required. Of even greater importance is the reduction in the effort required to wire the patchboards and to maintain and tabulate the data involved.

- (3) Output Indicators. A single pair of output indicators as used in the simple design of Model I offers the opportunity for a genuine reduction in analyzer cost and complexity. Model II utilized 520 neon lamps, and the associated wiring

added greatly to labor costs. While it is convenient to have all output data presented instantaneously, it is far from essential. The test probe used for Model I was unsatisfactory because it required too much time and care to scan the output terminals. However, a much more efficient scanning device can be built. Preliminary designs have been made for a linear scanner and for a rotary scanner-counter combination. Either of these would fit the proposed capacity of Model III.

The use of a single set of output indicators also permits certain improvements that are impractical for a large array of output indicators. Consider, for example, the case where both the preceding and following activities of a sequence line are affected by a change and, therefore, the lag value does not change. It would simplify reading output information, which only involves changing values, if this double-output signal could be automatically suppressed. This could be accomplished by a modification of the output circuitry. To carry this concept a step further, it has been pointed out that, when network interaction limits are being determined, only decreasing lags are of interest. Thus, another modification could be a control switch that would eliminate increasing lag output signals. Conversely, when increasing lag values were to be updated, the control switch could be turned to a second position to eliminate the decreasing lag output signals. While these circuit modifications are relatively simple to incorporate in a circuit having a single set of output indicators, they could not be justified for each of several hundred individual output circuits such as those in Model II.

For these reasons, it is proposed that a single set of output indicators and a suitable, efficient scanning device be used for Model III.

If a rotary scanner is used and scanning is performed

by turning a crank, a counter could be geared to the scanner to show the number of the sequence line or activity corresponding to the terminal points then in contact with the output indicators. Whenever the output indicators signalled a changing value during the scanning process, the counter could be read and the number of the network element recorded. Forms for the manual tabulation of these results will be discussed later in this chapter. A further possible refinement is the use of a printing counter that would eliminate visual observation and the necessity to stop and record the numbers corresponding to changed network elements. When an output signal was introduced by the terminals contacted by the scanner, the printing counter would automatically print out the activity or sequence line number corresponding to these terminal points. This refinement can be carried still a step further by a motor driven scanner-counter device that would eliminate the manually turned crank. However, the objective of the proposed network analyzer is to provide an economical and rugged unit that will be useful under field office conditions as well as in a main office. Such refinements must be carefully weighed in view of these objectives.

- (4) Zero-lag Input. The use of the network analyzer requires that the circuits representing each sequence line be closed if the sequence line has a zero-lag value and open if it has a positive-lag value. On Model II this is accomplished by throwing a toggle switch either to the closed or open position. There must be a separate switch to represent each sequence line in the network. This has the serious disadvantage that considerable time and effort is required to properly position this array of switches each time a different network patchboard is placed in the analyzer. This disadvantage will become even more pronounced with a proposed capacity of approximately 250 activities and

400 sequence lines.

An alternative is to close all zero-lag circuits by means of standard double-pronged plugs inserted in a removable patchboard. This patchboard may be a section of the same one on which the other network relationships are represented by patchcord wiring, or it may be a separate board with a separate receiver on the analyzer. In any case, the zero-lag relationships can be pre-wired, and preserved when the patchboards are removed from the analyzer and stored. As changes in zero-lag relationships take place, additional plugs can easily be inserted or existing ones removed from the patchboard. Since changes occur infrequently during updating procedures, the extra effort required to insert or remove a plug instead of throwing a switch is insignificant.

It is possible through the use of this extra patchboard to considerably reduce the labor required to wire the analyzer. On Model II the toggle switches are mounted on a different surface than the receiver, but all network circuits must be routed from the receiver through the toggle switches. If the receiver for the patchboard with the zero-lag plugs were placed on the same surface as the main receiver, or if all of this input wiring were performed on the same board, the length and complexity of the permanent wiring could be reduced and labor costs could be decreased. Panelboards and receivers with the capacity to allow this additional wiring are available as standard items. Therefore, it is recommended that Model III be designed to permit pre-wiring of zero-lag relationships on a removable patchboard.

- (5) Activity Input. On Model II, input power was introduced to the terminal of the changed activity by means of a slide switch. A separate switch was provided for each activity; these were three-position in order to permit a different "on" circuit for critical and non-critical conditions in

the analysis of the late-time data. With the elimination of the late-time circuits, some simplification is possible because a two-position switch becomes satisfactory.

An alternative is to use a pinboard arrangement which would permit the circuit between the power source and the terminal of any activity to be closed by the insertion of a pin or plug. By placing these receptacles on the surface where the activity output scanner is mounted, the amount of permanent wiring might be reduced. However, these proposals are marginal since slide switches are convenient and inexpensive, and the pinboard system requires about the same amount of permanent wiring as the switch system.

In summary, it is recommended that the network capacity of Model III be approximately quadrupled over Model II. The elimination of the late-time circuit will double the capacity of Model II. Therefore, the capacity recommendation actually involves only a doubling of the number of activities and sequence lines that can be handled by Model II. By eliminating the large bank of separate output indicators, the bank of zero-lag toggle switches, and, possibly, the activity input slide switches, a considerable saving in cost would result. It would be partially offset by the requirement of an efficient output scanning device and increased patchboard receiver capacity for the removable zero-lag patchboard. It is roughly estimated that the overall cost for the higher capacity analyzer would be only slightly more than for Model II.

Data Tabulation

The use of the network analyzer does not greatly affect the effort of recording and updating data. It is helpful if the sequence lines can be given a single number rather than the dual label that includes their preceding and following activities. Then permanent, single numbers may be used to label the sequence line output indicators or output terminal points if a scanning device is used. If the scanner has a counter device

the numbers on the counter will refer to single-numbered sequence lines and activities. Where all data is to be maintained directly on the diagram itself, each sequence line may be given a number on this diagram. If data is to be maintained on the "Sequence Line Data" sheets proposed in Chapter IV, the single numbers may be listed on this sheet in a separate column from the dual preceding-following numbers. In this case they need not be shown on the diagram. In working between the diagram and data sheets, the dual numbers would be employed. In working between the analyzer and data sheets, the single numbers would be used. There would be no reason for working directly between analyzer and diagram.

It may be helpful to review the updating procedure as it would be performed using the network analyzer. Consider first the case where all data is maintained on the diagram. Assume that each sequence line has been assigned a single number on the diagram. The first step in updating for a change is to use the analyzer to determine sequence lines having decreasing lags. As each such line is indicated by the analyzer, its current lag value is noted (or temporarily marked). The least of all of these lags values will determine the NIL, and the magnitude of the change for that cycle of updating can be established. These changes are made in the corresponding lag values on the diagram. Next sequence lines with increasing lags are determined by the analyzer. As each one is indicated its lag value is changed on the diagram. Each time a lag value is changed, in either direction, a check is made to determine whether or not the free float of the preceding activity is changed. If so, the new value is entered on the diagram. If any lags change from positive to zero or zero to positive, the corresponding sequence lines are made double or single, to indicate this change. Finally, the analyzer is used to determine the activities affected by the change, and their early start times are updated on the diagram.

There are certain disadvantages to maintaining all data directly on the diagram. After several changes the diagram becomes untidy, and this may lead to errors in reading data from it. Moreover, no permanent record of each separate change is maintained. As proposed in Chapter IV,

separate records for Node Data and Sequence Line Data may be kept. The analyzer may be used to update these records in the same general manner as just described for the diagram revisions. In order that the effects of a given change may be associated with that change, a separate column on the data sheets may be used to record the results of each change. This was done in the illustrations of manual updating discussed in Chapter V.

Another method maintaining a historical record of changes in a useful form is to use "Record of Change" sheets similar to those shown in Figures 35A, B, and C. Every change that occurs and requires updating is given a number and described. After the network elements affected by the change are listed and updated, the "Record of Change" sheet is filed in numerical order. The analyzer is used first to determine and list decreasing lag sequence lines. After the current lag values of these are checked in the Sequence Line Data records, the NIL and magnitude of change for that cycle of updating are determined. Then the analyzer is used to list the increasing lag sequence lines and the affected activities. After the Record of Change sheet has been completed, the corresponding changes may be made in the Node Data and Sequence Line Data records, possibly by an assistant or secretary. The Sequence Line Data records must be updated before another change is processed, but the Node Data records, which generally involve the largest number of changes, need not necessarily be updated until new scheduling data is required. Therefore some of the work may be postponed until a more convenient time, or saved for a secretary to handle. Another advantage is that a new column on the tabulation forms need not be used for every change, since the Record of Change file makes it possible to review and trace out the historical reasons for scheduling modifications by scanning entries in the file. The fact that the Record of Change sheets furnish an orderly, well-documented history of the cause and effect of all job changes is its chief justification.

Figures 35A, 35B, and 35C show, respectively, Change Nos. 1, 2, and 3 as discussed in Chapter V and entered on the tabulation sheets of Figures 23A, 23B, and 24. This change required three cycles. Using the

Record of Change No. <u>1</u>			
Date: <u>2/8/65</u>			
Description of Change:		Duration of activity No. 7 decreased from 15 days to 5 days.	
Reason for Change:		Method replanned to expedite job. Pier to be demolished by floating rig and moved to disposal area by barge instead of demolition by truck cranes and hauling on flatbeds.	
Decreasing Legs	Increasing Legs	Early Starts	
3 4 5 6 10 11	* none	10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	
NIL ----- <u>6</u>			
Total Change ----- <u>10</u>			
Amount Already Processed ----- <u>0</u>			
Amount Remaining ----- <u>10</u>			
Magnitude and Sign of Change - This Sheet ----- -6			

Figure 35A. Record of Change No. 1

Record of Change No. <u>2</u>			
<u>Date:</u> <u>2/8/65</u>			
<u>Description of Change:</u>		<i>See Change No. 1</i>	
<u>Reason for Change:</u>		<i>See Change No. 1</i>	
Decreasing Lags	Increasing Lags	Early Starts	
3 4 5 6 10 11	*		
NIL ----- <u>0</u> Total Change ----- <u>10</u> Amount Already Processed ----- <u>6</u> Amount Remaining ----- <u>4</u> Magnitude and Sign of Change - This Sheet ----- SL12 = Pos.			

Figure 35B. Record of Change No. 2

Record of Change No. <u>3</u>			
<u>Date:</u> <u>2/8/65</u> <u>Description of Change:</u> <u>Reason for Change:</u> <i>See Change No. 1</i> <i>See Change No. 1</i>			
Decreasing Logs	Increasing Logs	Early Starts	
none	12	10	
NIL ----- Total Change ----- Amount Already Processed ----- Amount Remaining ----- Magnitude and Sign of Change - This Sheet ----- <u>∞</u> <u>10</u> <u>6</u> <u>4</u> -----			
			-4

Figure 35C. Record of Change No. 3

network analyzer, the changed activity, No. 7, is energized, and the sequence lines having decreasing lags are listed. Note that for work with the analyzer, sequence lines are designated by single-number labels as obtained from the Sequence Line Data sheet of Figure 24. Having listed the decreasing-lag sequence lines, their current lag values are checked in the Sequence Line Data record to determine the NIL. An asterisk indicates the sequence line which determines this value. If the NIL is greater than zero, a change is possible. The analyzer is then used to determine the sequence lines with increasing lags and the activities whose early start times are affected. Before data in the Node Data and Sequence Line Data records may be updated, the magnitude of the change must be determined. This is the total change to be made less any portion of that change that has already been processed. If this figure is less than the NIL, all values are updated by this amount and processing is completed. If the NIL governs, as in the case shown in Figure 35A, all data is updated by the amount of the NIL and in the direction determined by the nature of the change.

The second cycle of this same change is shown as Change No. 2 in Figure 35B. Activity No. 7 is again energized on the analyzer, and the decreasing-lag sequence lines are listed. In this case the NIL is zero, so increasing-lag sequence lines and affected activities are not listed. Instead the network diagram is examined to determine which sequence line must be converted from a zero to a positive value. This change is made on the diagram, and the corresponding closed circuit in the analyzer is opened by throwing the appropriate toggle switch to the "off" position (or by taking whatever action is necessary depending on the design of the analyzer). Finally the third cycle is performed with the results indicated on Figure 35C. In this case the NIL is infinite since no decreasing-lag sequence lines result, and the remaining portion of the change can be completed.

CHAPTER VII

THE USE OF SUBNETWORKS

General

At the beginning of Chapter III three approaches were proposed for extending the range of practicality of non-computer methods. The last of these was to work with a section of the overall network rather than with the entire diagram. The last two chapters have suggested that the manual methods of Chapter V are practical for networks up to approximately 100 activities and that network analyzer methods of Chapter VI would extend this range to approximately 250 activities. These are not well-defined limits but, rather, depend upon the frequency of updating and the availability and effectiveness of alternative methods. However, there are possibilities for scheduling, updating, and replanning considerably larger networks if effective methods are developed for breaking these networks into smaller subnets. This approach has the further advantage that it is often better to expand a subnetwork in detail and to analyze it more closely than to work with an overall network that not only is larger but also covers a considerable span of time. Three types of subnetworks will be considered in this chapter.

Single Terminal Interfaces Only

The simplest type of subnetwork is one that is connected to the master network at only two points, one being at the beginning of the initial activity and the other at the end of the final activity of the subnetwork. Finding a group of activities within a larger network that meets these requirements and, therefore, can be treated separately as a subnetwork, is rather infrequent. A more common case is that of a single activity, or a small group of activities, that meets the interface requirements and is expanded into a larger number of more detailed activities. For example, a general contractor might represent the work

to be performed by a particular subcontractor as a single activity on the master diagram; however, it might be beneficial for the subcontractor to expand this single activity into an entire subnetwork in order to analyze the project at his level of operations.

This type of subnetwork can be treated as an independent network, except that it is subject to a variable starting date as determined by its tie to the master network. Insofar as the master network is concerned, the subnet can be represented by a single equivalent activity which functions in the same manner as any other individual activity. The critical path of the subnetwork will give the duration for the single equivalent activity. A time-cost curve for the subnet will give time-cost tradeoff data for the single equivalent activity. And, of course, the resource schedules developed for the subnet will establish the resource requirements for the single equivalent activity.

The use of subnetworks to expand single activities from the master network has two important advantages. First, this work is temporarily isolated and can receive concentrated attention. Here the network involved is small enough that non-computer methods are practical; also, this analysis is of the type that is generally performed by operational-level personnel in the field office. The second advantage is that the master network is kept free of minute details and, therefore, can better serve its purpose of permitting overall project planning and review. Sometimes, by keeping the master diagram free of excessive detail, it also will be of such a size that non-computer methods are practical for planning, scheduling, and updating it.

Intermediate Interfaces

Often a subnetwork has intermediate ties between its activities and other activities of the master network in addition to the terminal ties of the preceding case. An ability to handle such cases will extend the possibilities for network breakdown. A system that provides this ability has been developed for complex PERT networks.¹ The purpose

¹USAF PERT Cost System Description Manual, Preliminary Draft, Appendixes A and B, March, 1963, Secretariat, AFSC PERT Control Board, Hq. AFSC (SCCS), Washington 25, D.C.

of this system is to produce a master network called a "Program Management Network" which combines several detailed networks "for which the various contractors and contributing agencies are responsible."¹ The detailed networks have certain common interface events as well as terminal events and other important "milestone" events that it is desirable to include in the master network. However, it is not wise to include the bulk of the detail of the component networks in the master network.

The approach used in this "PERT" system is to first condense the detailed networks into the smallest possible network that will include all interface and milestone events. These are connected by artificial activities of such duration that the essential time relationships of the original network are preserved. The condensed networks are then combined into a single integrated network using the common interface events to tie the separate ones together. Routine time calculations are then performed on this integrated, master network to determine the dates of each event and the overall critical path. These event dates are then introduced back into the detailed networks as scheduled event times if they are later than those previously calculated. Finally, the detailed network is updated if it has been necessary to reschedule any of the events. Whenever changes occur in the detailed networks, the cycle should be repeated. The condensation of the individual networks is accomplished by an operational computer program developed by the Air Force Systems Command. The system is based on the use of the event-labeled arrow diagram historically characteristic of the PERT approach.

This system can be adapted to the non-computer, precedence diagramming approach being presented in this report. It is not essential that the detailed networks represent separate projects, but this, too, is a possibility for multi-project scheduling. Usually the detailed networks will represent portions of the overall project that might logically be considered separately, even though there are certain

¹ Ibid., Figure A.1, p. A.4.

interface events within their networks that require ties to other portions of the project. For example, a heavy construction project involving a dam, powerhouse, spillway, and diversion tunnel might lend itself to this treatment. A building project involving several different buildings or several wings of the same building, or several floors of a multi-story building also would suggest uses of this approach. It is not even essential that the various detailed networks represent divisions of the overall project that must be logical from the standpoint of subject matter groupings. It is conceivable, for example, to start with a master, detailed network that is too large to handle practically by non-computer methods and to divide it into several subnetworks by a subdivision arrangement that minimizes the number of "cut" sequence lines. Whenever it is necessary to cut a sequence line, an interface event must be inserted to represent the common tie between the resulting subnetworks. These are then condensed and the integrated master network constructed. This integrated master network simply becomes a condensed version of the original detailed master network. If it has been possible to subdivide the original master network without the introduction of an excessive number of interface events, the integrated network will be small enough to handle by non-computer methods. Whenever changes are made or occur unavoidably during field performance, these changes are processed through the detailed subnetworks to the condensed subnetworks and then to the integrated master network. The results are then introduced back into the detailed subnetworks which are updated as necessary. Finally the results may be transferred to the detailed master network if it proves desirable to maintain this network. By this approach a detailed project network too large to handle directly by non-computer methods can still be analyzed, replanned, and updated in this manner. It is probable that if certain important milestone events are included in the condensed subnetworks, the integrated project network will make the detailed master network unnecessary. This decision is, of course, optional.

The chief problem in adapting this USAF PERT system to non-computer methods is to develop procedures for condensing networks that are not dependent on a computer program. Figure 36 shows a detailed subnetwork

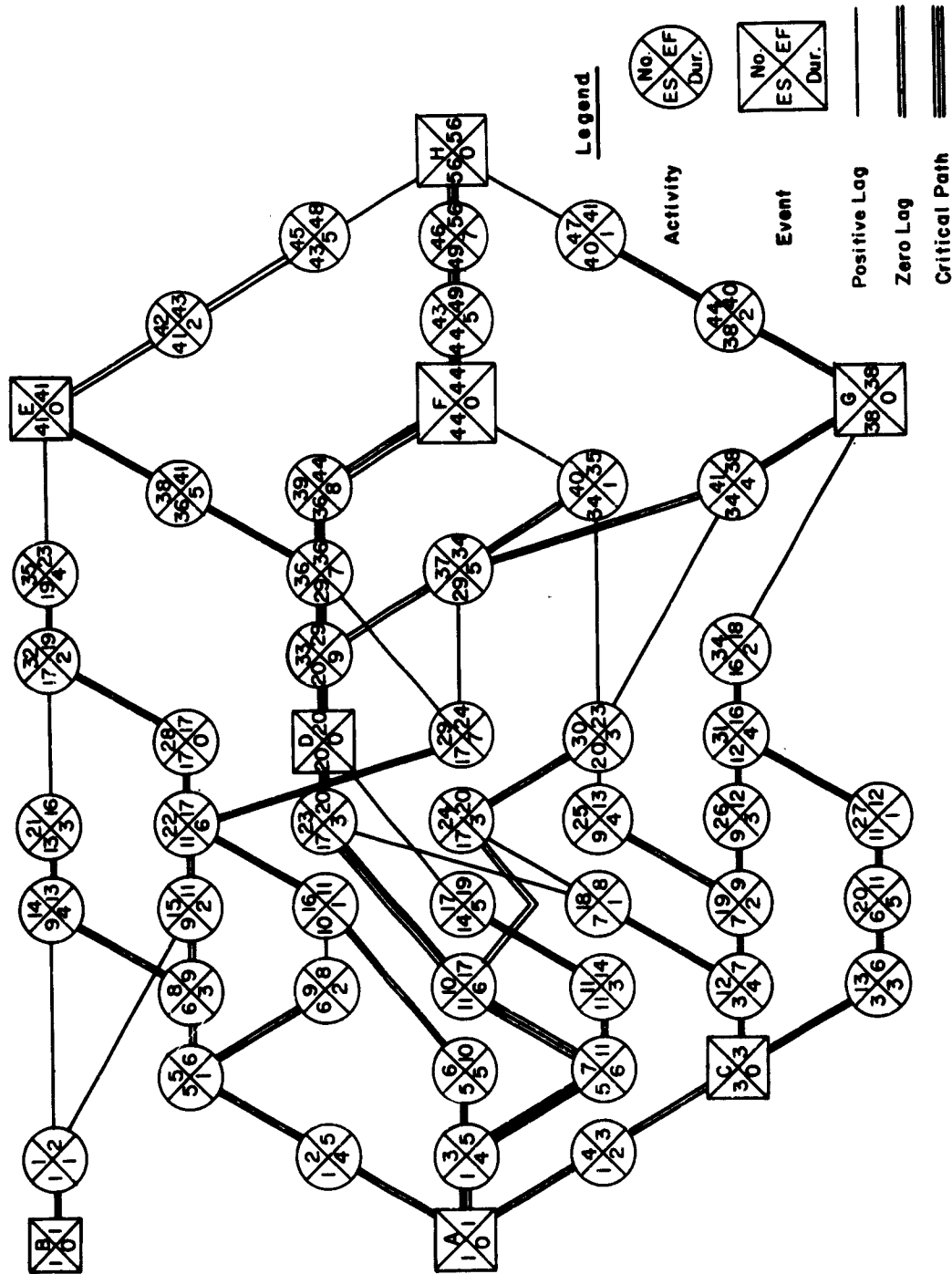


Figure 36. Detailed Subnet with Interface and Milestone Events

that includes 47 activities and 8 interface and milestone events. Events A and H are the start and finish, respectively, of the overall project, and are therefore terminal milestone events for this subnet. Events D and F are internal events of this subnet which are of sufficient importance to be included as milestone events. B, C, E, and G are interface events having ties with other subnets. This detailed network is the precedence diagram equivalent to the PERT network shown in Figure B.1 of the AFSC manual previously cited. It may be reduced to the condensed network shown in Figure 37. The activities shown on this diagram are artificial ones whose durations represent that of the critical path between the two corresponding events on the detailed diagram. A method for determining when these equivalent condensed activities are needed for finding their durations will be briefly outlined.

First, the final network event, Event H, is placed on the right end of the sheet on which the condensed network is to be developed. Working backwards to the left, Event G is added next. The fact that G and H should be linked by an artificial connecting activity can most

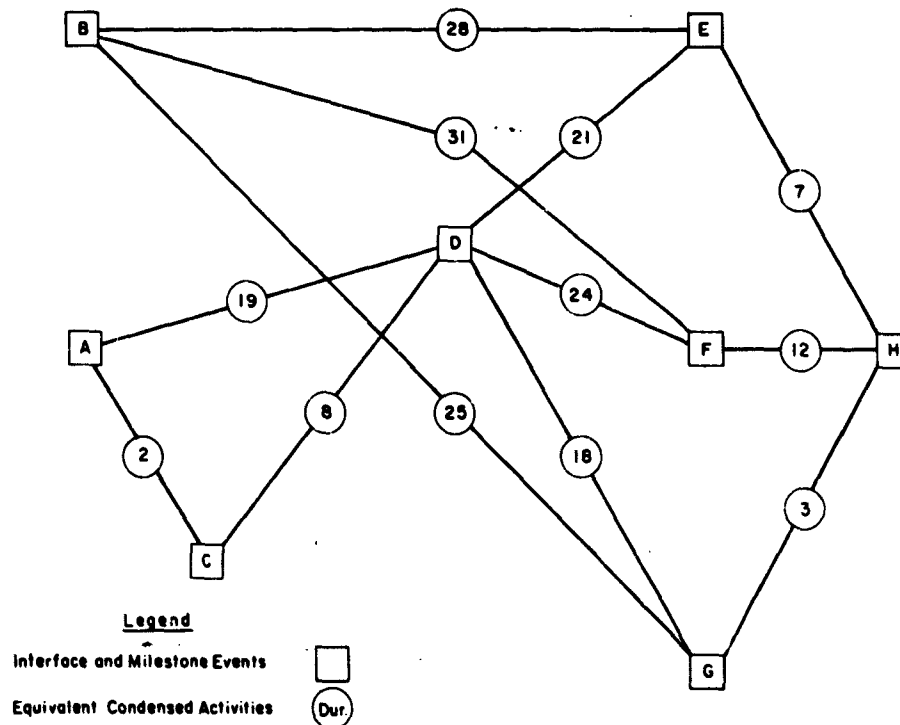


Figure 37. Condensed Subnetwork

easily be determined by observing the detailed diagram and noting that there is a path from Event G to Event H. The duration of the connecting activity must quite obviously be three days since this is the sum of the durations of the activities on the only existing path between the two events on the detailed diagram. However, in a more complex network, this duration may not be so obvious. A more general approach is to temporarily reschedule Event G to a later date that will cause a zero-lag path to be developed between it and Event H. The difference in the event dates at this point will determine the duration of the connecting activity on the condensed diagram. The methods of Chapter V may be employed for this rescheduling, but these procedures are accomplished most effectively by the use of the network analyzer described in Chapter VI. In this case, rescheduling Event G involves a NIL of 15 days as determined by sequence line 47-H. If Event G is scheduled for the 53rd day (38 plus 15), a zero-lag path is established to Event H. The critical path between the two events is 3 days (56 minus 53) and this becomes the duration of the connecting activity on the condensed diagram. The rescheduling of Event G is only temporary for the purpose of determining the duration of the equivalent condensed activity. Its start date on the detailed diagram is then returned to the original date.

Next Event F is placed on the sheet. An examination of the detailed diagram indicates a path between Events F and H and no path between Events F and G. The path between F and H is already a zero-lag path. Therefore the difference in the event times of 12 days (56 minus 44) gives the duration of the connecting activity directly. Event E now is placed on the sheet. There is a path from E to H and none from E to F or G. The NIL for rescheduling Event E to a later date is 8 days as determined by sequence line 45-H. If Event E is temporarily rescheduled to the 49th day (41 plus 8), a zero-lag path is created to Event H. The duration of the critical path between the two events is 7 days (56 minus 49) and this is entered as the duration of the connecting activity on the condensed diagram.

Proceeding further to the left Activity D is added. There are paths between Event D and all of the later events. Moreover, there

are zero-lag paths, so it is not necessary to reschedule Event D. The connecting activities between Event D and Events E, F, and G are respectively 21, 24, and 18 days as determined by the difference in event dates. In this case, the only zero-lag path between D and H passes through F. If there were an alternate zero-lag path, a connecting activity with a duration of 36 days (56 minus 20) should be inserted directly between Events D and H. But here it is unnecessary. Next Event C is placed. There are paths between it and all of the following events, but none are zero-lag paths. Rescheduling Event C involves a NIL of 7 days as determined by Sequence Line 25-30. However, this still does not produce any zero-lag paths. Further rescheduling involves a NIL of 2 days as determined by both Sequence Lines 18-23 and 18-24. This introduces a zero-lag path between Events C and D and thence to all the other events. Event C has been temporarily rescheduled to the 12th day, and the duration of the connecting activity to Event D is 8 days (20 minus 12). Since there are no zero-lag paths to other events besides the one going through Event D, no direct connections to any other is needed. Note that there are other paths between Events C and G, for example, than the one through Event D. Subsequent changes in project performance require that the condensation procedure be repeated, since it is possible that a network having different connecting activities and sequence lines than those originally developed may evolve.

Event B is now added, noting that there are no possible paths to Events C or D. There are to the remainder of the events. The initial NIL for Event B is 7 days, determined by Sequence Lines 1-9 and 1-15, but it does not provide any zero-lag paths. A further rescheduling of Event B has a NIL of 5 days determined by Sequence Lines 29-36 and 29-37. This creates zero-lag paths to Events E, F, and G. The critical path durations are, respectively, 28 days (41 minus 13), 31 days (44 minus 13), and 25 days (38 minus 13). A zero-lag path is also created to Event H, but since it must pass through Event F a direct connection is not required. Finally Event A is added. It requires no rescheduling since there are zero-lag paths to all other

events. Note here that a connection is required between Events A and D even though there are connections between A and C and between C and D.

The condensed network of this example has 12 activities in place of the 47 activities of the original detailed network. Both have the 8 milestone and interface events. Therefore the total reduction achieved is a decrease from 55 nodes (activities and events) to 20. As network size increases, the ratio of reduction will generally improve since there will usually be a lower proportion of event nodes and of direct connections between events.

The condensed network of the detailed subnet is combined with other condensed networks representing the remaining subnets to form an integrated master network. Figure 38 shows an integrated network composed of three condensed subnets. It is a precedence diagramming presentation of the integrated PERT network shown as Figure B.2 of the AFSC manual.¹ Routine time calculations have been performed on Figure 38 to determine event times for the master network. These event times, if later than the corresponding event times originally calculated for the detailed subnets, are introduced back into the detailed subnets as scheduled times. Figure 39 shows the revised subnet. Note that the critical path for the master network (and, therefore, for the entire project) passes through the subnet but is entirely different than the critical path developed for the subnet when the scheduled times of the interface events were not considered. Note also that if the network of Figure 39 were condensed, the results would be exactly the same as shown in Figure 37. If calculations are performed correctly, there should be no "looping."

This approach has potential value where subnets can be established that do not have too many interfaces. It permits the subnets to be expanded in detail and at the same time preserves a master control network that is not encumbered with this detail. Skillfully used, it permits complex projects to be reduced to network sizes that can be handled by non-computer methods. Limited experience with these

¹ Ibid., p. B.3.

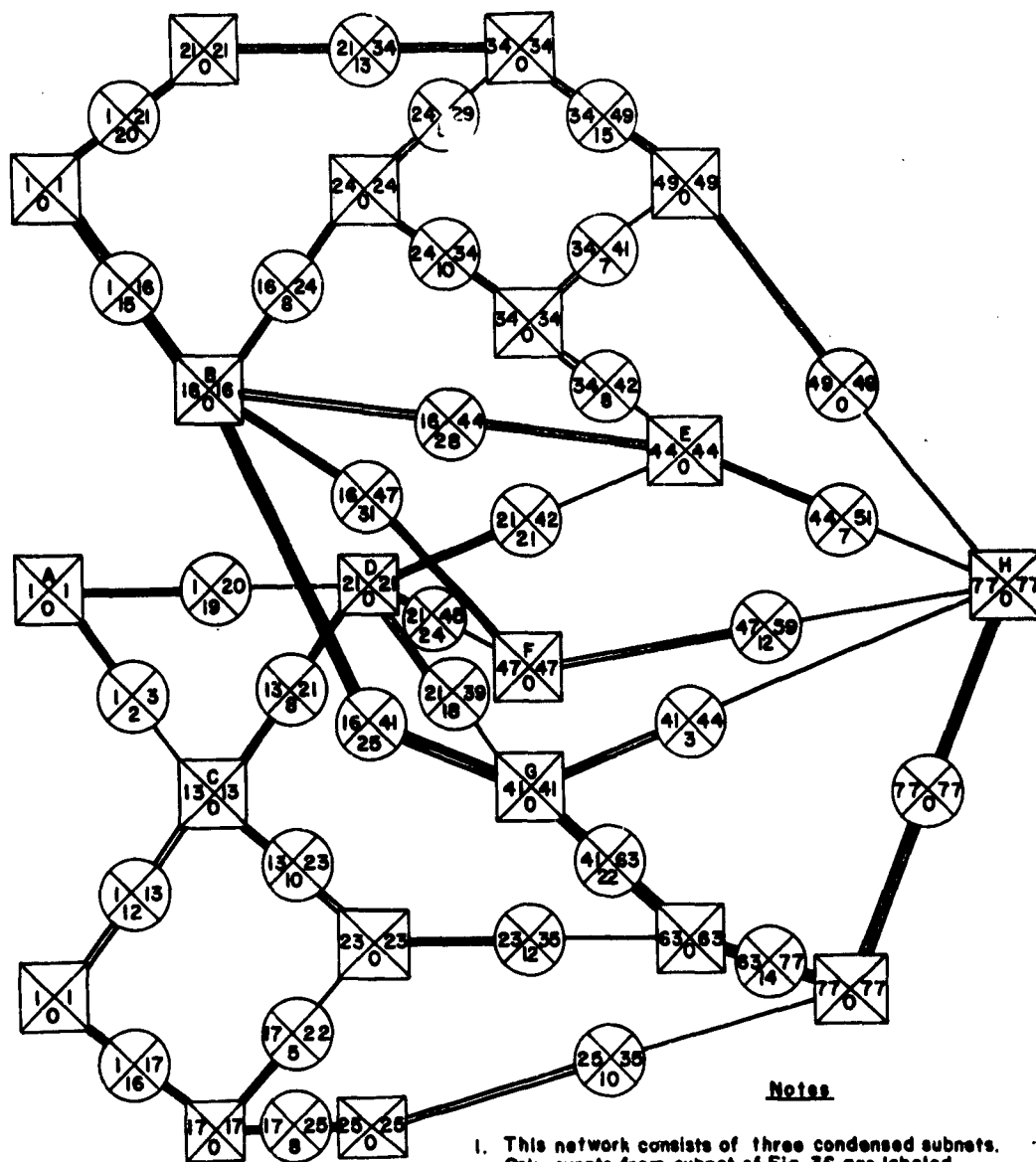


Figure 38. Integrated Master Network

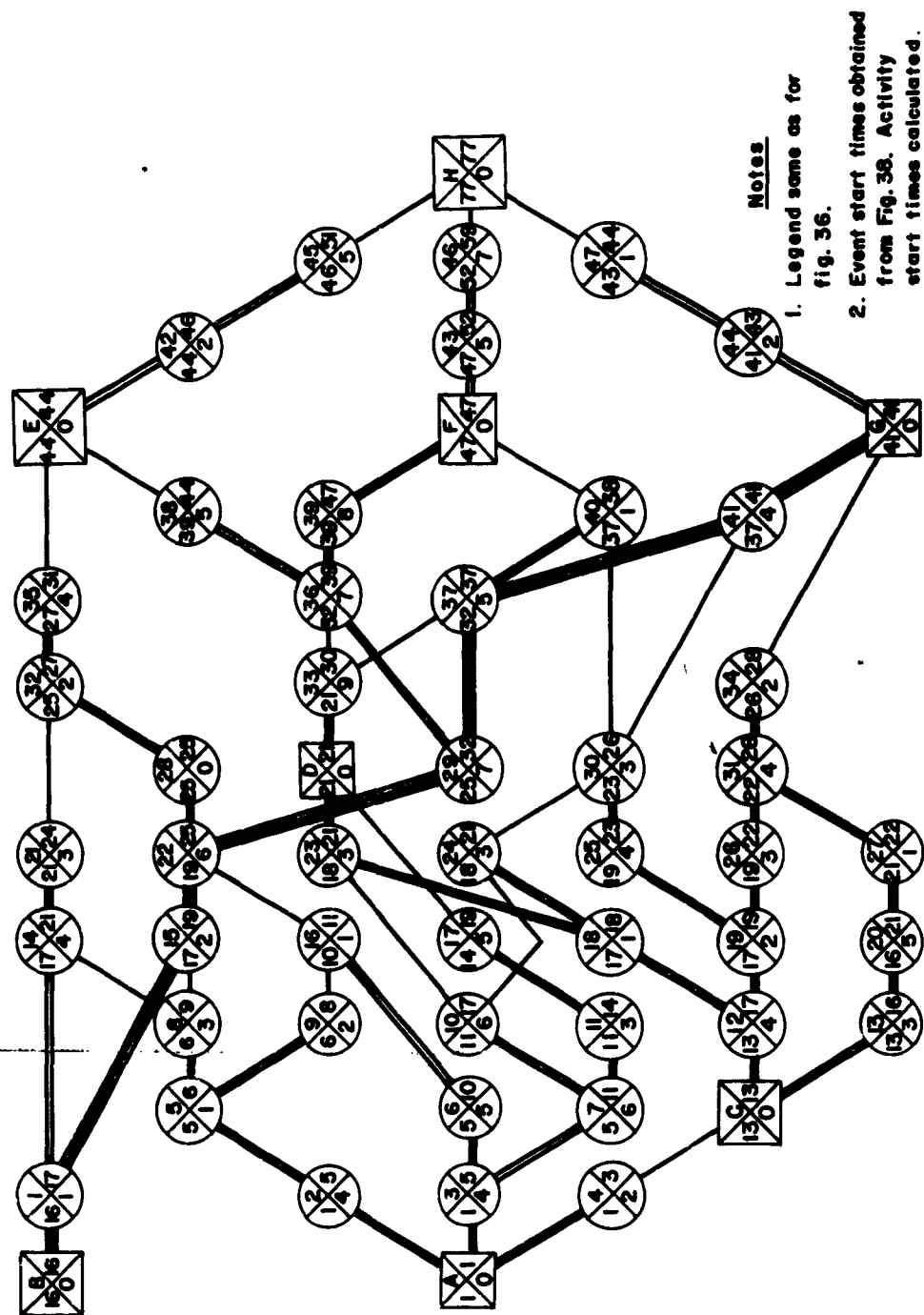


Figure 39. Updated Detailed Subnetwork

procedures indicates that the network analyzer described in Chapter VI is very helpful in the network condensing phase.

An unsuccessful attempt was made to subdivide a large network for a building job into subnets representing the activities of the various sub-trades. The intent was to produce subnets that each subcontractor could develop in detail around certain interface events that introduce external control or affect others. This subdivision was not efficient because there were too many interfaces to achieve an appreciable reduction in network size. For example, the work of the mechanical and electrical subcontractors was so closely tied to that of the general contractor or other subs that many interfaces were present. On the other hand, subdivision by building area or floor level indicated much better opportunities for simplification by subnets. Certain projects with naturally defined groupings of activities and having relatively few ties with other groupings, lend themselves very nicely to this form of network breakdown.

Dateline Cutoffs - General

A third type of network subdivision will be proposed here. The author believes that it will have more general application than the two systems already described. It is ideally suited for the combined application of non-computer and computer processing, where the master network is updated periodically on the computer but the detailed, job-level, day-by-day planning and updating is accomplished using subnets and non-computer procedures. However, it also makes a strictly non-computer approach possible for many networks that would otherwise be too large for such an approach.

In the application of this system, an overall master network representing the entire project is prepared first. Having estimated activity durations, routine time calculations are made, including late start and finish data. Since a large network is involved, a computer is advantageous in making these calculations. If a computer is not available, it is still usually practical to make these basic time calculations by manual methods since frequent recomputation is not to

be required. Once the scheduling of the individual activities has been calculated and the project duration and critical path are known, some replanning involving the entire network may be in order. If the completion date is not satisfactory, it may be necessary to change the job plan or to make certain time-cost tradeoffs that can be developed without an extremely detailed analysis. If major resource availability problems are apparent, some gross replanning to eliminate excessive requirements should also be attempted. When computer facilities and programs are available for time-cost tradeoffs and resource allocation, they can be used at this stage. When they are not, overall project planning will usually depend on good judgment and skillful analysis supplemented by any of the applicable non-computer procedures already presented in this report. However, the network breakdown system to be described below can also be used in replanning the overall network. This application will be outlined later.

Once a reasonably satisfactory general plan and schedule for the overall job has been developed, there is serious question that detailed planning and scheduling based on the entire network diagram will be productive. Day-by-day updating of scheduling of future activities and detailed rescheduling of activities to smooth out resource requirements (as well as to eliminate major peaks and dips in them) are examples of procedures that are important in reducing construction costs. But there is little justification for the daily updating of a large number of activities that are to be performed in the distant future or for detailed rescheduling of these activities to smooth out resource requirements many weeks or months ahead. The results obtained are not significant since many other changes which will affect them will almost surely occur in the intervening period. Attempts to work with the entire network to accomplish this detailed planning and frequent updating either lead to a considerable amount of wasted effort or, more frequently, a complete failure because the work on the network is discontinued as being too time consuming.

This suggests, as a possible solution, a breakdown on a calendar basis in order that the job plan for the portion scheduled for the

immediate future can be considered separately. It may then be analyzed and replanned frequently and in detail, and expanded if it seems desirable to do so. All of these steps would be practical, even by non-computer methods, since the subnetwork only would be a fraction of the size of the overall network. However, this subnetwork must retain the important relationships present in the overall network. Specifically, it must be tied to the event representing final project completion in order (1) that any changes in the subnet will be correctly reflected in the project duration, and (2) that the critical activities in the subnet will be recognized at all times.

The initial breakdown is into two parts. One presents the job plan for the period from the beginning up to some dateline cutoff. The second part is the remainder of the overall network from the cutoff interface to the end. The dateline cutoff may be chosen to make the first part, the working subnetwork, span a given calendar period that is realistic for advance planning efforts, such as the next one or two months. Or it may be chosen to make the size of the subnetwork compatible with the updating method. For example, if manual procedures are to be employed, it may be desirable to confine the size of the subnetwork to approximately 100 activities, and the dateline cutoff should be chosen accordingly. As time passes, additional subdivisions are made. At each division a new subnet of the same chosen time span, or of some limiting size, is formed. Therefore the new remainder network becomes progressively smaller with each subdivision. As an example, consider a job that commences on February 1 and is scheduled for completion on November 15. After an overall network diagram has been constructed, basic time calculations performed, and any gross replanning to make the job plan feasible accomplished, a subnet including all activities from project commencement to March 1 is constructed. This subnet is used for detailed replanning of the work in the first monthly period and may be expanded to show considerably more detail than was desirable on the master diagram. It is kept updated on a daily basis in the job office. The remainder of the original overall network, covering the project plan from March 1

to November 15, is available for updating whenever it is necessary to do so. It may become desirable to analyze it in certain replanning situations. However the bulk of work will be performed by non-computer procedures on the February subnetwork, and the remainder network will be updated at the end of the month, possibly by computer. After the first month's subnet has been analyzed, expanded, and replanned, and before the end of February, a second subnet representing the month of March is constructed. This subnet is updated and detailed planning of it commences. The remainder network now extends from April 1 to November 15. It is generally desirable to keep one subnet ahead of the current one. Therefore under this system there will usually be three subnets in existence at any one time. One will represent the current period; one will represent the next period ahead; and the third will cover the remainder of the project. Of course, it is possible to introduce as many intermediate subnets as the planner desired. However, this approach is based on the principle that it does not pay to make detailed analyses too far into the future. So the three subnets described above generally satisfy the maximum breakdown requirements. By the time the project has progressed into March, the February subnet has been discarded (or filed), the March subnet is in current use, the April subnet is being developed, and the remainder subnet from May 1 to November 15 is available if needed. This process continues until the remainder subnet finally becomes the current subnet and carries the job through to completion.

Although the primary objective of this system is to permit detailed planning of a limited portion of the overall network, it was stated earlier in this section that it might also be used in the gross replanning that is performed initially on the overall network. If this has been constructed and the basic time calculations performed, it may then be broken into a number of subnets using dateline cutoffs. In this case the number of subnets is determined by the limiting size for the method of analysis to be used. If the overall network contained 600 activities and a network analyzer having a 250 activity capacity is available for replanning and updating, the master network

would be broken into three subnets each having approximately 200 activities. Any change that was proposed would be made in the subnet containing the activities involved. Updating of this subnet would convey the effects of the change to the interface activities between it and the following subnet. These changes would be transferred to the corresponding incoming interface activities of the next subnet, and that subnet would be updated. This process would be repeated until the change was conveyed to the final project activity. This is a fairly cumbersome method, but it permits a limited amount of replanning by non-computer procedures on networks that are beyond the normal size limit for them. Actually, as much work is performed as would be required if calculations were done directly on the master network. Instead, the work is accomplished in several steps, each involving a smaller network that is less formidable. More important, however, is the fact that a tool such as the network analyzer can be utilized even though the size of the overall network far exceeds its capacity.

Dateline Cutoffs - Procedures

To demonstrate the procedures for this type of network subdivision, the network of Figure 40 will be used. The project commences on June 18 and is scheduled for completion on September 28 (the figure indicates September 29 which must be interpreted as the end of the preceding day). The size of this network is much too small, of course, to require any network breakdown, but the principle can be illustrated just as well on a simple network. Although not essential, the diagram has been plotted to a time scale in order that vertical datelines will serve as cutoff lines. Any initial replanning of the overall project has been performed and the results of the basic time calculations, updated to reflect this replanning, are shown on the diagram. For example, Activities 10 and 25 have been assigned scheduled starting dates later than their earliest start times in order to satisfy resource restraints. Note that late start and finish times are calculated and included in the figure because they will be needed.

Since the job begins in mid-June and not many activities are

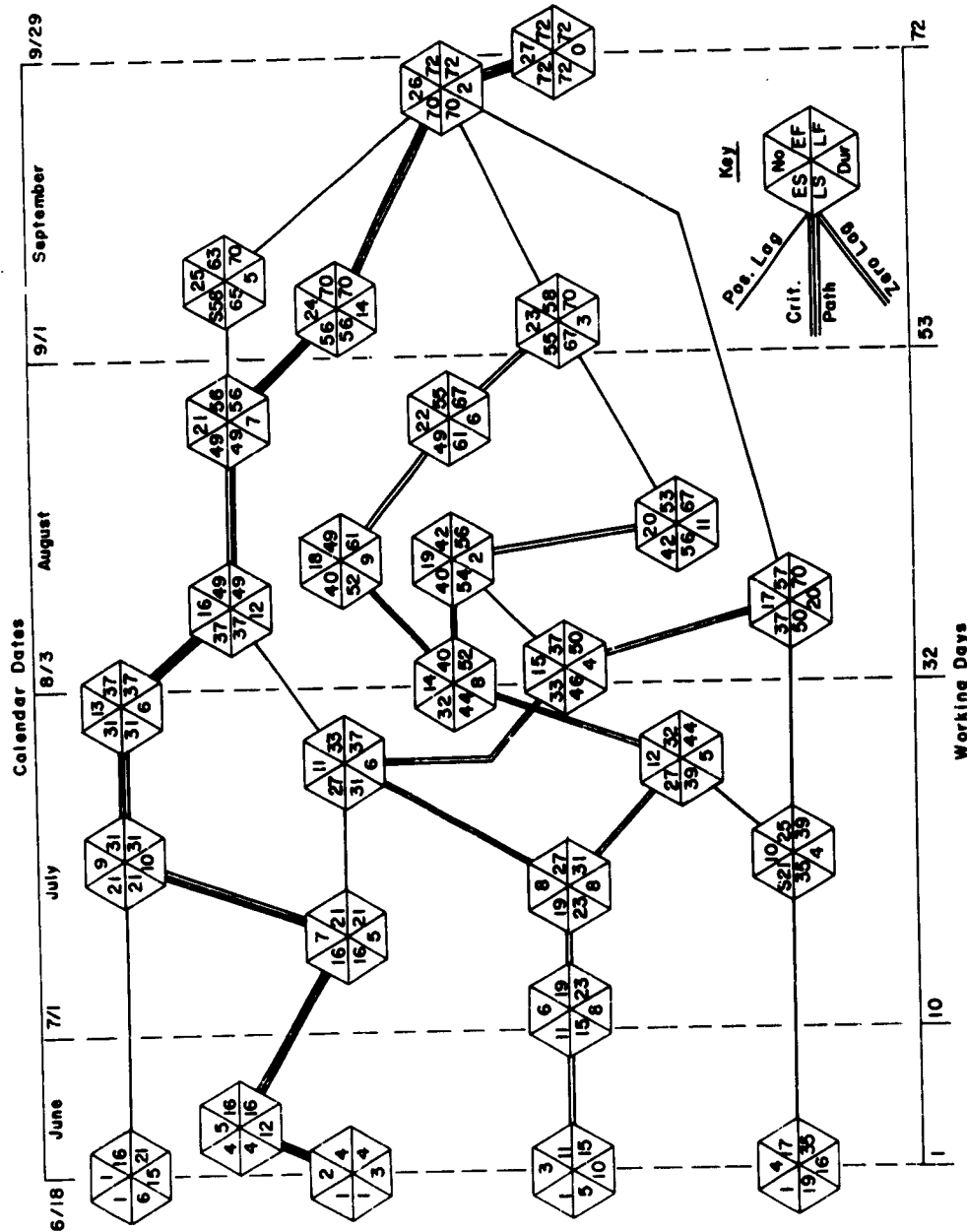


Figure 40. Master Project Network

included in the time span through July 1, the first cutoff date is selected as August 1. This date corresponds to the 32nd working day-- of the job. Subnet A, shown on Figure 41, covers the time span from the beginning of the job through the month of July and provides the first subnet for detailed replanning and day-by-day updating. Artificial tie nodes have been inserted between each interface immediately to the left of the cutoff line and the event representing project completion. When an activity begins on the cutoff dateline it should be included also. Therefore Activity 14, which begins on the 32nd working day, is shown on Subnet A.

The next step is to calculate the durations for the tie nodes. Each duration is set equal to the difference between the early start time of the final project activity (or the time of the event representing project completion) and the latest finish time (shown on Figure 40) of the preceding interface activity. Once determined, the durations of these tie nodes are not changed unless duration or planning changes are made in the portion of the network to the right of the cutoff line. If such changes are made in the remainder network or on a later subnet, then late start and finish data must be updated and tie node durations recalculated. These tie nodes will serve to convey the effects of all changes in Subnet A to the final project event and also to indicate the formation of new critical paths, or the conversion of existing critical paths to non-critical ones. This means that the project completion date is updated with every change made in the subnet and that the critical activities within this subnet are always known. Whenever a non-critical tie node becomes critical, the new critical activities in the subnet may be determined by tracing out the zero-lag sequence line path from that tie node back to the left end of the diagram. This, of course, does not reveal the identity of the newly-formed critical activities within the remainder network. In order to determine these, the remainder network must be updated. It is also possible for changes in the subnet to produce new critical activities in the remainder network even without a non-critical tie node becoming critical. This would occur when a critical branch

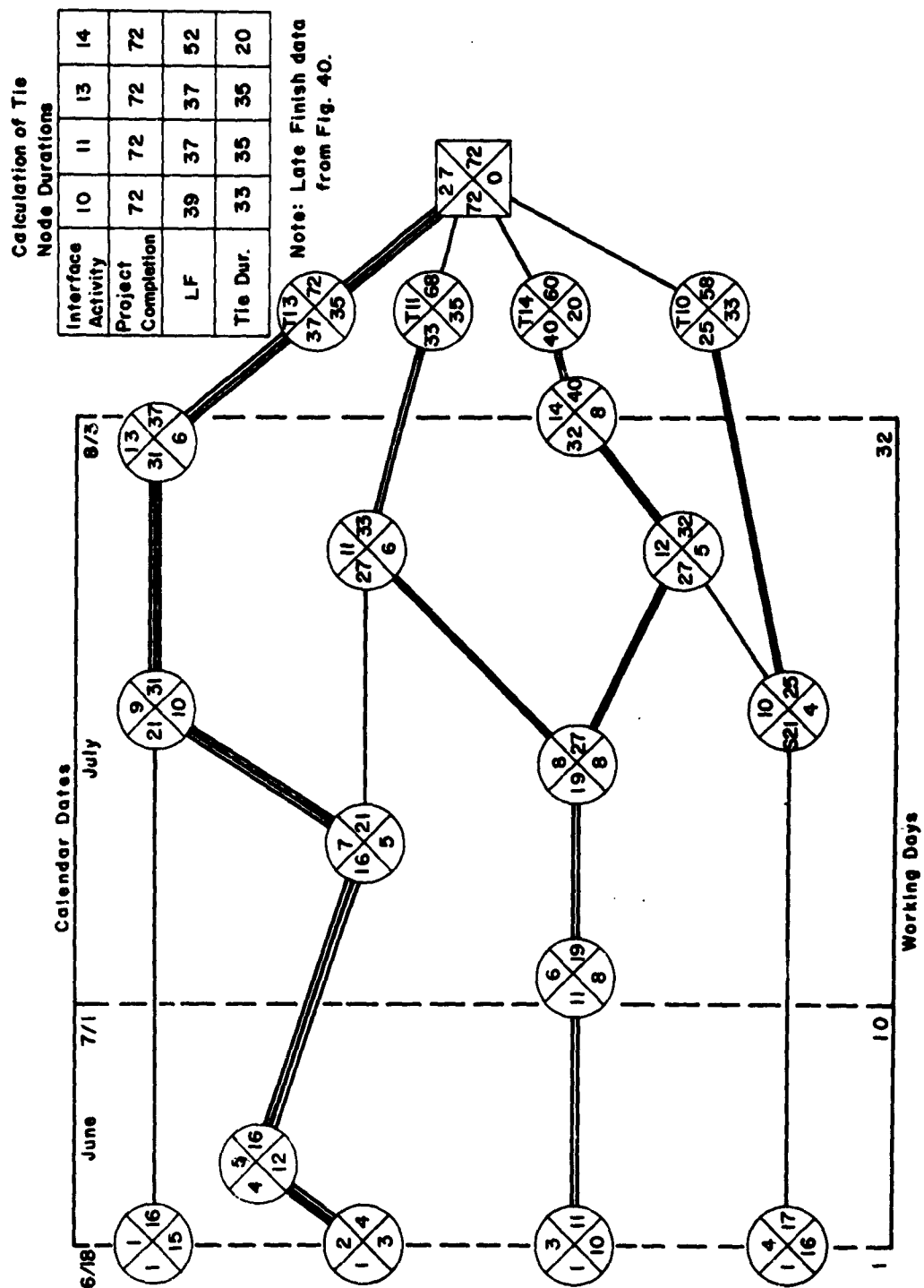


Figure 41. Subnet A with Interface Tie Nodes

path were formed between the existing critical path and the final project event. This would only be revealed when the remainder network was updated. Therefore, it must be remembered that the tie nodes only assure knowledge of the critical activities within the subnet. They neither identify nor guarantee an indication of new critical activities in the remainder network. However, this is not a serious shortcoming. As long as project management can obtain an updated project completion date and a complete set of updated scheduling data, including knowledge of all critical activities, for the time span covered by the current subnet (and, generally, one additional subnet ahead), the job can be effectively controlled and managed. If updating of activities in the more distant future is necessary, the remainder network may be updated when desired or at established periodic intervals.

One reason for updating the entire project network, either periodically or after several changes have occurred, is to pinpoint any major resource problems well in advance. It was assumed that at the beginning of the project some overall planning had been performed to eliminate prohibitive resource requirements at any time during the span of the entire job. It is desirable to review this overall planning occasionally. Another reason for updating the entire project is to permit time-cost tradeoffs based on a knowledge of all critical activities rather than only those within the subnets. When a second parallel critical path is formed within Subnet A, for example, there may be a favorable opportunity to expedite a critical activity on one of these paths but not on the other within the time span of the subnet. If the entire job were updated and examined, there might be a chance to expedite one of the activities on the portion of the second path that is located to the right of the subnet cutoff dateline. Or the two paths might merge to the right of the dateline cutoff, and a single common critical activity might be expedited more economically to reduce project duration. These opportunities would be missed if all planning and updating were confined to the subnet. This shortcoming of the system of subnets for time-cost tradeoff analysis is

probably its weakest point. It may be minimized by making major time-cost tradeoffs prior to network breakdown and by periodically updating and reviewing the entire network for new time-cost tradeoff possibilities. But even without these precautions, an economical time-cost tradeoff involving two or more activities within the current subnet will very often prove wiser than one involving an activity in the more distant future even though the latter produces a somewhat more favorable cost. The old adage that "A bird in hand is worth two in the bush" is applicable here; if the opportunity to economically expedite an activity in the current period is passed up in favor of expediting a later activity, the opportunity is lost forever as far as the current activity is concerned and, due to intervening scheduling changes, may never materialize in the case of the later activity. However, if the earlier activity is expedited and it turns out that there is an opportunity to expedite a later activity on the same critical path more economically, it is often possible to find a corresponding time-cost "sell-back" that will still permit receiving the benefits. Such sell-back is possible when there is a critical activity on that path that can be accomplished more economically if more time is available for it.

There is an alternate approach to the use of tie nodes in the subnet. Referring again to Subnet A as shown in Figure 41, a tie node has been introduced following every activity in the subnet whose departing sequence line (or lines) intersect the dateline cutoff. While only four tie nodes were added in this simple example, a large number might be required in a complex network. Since these tie nodes are treated as activities, there is a corresponding increase in the number that must be handled in updating the subnet. If the network analyzer described in Chapter VI were to be used, for example, each of the tie nodes would have to be wired on the patchboard and would use up a portion of the capacity of the unit. If the capacity is already taxed by the size of the subnet, this creates a problem. Figure 42 shows an alternative presentation of Subnet A where the activities just to the left of the cutoff dateline, or on it, have

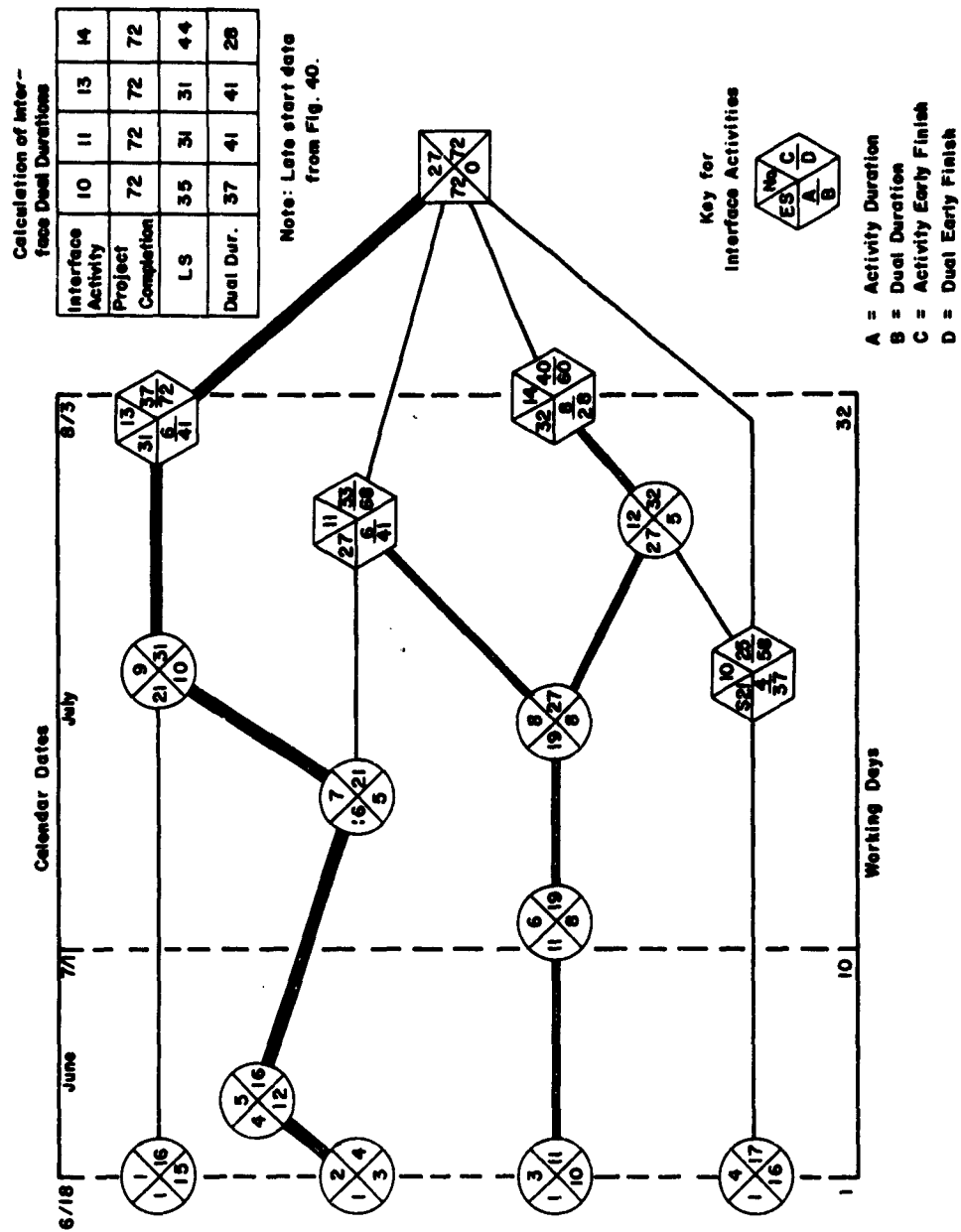


Figure 42. Subnet A with Dual Value Interface Data

dual data entries instead of following tie nodes. These dual entries are for the activity duration and early (or scheduled) finish time. The numerators represent, respectively, the actual activity duration and the corresponding early finish time calculated by adding this duration to the early start time. This early finish time is needed for two purposes. First, it provides data for the initial interface activities of the next subnet or for the remainder network. Second, it is required for internal updating of the current subnet when an activity, whose departing sequence line is intersected by the date-line cutoff, has another departing sequence line that extends to an activity within the subnet. An example of such a case is presented by Activity 10 in Figure 42. The denominators replace the tie node data. The duration denominator is calculated by subtracting the activity's latest start time (not latest finish time, as in the case of the tie node) from the earliest start time of the final project activity (or from the event time of project completion). The early finish denominator, then is computed by adding this duration denominator to the early start time of the activity. These denominator figures are used to update the final project activity or event. When the lag of a sequence line between a subnet interface activity and the final project event, calculated from this denominator value, changes from positive to zero or from zero to positive, a new critical path or the deletion of an existing critical path, respectively, is indicated.

The dual data system requires little modification in either the manual or the network analyzer procedures for updating the networks. However, care must be exercised to use the appropriate dual value for the purpose required. Otherwise procedures, including record keeping on separate tabulations when desired, are the same. The dateline cutoff system for network breakdown, whether accomplished with tie nodes or with dual data for interface activities, is based on making changes in the current subnet and keeping this subnet up to date. Subsequent subnets or the remainder network may also be updated from the current subnet data. However, if other changes are made in later subnets or

in the remainder network, these changes may alter the durations of the tie nodes or the dual durations of the interface activities of earlier networks by affecting the late start or finish data from which these durations are calculated. Therefore, whenever a change is made in a later network, late time data for the entire network should be updated and tie node durations or interface activity dual durations recalculated.

The subnets to be used for detail analysis and day-by-day updating will probably be expanded. The techniques described in Chapter V for adding new activities and for changing activity durations are useful for this expansion. An expanded and updated Subnet A is shown on Figure 43. New activities have been given alpha-numeric labels to indicate their relationship to the activities shown in Figure 42. While this labeling is satisfactory for manual methods, numeric labels are more convenient if the network analyzer is to be employed. They also are necessary if supplementary computer processing is involved.

Only a few added activities are shown in Figure 43 to illustrate some of the common types of changes involved in expanding a subnet. For example, Activities 3 and 9 have been broken down to show more detail than was desirable on the overall network. Activities Nos. 6 and 8 have been broken down in order that they might be overlapped with one another and with Activity No. 3, thereby reducing the completion date of Activity No. 8 by seven working days. Critical Activity No. 5 was reduced in duration by three days by recognizing that there was two days of preparatory work that was independent of the completion of Activity No. 2 and another day of finish work that could be performed after Activity No. 7 had begun. In a similar manner the duration of critical Activity No. 9 was reduced by two days by recognition that certain preparatory work could be performed after a portion of Activity No. 1 was completed. These latter two changes reduce project duration by a total of five days. Activity No. 10 had originally been scheduled at a date later than its early start time because of resource restrictions. The scarce resource is released upon completion of Activity No. 7. A more detailed study reveals that Activity No. 7 should

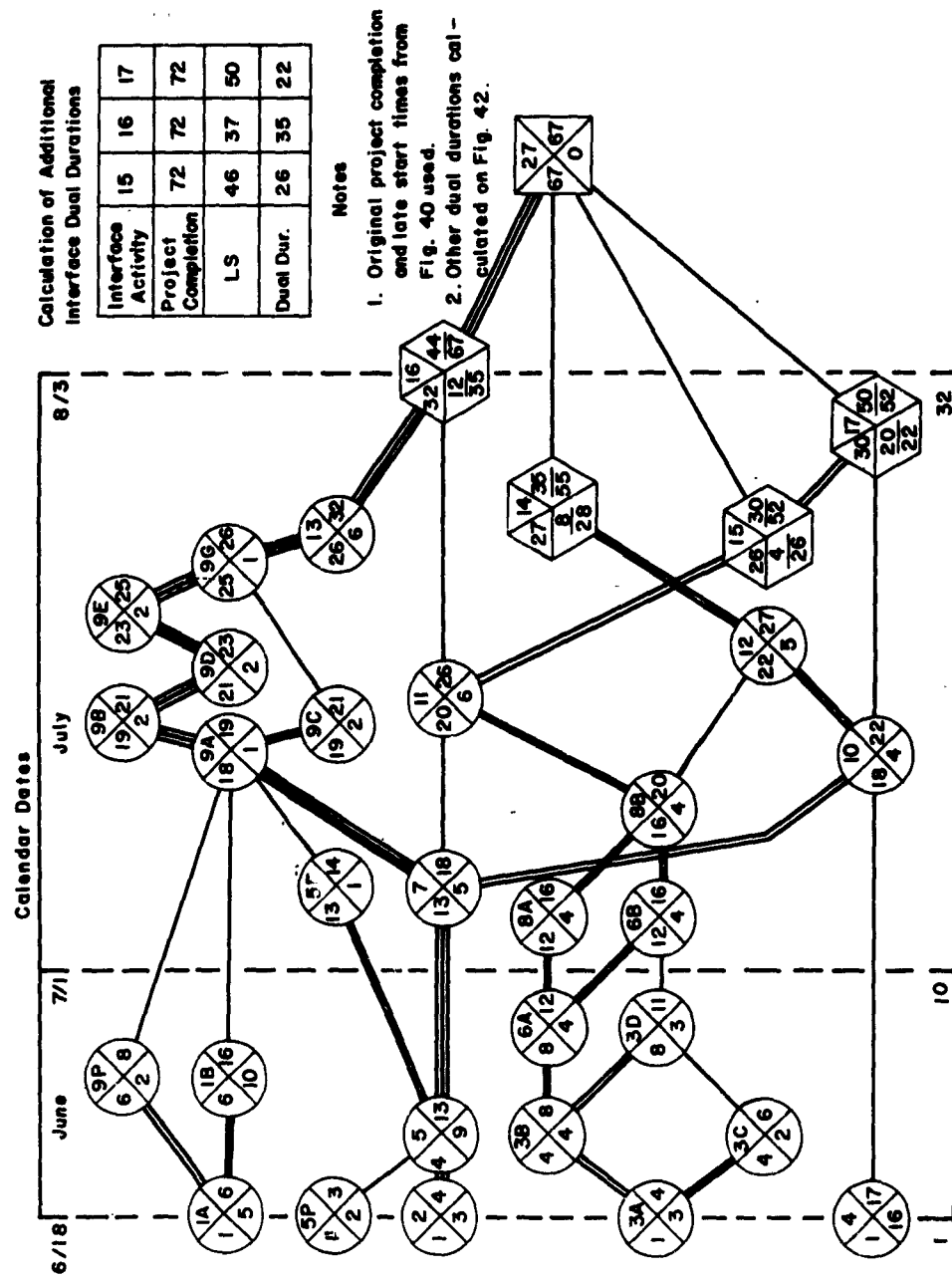


Figure 43. Subnet A Expanded and Updated

definitely be performed first and that there is no choice but to postpone Activity No. 10 until it is completed. Therefore a sequence line has been added between these two activities. Due to the changes in scheduling that is produced, Activities Nos. 15, 16, and 17 have been added to the diagram, since they may now be started either before or on the 32nd working day. Note that the calculations for the dual duration of these activities are based on the original project duration and late start data as shown on Figure 40. This is permissible even though the changes made in Subnet A have caused project completion and the late start times of the interface activities to change. As long as no changes have been made in the network between the interface activities and the final activity, late start times and project completion time shift together. Therefore, the same results are obtained for the dual durations whether original data or updated data is used. Although the modifications made in expanding Subnet A affect project duration, they do not produce a new critical path within the subnet.

To illustrate the introduction of a new critical path, suppose that after the project was started a procurement delay caused Activity No. 4 to take fourteen working days longer. Figure 44 shows a routine updating of Subnet A. The change in project duration resulting from this delay, and the new critical activities are shown, and the previously critical activities are now indicated as non-critical. A critical path to the right of interface Activity No. 14 is known to exist, but updating of the remainder network would be required in order to determine the specific activities within it that have become critical.

Activities Nos. 12, 14, and 17 have been shifted to starting dates later than the 32nd working day cutoff dateline for Subnet A. They could be deleted from the subnet, in which case Activities Nos. 8B and 10 would become interface nodes. However, once work has commenced on the portion of the project covered by a subnet, it is generally preferable to retain such activities and to revise the data for them rather than to delete them and convert other activities to interface ones. It is also preferable to drop the time scales rather than to shift

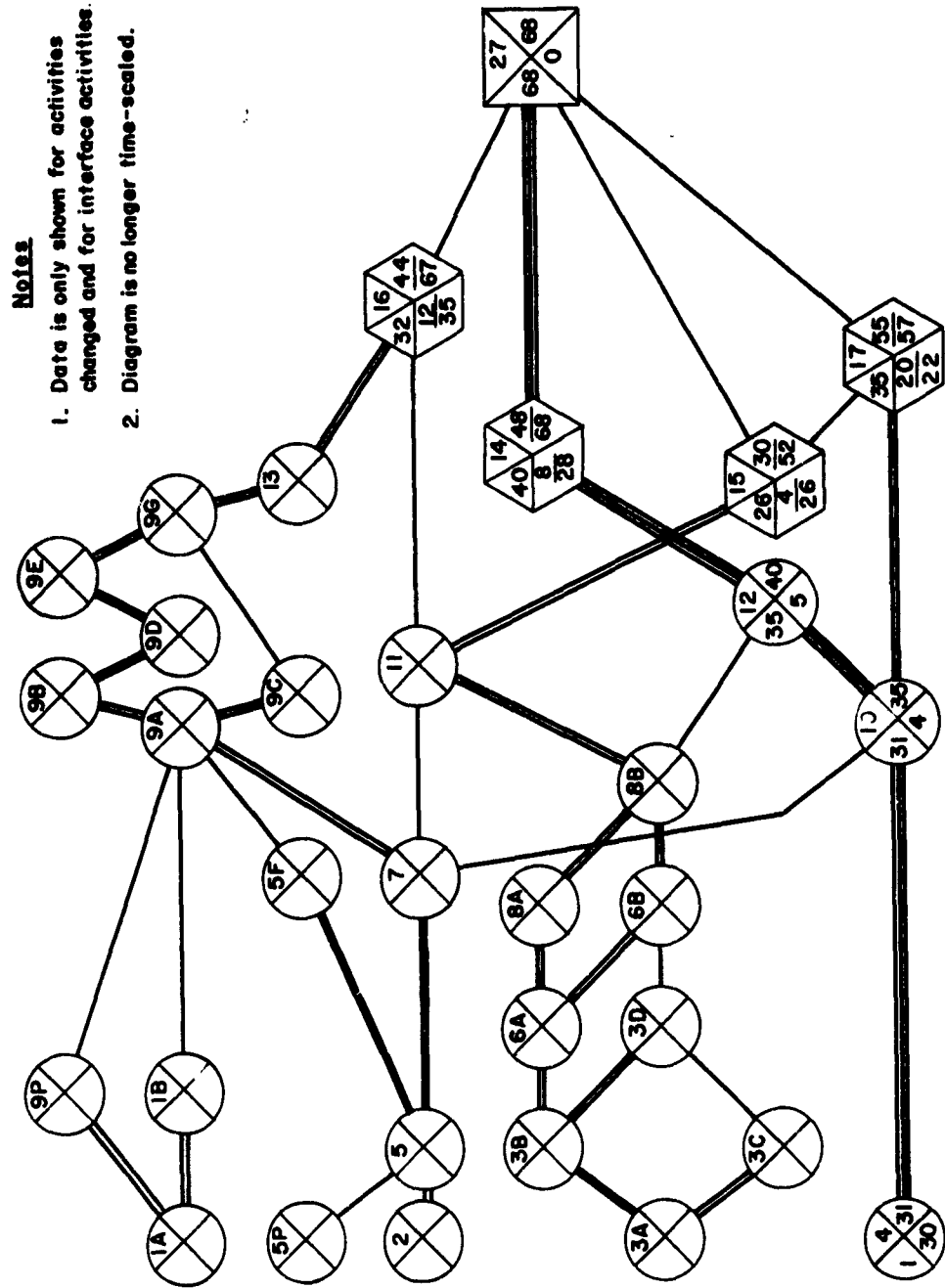


Figure 44. Subnet A Updated to Show New Critical Path

nodes to new positions. Of course, if changes caused later activities to be rescheduled to dates earlier than the dateline cutoff, these should be added to the right portion of the subnet diagram and interface data calculated for them. Figure 44 reflects this treatment.

While work covered by Subnet A is in progress, Subnet B should be prepared to extend detailed planning through the next time period. This subnetwork would be expanded from the basic plan of the overall network in the same general manner as Subnet A was expanded. Subnet B (without any expansion of detail) is shown on Figure 45. It covers the month of August and extends from the 32nd to the 53rd working day. It is plotted to a time scale which may be discarded when further changes occur. Before plotting Subnet B, it is usually advantageous to update the entire master network using the latest available information. Any further replanning of the overall project should be considered at this time. For example, in this case the planner decided to reschedule Activity No. 25 to start at its earliest start date rather than at the later date selected previously.

To begin construction of Subnet B, all activities in progress on the initial cutoff dateline (the 32nd day) are plotted on that line regardless of their start date. Then the remaining activities that commence later but before the terminal cutoff dateline are added to the diagram. In Figure 45, the dual value interface system has been employed. The calculated durations are shown and may be checked by the reader as a practice exercise. Where activities in Subnet B are preceded by completed activities in Subnet A, reference ties to the left margin are given for informational purposes. These ties are not essential for computational purposes, since the earlier activities are assumed to be completed and can have no further effect on the activities of Subnet B.

In summary, this system of network breakdown is based on the desirability of doing detailed planning and day-by-day updating for only that portion of the project to be performed in the near future. Moreover, since this is the type of planning and updating that should be done at the job office, smaller subnets permit the use of

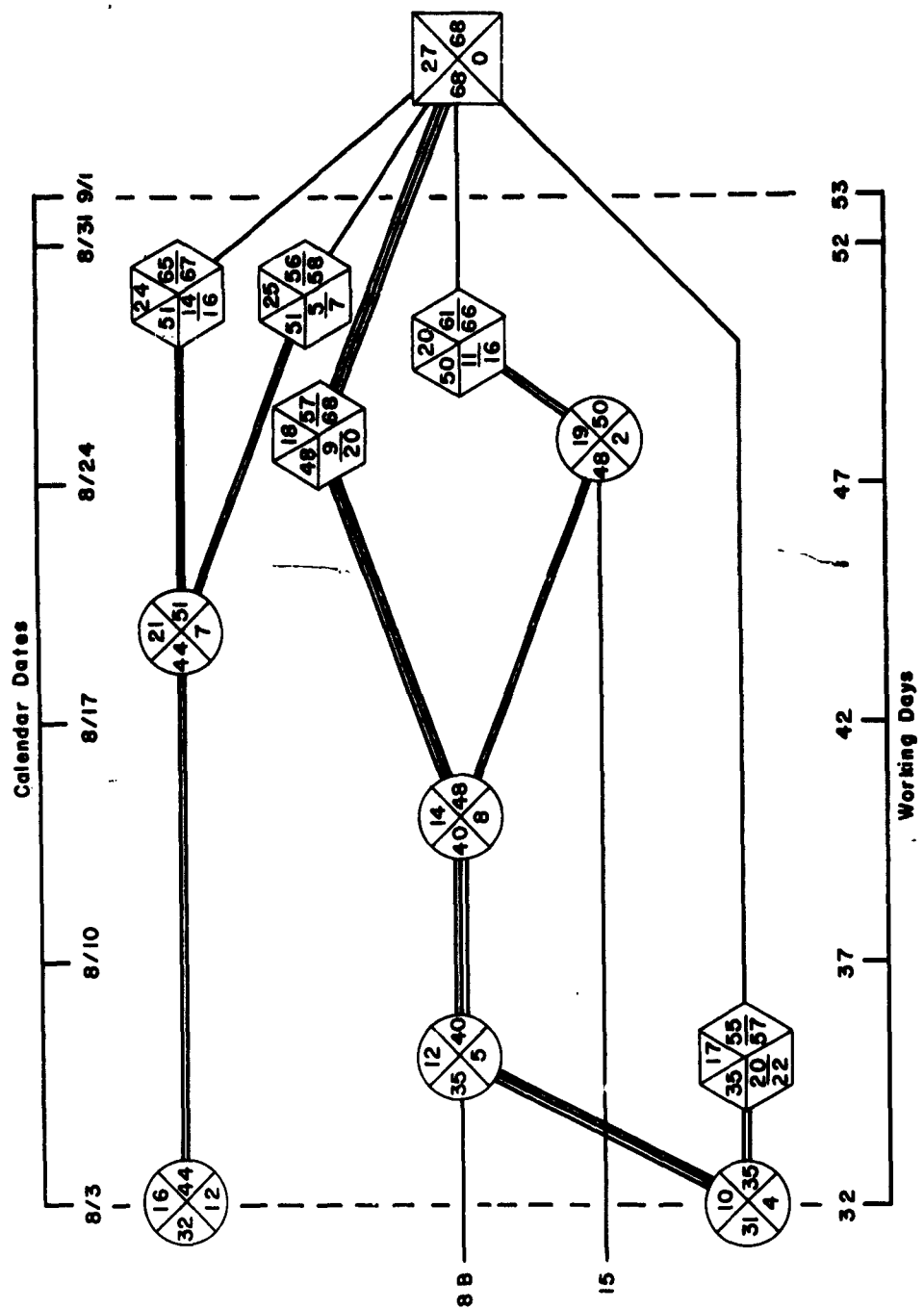


Figure 45. Subnet B

non-computer procedures that are compatible with the job office level of activity. This system assumes that the overall project network will be developed, showing only the detail that is essential to gross planning, and that the basic time calculations will be made. This may be accomplished by computer or non-computer methods, depending on which is more convenient. Then some overall replanning of the project may be in order if the project duration is not satisfactory, if some obviously excessive resource requirements are apparent, or if some advantageous time-cost tradeoffs can be discovered. If special computer programs are available to assist in these types of project replanning, they should be utilized. After the overall plan is deemed satisfactory, the first subnet, representing the first time period of the project, is developed. The corresponding portion of the master network will usually be considerably expanded to show the detail warranted on the job office level for work to be undertaken immediately. Detailed planning, including attempts to smooth resource requirements as well as to keep them within acceptable limits, is justified for the period ahead represented by the span of the subnet. Daily updating of activity scheduling for this limited period into the future is of real value even though it will be largely a waste of effort for activities of the more distant future. After this detailed planning is well along and the field work has progressed into this time span, the subnet for the next time period should be developed in a similar manner. This procedure should be repeated as the field work progresses into each new subnet time span. The completion of a subnet period is a logical point at which to update the master network. This periodic updating might ideally be accomplished by computer methods, making this system very appropriate for the combined use of both computer and non-computer methods. Again, any replanning of the overall network that is practical should be undertaken at these periodic datings. The data obtained also furnishes an independent check on the day-by-day updating calculations that have been made up to that point.

CLOSURE

This report has proposed and discussed certain measures that will extend the range of application of non-computer methods in planning and scheduling by critical path techniques. The emphasis has been on the development of and mechanics for these proposed measures. Their use for updating data as changes occur should be apparent and not require further treatment.

The use of these measures in the replanning of projects to improve performance and reduce costs is potentially of even greater importance. This replanning, frequently referred to in the report, involves such techniques as time-cost tradeoffs, resource allocation, and further detailed analysis of the basic project plan. The author believes that each of these techniques can be most intelligently applied if the planner is provided the means for tentatively making a proposed change, updating the plan and schedule accordingly, observing the effects produced, and reaching a decision, all before further replanning is continued. This approach becomes practical, at the job level, with the introduction of the procedures that have been developed here.

A more detailed treatment of the implementation of these methods in replanning construction projects is the subject matter of a report now in preparation under this contract.

A detailed bibliography containing 139 references closely related to critical path techniques was prepared by Robert C. McLean, Research Assistant under this contract, and is presented in his report entitled "A Basic Critical Path Method with Introduction to Several Special Applications," issued September, 1964.

The author wishes to acknowledge the assistance rendered by Professor C. H. Oglesby, Department of Civil Engineering, Stanford University, in reviewing and editing this entire report.