J. E. Rowings, D. J. Harmelink

Micro-Computer Based Linear Scheduling Application for Highway Construction Project Control

Final Report

October 1995

**Iowa Department
of Transportation**

report

**College of
Engineering
Iowa State University**

J. E. Rowings, D. J. Harmelink

Micro-Computer Based Linear Scheduling Application for Highway Construction Project Control

Final Report

October 1995

TABLE OF CONTENTS

INTRODUCTION

Today, more than ever before, the critical path method (CPM) scheduling technique is being applied to a myriad of construction projects. The primary reason for this is that micro-computers have made CPM scheduling inexpensive and relatively easy to use. In the construction industry, CPM scheduling has traditionally been applied to building and industrial process types of construction projects. Over the past several years however, the technique has been applied to highway type construction projects by the transportation departments in most states. Recently, the ability of CPM to model highway type projects and provide useful information has come into question. This work proposes an alternative technique for scheduling highway construction projects commonly known as "linear scheduling."

A literature review is presented to provide a history and a summary of the efforts to utilize linear scheduling for various types of work. Following the literature review, a discussion of two research projects conducted with the Iowa Department of Transportation involving alternative scheduling techniques and ultimately linear scheduling, will introduce the section on the development of the Linear Scheduling Model (LSM). The third section will include a comparison of the Linear Scheduling Model (LSM) to conventional Critical Path Method (CPM) scheduling techniques. The final section of this document will include routines developed in AutoLISP which allow AutoCAD to perform LSM tasks.

LITERATURE REVIEW

This review of literature will concentrate primarily on linear scheduling techniques. The term "Linear Scheduling Methods" will be used to describe a variety of related scheduling techniques. The origins of these scheduling methods are not clear and may, in fact, have multiple origins quite possibly in different countries. There is a wide variety of names and approaches associated with Linear Scheduling Methods but they have common features. The methods are usually concerned with repetitive units of work and the effects of resource allocation on the rates at which these units are produced. Several of the methods presented in the literature will be discussed.

Line of Balance (LOB)

The LOB technique is a scheduling method developed by the U. S. Navy in the early 1950s. It was first applied to industrial manufacturing and production control, where the objective was to attain or evaluate the flow rate of finished products in a production line (Sarraj 1990).

The objective of LOB scheduling for a repetitive process may be summarized as ensuring that (Arditi and Albulak 1986):

1. A programmed rate of completed units is met

2. A constant rate of repetitive work is maintained

3. Labor and plant move through the project in a continuous manner such that a balanced labor force is maintained and kept fully employed.

4. The cost benefits of repetitive working are achieved.

The LOB technique requires the following three inputs (Carr and Meyer 1974):

1. A unit network showing activity dependencies and time required between activity

    and unit completion;

2. An objective chart showing cumulative calendar schedule of unit completion; and

3. A progress chart showing the completion of the activities for each unit.

The unit network produced for LOB is similar to a traditional activity-on-arrow

network except that it is a network showing assembly operations for one unit of many to be

produced (Johnson 1981). The required time between the completion of each activity and the

completion of the unit is calculated in a way similar to the earliest start times in CPM, but

depends on unit completion rather than on unit start (Carr and Meyer 1974). The objective

diagram of the LOB technique is, in many, ways similar to the typical LSM diagram (see

Figure 1). Both use time as one axis, usually horizontal, and some measure of production as

the vertical axis. The objective diagram of LOB is used to schedule or record the cumulative

events of unit completion versus project time. The LSM diagram, however, is used to plan or

record progress on multiple activities that are moving continuously in sequence along the

length of a single project.

The progress diagram, prepared for a particular date, graphs a vertical bar for each

task which shows the actual number of units completed (see Figure 2). Actual progress is

compared to the line of balance, the unit activities which must be completed by the date of

interest if the unit completion schedule of the objective chart is to be met for the dependencies

and duration requirements shown on the unit network.
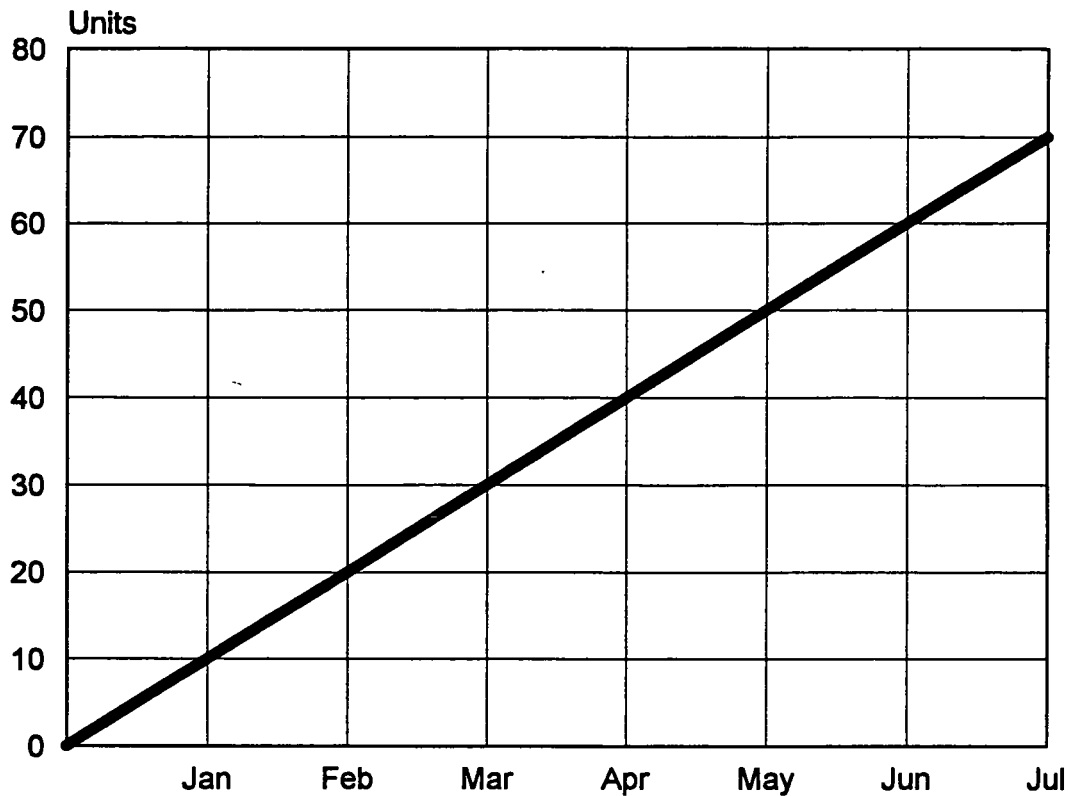
Figure 1 Objective Chart

Several researchers and practitioners have applied the LOB method, in some form or

another, in the construction industry as a scheduling technique. Khisty (1970) applied the

LOB technique in the classical sense of manufacturing, giving as examples the training of a

large number of supervisors, the production and supply of precast concrete beams, and

improvements and repairs to a harbor (Khisty 1970).

Carr and Meyer (1974) concluded that LOB is a practical tool for planning,

scheduling, and controlling the construction of repetitive building units. It can be useful for

Units
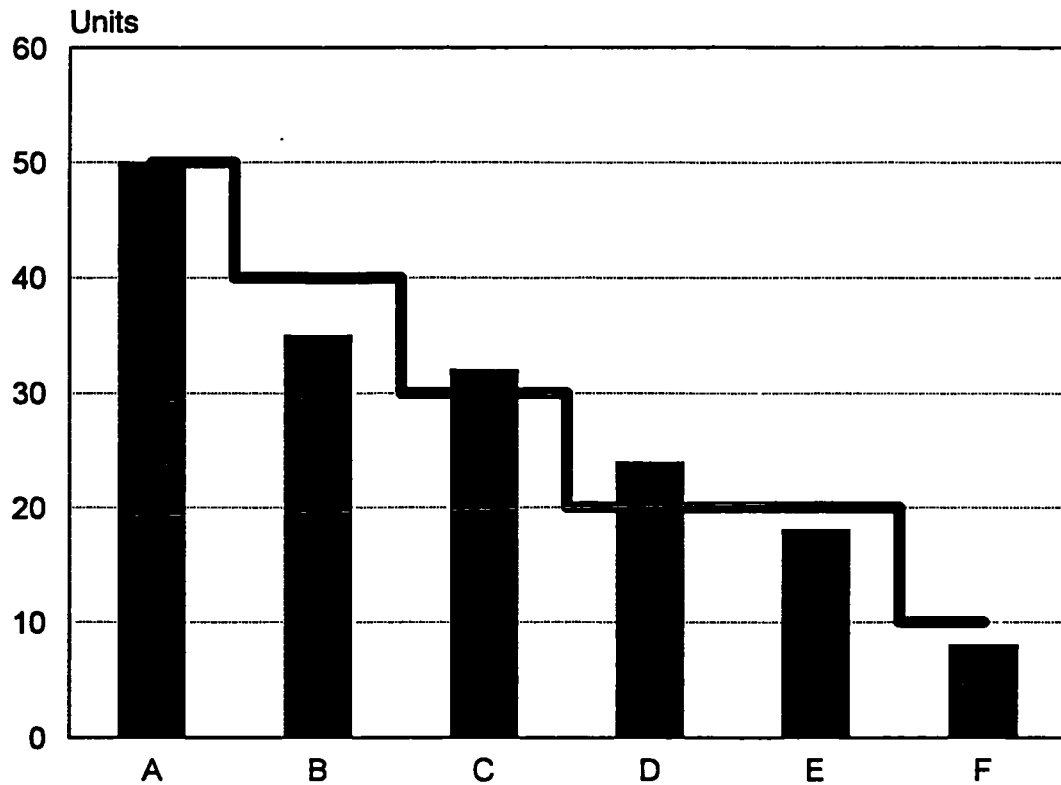
Figure 2 Progress Chart for Specific Date

decisions of field personnel because it is easy to calculate and understand, and can be adapted

to a variety of project types. They state that it is a valuable addition to the methods available

to the construction manager in directing his projects (Carr and Meyer 1974).

Arditi and Albulak (1986) applied LOB techniques to pavement construction and

made the following observations:

1. LOB is extremely sensitive to duration estimates for the unit activities.

2. Stage buffers need to be used to compensate for delays in production.

3. LOB does not incorporate the "learning" phenomenon, stage buffers are used to adjust.

4. An LOB schedule is easier to prepare than a network schedule, especially as repetition increases.

5. Degree of detail on LOB diagram must be carefully evaluated to control legibility.

6. The choice of appropriate scale is critical for understanding and communicating the information contained in the LOB schedule.

7. Modeling an entire project with only a unit network is not a reliable solution.

8. Since the LOB schedule is based on production rates that depend on resource useage, it yields good insight into the project at an early stage of the scheduling process.

9. Care must be taken to insure that floats are not used arbitrarily or indiscriminately in the preparation of the LOB schedule.

10. Foreman·and subcontractors were more receptive to LOB diagrams than arrow diagrams, but not receptive enough to replace bar charts.

11. It is possible to show progress on an LOB diagram.

Arditi and Albulak (1986) also expressed some shortcomings of the LOB method. They encountered major problems with the presentation of the LOB information, and suggested the use of colored graphics to assist the user in differentiating between overlapping activities with equal rates of production. The legibility of the diagram was also very sensitive to the drawing scale. Finally, since the LOB schedule is extremely sensitive to errors in the

activity duration estimates, a great deal of care must be exerted in the selection of production rates (Arditi and Albulak 1986).

Sarraj (1990) presented a formal development of the LOB method and the required algorithms to put the technique on a mathematical basis. By using the developed program, it is possible to find the project duration, the actual delivery schedule, and all related information regardless of the size of the project. Sarraj (1990) claims that by using this method in its formalized form, there is no need for any diagram to be drawn as a means of defining the schedules. The graphical representation of the required and real schedules is used for illustration purposes in the process of project control (Sarraj 1990).

Linear Scheduling

The most obvious difference between LOB techniques and linear scheduling methods may only be a question of emphasis. LOB places emphasis on the balance line of the progress diagram while LSM emphasizes a diagram similar to the objective diagram. A typical LSM diagram depicts time along the horizontal axis and some measure of repetitive units along the vertical axis. Instead of showing only the completion of units, the LSM diagram shows lines representing all of the activities involved in the completion of the repetitive units.

Since the early 1970s, several variations and methods of linear scheduling have been presented. Some of these methods use the term linear scheduling; some do not. A chronological history of these methods along with their various names will be presented.

In the early 1970s, Peer and Selinger developed a linear scheduling method in the process of analyzing parameters affecting construction time in repetitive housing projects

(Johnson 1981). This method became known as the construction planning technique (CPT) and has been applied to projects with discrete activities as well as linear activities. Peer states that the construction planning process can be divided into the following steps (Peer 1974):

1. To break down the project into constituent components.

2. To divide realization of these processes between adequate production crews.

3. To define technological connections between crews and activity categories.

4. To decide which production line should dictate the progress pace of the project, due to economical considerations or resource limits.

5. To make an approximate estimate of the resulting construction time and decide how many production units should be employed in parallel.

6. To balance the progress of noncritical production lines with that of the chosen critical one, aiming at achieving working continuity.

7. To check the possibility, within practical limits, of shortening construction time by introduction of planned breaks in continuity or changes in crew size.

8. To analyze the whole process in terms of time and activity durations and produce a plan.

Peer (1974) further proposes that planning the construction process is not a problem of finding an incidental critical path from arbitrarily determined single activity durations. On the contrary, one of the main steps in planning production is to define what should be made critical. This may be dictated by cost considerations or shortage of certain skilled labor. Peer

Table 1 Planning Procedures Compared

NO.	Construction Planning Technique	Network Analysis
1.	Breakdown of project into constituent component processes	Prepare a list of single activities
2.	Divide realization of these processes between production crews	
3.	Define technological connection between crews and activity categories	Define technological connections between single activities
4.	Chose bottleneck of the project and analyze its possible progress speed	
5.	Estimate total time and determine number of organizational units	
6.	Balance the whole production process	
7.	Check possibility of shortening time by introduction of practical breaks in continuity or change in crew sizes	
8.	Analyze the whole process in terms of time and activity duration	Estimate single activity durations and calculate a fortuitous critical path and available floats
9.	Produce a progress schedule	Produce a progress schedule

(1974) states that many methods of including resource allocation in network analysis have been employed, but that all of these programs remain sophisticated only from a theoretical point of view and are not capable of changing the original unrealistic schedule into a practical solution for the site. A comparison of the procedures is shown in Table 1.

The article further suggests that since available network-analysis methods cannot produce a realistic schedule, development of other construction planning procedures is

essential. The input of such programs should be based on production data, (e.g., quantities of work, production rates, external constraints, and other production characteristics), which are available or can be defined.

O'Brein (1975) proposed a method of scheduling repetitive units (floors of a high-rise building) called Vertical Production Method (VPM). He states that when the high-rise project reaches the first typical floor, the whole project momentum shifts gears. Even though each floor becomes an individual project with a well-defined start and completion, the schedule is no longer determined simply by the length of that project per floor. Rather, the schedule is now controlled by the time required for each of the major trades to work upward through the building (O'Brein 1975).

The method demonstrates a graphical method for developing the schedule for a high-rise project. A traditional LSM diagram is produced and activities are placed on the diagram based on production rates and connections to related activities. O'Brein (1975) states that the method produces a schedule which is not only the equivalent of a network, but is more readable for most contractors and subcontractors, the method is less costly, and produces a result much more rapidly. O'Brein (1975) believes that it is possible to computerize the VPM to produce an overall complete CPM network for the entire project, but does not elaborate on how this would be accomplished.

Selinger (1980) proposed a method using dynamic programming to find a solution for the minimization of time duration on linear civil engineering projects. Since the minimal duration involved the optimal quantities of resources, cost of the project would be minimized

as well. This method is based on the labor requirement and feasible crew size, rather than on activity durations determined in advance. The method allows for continuity between activities and is more suitable for the practical needs of a linear construction project (Selinger 1980).

Birrell (1980) stated that the CPM/PERT concepts, while marginally useful, are comparatively foreign to the construction process which has heuristically evolved over centuries to satisfy the needs of the situation of construction, and that the typical CPM approach to resource allocation is too simple for construction. Therefore, CPM tends to be unsuitable to meet the needs of the construction situation. The author proposes that the simplest and most appropriate way of modeling resource requirements in a project is to consider each work squad as a continuous flow. He suggests, that in construction, the crucial planning decisions are not of accurate durations of specific operations, operation duration floats, individual task resource allocations, calculations of probabilities of tasks finishing as planned, etc., but are (Birrell 1979) rather as follows:

1. The sequence of the object to be constructed into the most suitable "work locations;"

2. The sequence in which all the work squads will pass through these locations;

3. The analysis of the required work into suitable types or groups for individual work squads in that project in that locality; and

4. The fineness or accuracy in the measurement of time for that construction project.

Birrell produces a matrix for the construction process with time phases on the horizontal axis and work locations of the vertical axis. Each cell in the matrix represents a

"work package" which represents a quantity of work to be carried out by a specific work squad. Each work squad flows through the matrix along a diagonal path. Birrell proposes applying queuing theory to account for interaction between crews and thereby maximize production rates.

Johnson (1981) developed a linear scheduling method for highway construction. He proposed that many transportation projects exhibit a linear characteristic, in that the activities progress continuously in sequence along the length of the project. A traditional linear schedule diagram was depicted that included graphical representations of activities other than diagonal lines as well as varying production rates on linear activities. A method for time-cost analysis was also presented. This work determined that linear scheduling has possible applications in transportation construction projects as well as other types of repetitive projects and made the following notes (Johnson 1981):

1. LSM provides more information concerning the planned method of construction than a bar chart.

2. In certain types of projects, LSM offers some advantages over the network approach. Network methods are a more powerful tool for most situations, especially for projects with discrete activities. However, in repetitive portions of projects, LSM more quickly conveys the nature of the problem, aids in its solution and presents the intent. In a single project having both types of work, each type of scheduling can be applied to respective portions and coordinated.

3. Among the transportation-related projects that could be scheduled using LSM are highway construction, highway resurfacing and maintenance, airport runway construction and resurfacing, tunnels, mass transit systems, pipelines and railroads.

4. Although the method is not new, it has been given very little exposure among highway contractors.

5. Highway contractors when contacted, indicated interest in the method and were of the opinion that it may have some potential.

6. The LSM scheduling method can assist in organizing construction work and reducing construction time; thus, it has measurable benefits in cost and safety which can offset the cost of development.

7. Implementation of LSM will require transfer of the scheduling technology to contractors. This would be followed by trial field use, feedback, improvements and reuse until the method, if beneficial, is accepted.

8. Contract-letting agencies might consider either allowing LSM as an alternative to a required bar chart or requiring both on some projects to encourage trial use by contractors.

9. Perhaps the most significant advantage of LSM is the simplicity with which it can convey a detailed work schedule. When the schedule is easily understood by larger proportions of field staff and workers, the schedule becomes a goal which can lead to improvements in productivity and reduced cost. Schedules

developed and analyzed using more powerful network analysis methods,

perhaps involving lead/lag techniques, can be charted in the form of LSM

diagrams as a means of simply conveying the analysis results.

Stradal and Cacha (1982) present a scheduling method labeled Time Space Scheduling

Method for repetitive activities and for scheduling work in sections of a project. The diagram

produced is of "rhythmical" flow of production lines. The horizontal axis represents location

or repetitive units while the vertical axis represents time increasing down the axis. The article

presents examples of schedules for a variety of projects with various graphical constructs

representing activities. Stradal and Cacha (1982) state that the method's main advantages are:

the planning of work in sections, the possibility of arrangement of a flow line which

contributes to a smooth use of capacities, learning effects, and the intelligibility of the

diagram. The main limitation is in the character of complex projects, where a mutual scale for

all parts cannot be fixed. The time space scheduling method is useful for line-formed projects

(sewers, roads, bridges, tunnels) and buildings (especially tall buildings), and above all, in

projects suitable for flow line methods (Stradal 1981).

Chrzanowski and Johnston (1988) applied LSM to an actual roadway construction

project. CPM was initially used to schedule the project. LSM was used at a later date and

during development some improvements became apparent. The linear schedule diagram was

similar to the diagram presented earlier by Johnson (1981). This diagram shows earthwork

operations as a series of quadrilateral plots. Earthwork operations do not move along a

continuous path as do linear activities, but instead an entire area is worked on until the

subgrade is attained. This diagram also demonstrated a space between linear activities to represent cure time. (Chrzanowski and Johnston 1988) make several observations regarding LSM:

1. The most obvious characteristic of LSM is simplicity.

2. The LSM diagrams easily convey detailed information that is comparable to what may be derived from an equivalent CPM schedule.

3. Job progress, resource allocations, and schedule changes can be performed quickly.

4. The user receives fairly detailed information without being confronted with numerical data and the degree of abstraction found in network methods.

5. The use of LSM is restricted to construction projects consisting of repetitive activities.

6. Discrete activities can be included, but if more detail is required, the activity must be referenced to a network schedule.

7. LSM is essentially graphical and cannot be adapted to numerical computerization as readily as network models. However, CAD drafting systems could be programmed to facilitate preparation and revision.

8. LSM would best be utilized as a complement to CPM.

Russell and Caselton (1988) present a method for making linear scheduling a practical computer-based tool for repetitive construction projects. They developed a two-state variable, N-stage dynamic program formulation of the linear scheduling problem that accounts for several of the realities of repetitive construction, including generalized precedence

16

relationships and the ability to treat a variety of work continuity constraints. A sensitivity

analysis procedure is described that permits the identification of near optimal solutions which

provide the user with schedule alternatives that might better suit additional nonquantifiable

criteria. Russell and Caselton (1988) suggest that future research include generalization of the

dynamic programming formulation to include certain nonserial cases, investigation of

appropriate functional representation and values of the time penalties when work interruptions

occur, and the inclusion of both cost and time in the optimization criteria.

Moselhi and El-Rayes (1992) present a dynamic programming model for generating

optimized schedules for repetitive projects that incorporate cost as an important decision

variable in the optimization process. For each repetitive activity in the project, the model

assists the planner in selecting the optimum crew formation from a set of possible alternatives.

Vorster, Beliveau, and Bafna (1992) propose that a standard format and set of

symbols be developed for use in drawing linear schedules. Vorster and Bafna (1992) in a

discussion of a paper by Sarraj (1990) suggest that projects that are characterized as linear

may be divided into two categories. The first category includes projects that are linear due to

the uniform repetition of a unit network throughout the project. The second category of

linear projects are characterized by a group of linear activities that progress along the

geometrical layout of the project. Highway projects are excellent examples of this category.

Vorster and Bafna propose that the term linear scheduling method (LSM) be used for the

special graphical scheduling technique suited to projects that have linear geometrical

characteristics but are not characterized by repetitive unit networks. The term line-of-balance

(LOB) should be reserved for the method used to schedule projects characterized by repetitive execution of unit networks describing repetitive operations.

Russell and Wong (1993) developed a set of planning structures that provide a means of marrying critical-path planning with linear scheduling. These structures include the continuous activity, the ordered activity, the shadow activity, and the cyclic activity and are a superset of the traditional CPM activities (Russell and Wong 1993).

LINEAR SCHEDULING MODEL

This section will describe the development of the Linear Scheduling Model (LSM). The Linear Scheduling Model will be capable of identifying controlling activities and a controlling activity path from the activities portrayed on a linear schedule. The controlling activity path will be identified through the performance of an upward and downward pass, analogous to the forward and backward pass used in CPM scheduling techniques. The model will identify non-controlling activity segments and provide a means of identifying rate float in linear activities. Rate float can also be compared to the various types of float found in CPM scheduling techniques. Once the basic planning model has been developed, project management techniques such as activity progress analysis and reporting, project updating and status reporting, and as-built linear schedules will be described. At the completion of this section, an approach for performing basic planning and project management functions using the Linear Scheduling Model (LSM) will have been fully developed. The section will begin, however, with a brief discussion of the research effort which lead to the development of the Linear Scheduling Model.

Research

Two research projects involving linear scheduling for highway construction have been completed for the Iowa Department of Transportation (IDOT) by Rowings and Harmelink (1993, 1994)(43, 44). The projects were undertaken because the Office of Construction's expectations for enhanced project control and improved communication of project objectives have not been fully met by the use of CPM.

The first project identified a technique known as the linear scheduling as an alternative to CPM on certain highway construction projects. In the initial phase of the project, the linear scheduling technique was applied to a highway construction project that was currently in progress. As-planned and as-built data was obtained for some of the major activities and a prototype linear schedule was developed. The results of this effort indicated that the linear scheduling merited further investigation.

The second research project was implemented to allow the research team an opportunity to evaluate LSM on a small diverse group of projects. Unlike the first phase of the project, the research team was closely involved in the project from early in the planning phase throughout the completion of the projects. Three projects were ultimately selected. LSM as-planned and as-built schedules and CPM schedules were created for all three projects.

By applying linear scheduling to several projects from early in the planning phase through the construction of the project, the technique could be refined to meet the needs of the users in the following ways:

1. The benefits linear scheduling may present for Iowa Department of Transportation (IDOT) personnel in the areas of understanding the contractors "plan of attack" and also being able to monitor actual work progress would be identified.

2. A determination of how construction managers would use linear scheduling to manage their projects would be made. The necessary information, and form of the information, required by managers to function effectively would be

identified.

3. A consistent set of symbols to describe the various types of activities involved in a typical highway construction project would be developed.

4. A comparison of how CPM is used by Contractors and IDOT currently with how linear scheduling may be used would be made.

The following is a brief description of the three highway construction projects selected for the research effort:

Project I - 17.6 miles of P.C.C. inlay on I-80 east of Iowa City, Iowa. The work involved total reconstruction of both eastbound and westbound lanes. The project was allowed 210 working days and 185 closure days, and a CPM schedule was required.

Project II - 1.8 miles of grade and P.C.C. pave on Iowa 92 east of Indianola, Iowa. The project was allowed 65 working days and a schedule was not required.

Project III - 1.85 miles of grade, drainage and P.C.C. pave on 53rd Street in West Des Moines, Iowa. The project was allowed 160 working days and a CPM schedule was required.

Linear schedules were prepared for each of these projects by using time and duration information from the CPM schedule and adding information concerning the location and direction of activities. These schedules were presented to the contractor and the IDOT personnel at the beginning of the project, and as-built schedules were prepared weekly and presented at progress meetings.

The As-Built Project Schedules

As-built schedules were created for all three projects. Portions of these schedules are presented in Figures 3 through 5 respectively. In the production of these schedules, an attempt was made to use a common set of elements.

In all of the diagrams, the x-axis represents the project site denoted as stations, usually in increments of 5 or 10. The y-axes are always a measure of work periods, but they are also a means of reconciling the four individual measures of time included in these diagrams. The y-axes provide a correlation between actual calendar days and work periods. The thick vertical lines either along the left-most y-axis or closest to the y-axis line indicate planned working days, days that the contractor plans on working, and provide a calendar to establish planned activity dates. The thick vertical lines on the right-most axes indicate the charged working days. Each time period charged, in half-day increments, is indicated by a thick vertical line.

A box with a list of activity descriptions and a number of symbols is included on each schedule (shown here on the facing page). Whenever possible the code number corresponds to the number assigned to pay items in the project drawings. This number is also used to organize the quantity tabulation sheets recorded by the inspectors and serves as a legend for identifying activities on the schedule. If limited space on the diagram does not allow the use of the activity description, the code number of a symbol is used.

There is an axis legend included on each as-built schedule that describes the items included on the axes (shown here on the facing page).

IM-80-7(59)247--13-52

Linear Schedule Final As-built

Johnson County Inlay

Prepared By: D. Harmelink

Activity Codes

1. P. C. Concrete, Class C, 12"
8. Backfill Special
9. Subbase, Granular
11. Excavation Class 13, Rdw and Borrow
13. Bridge Approach Section
14. Removal of Pavement
15. Paved Shoulder, P.C.C.
22. Culvert Concrete Roadway Pipe, 24" dia.
28. Seeding, Fertilizing
31. Silt Fence for Ditch Checks
GR. Guardrail, Formed Steel Beam
68. Subdrain Longitudinal, Shld. 4" dia.
NA. Breaking Pavement
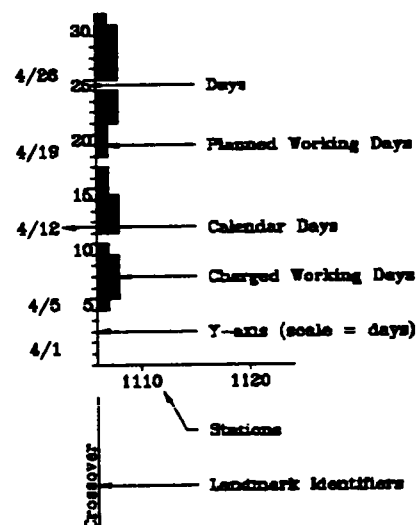NA. Rock Subbase Trim

Axis Legend

Figure 3 Project I

23
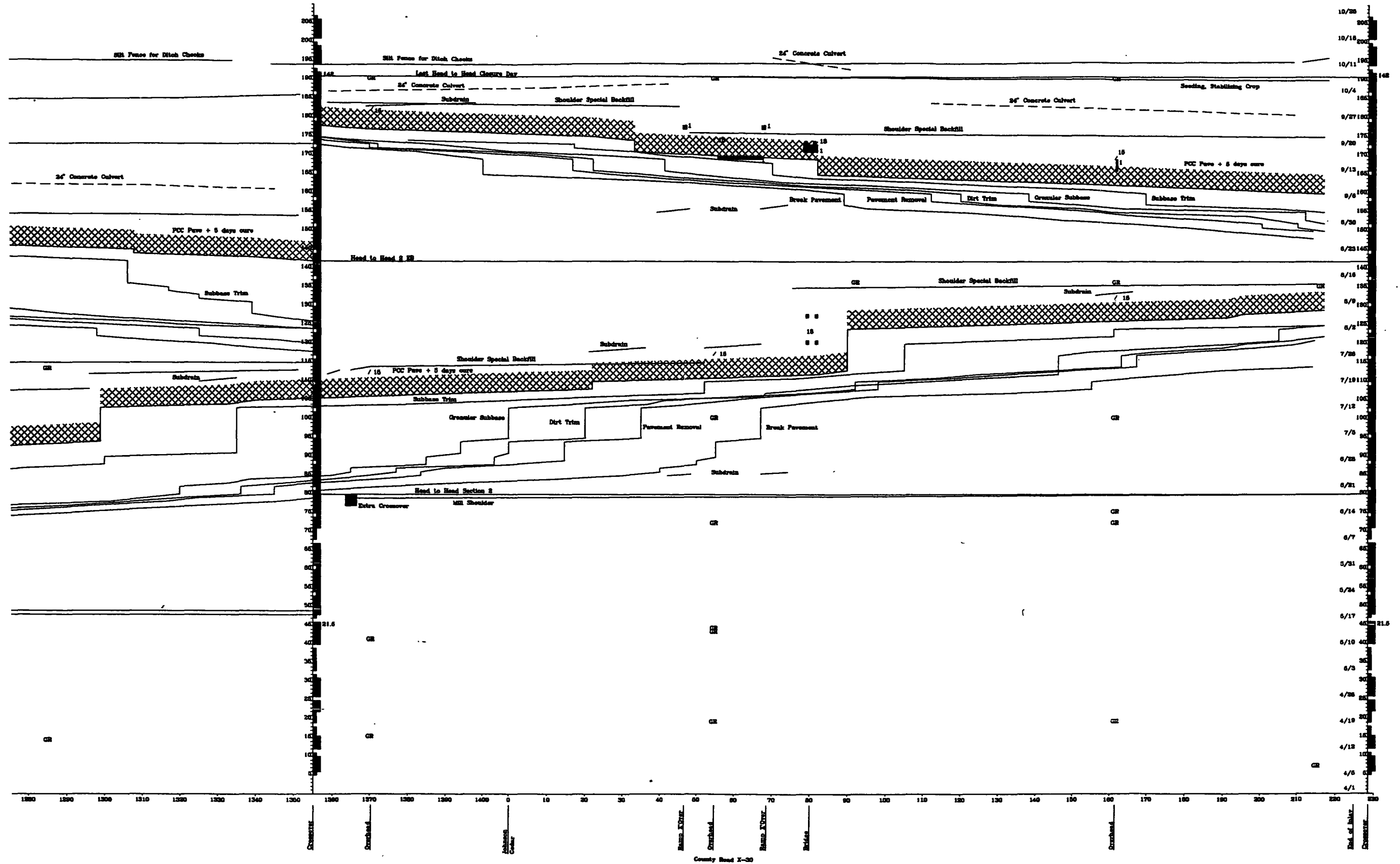
Activity Codes

1	Excavation Class 10 Rdw.and Borrow
5	Pavement Removal
6	Special Backfill
9	PCC Pave Shoulder
10	PCC Pave Bikepath/Sidewalk
12	PCC Pavement
I	Intakes
16	Storm Sewer
C	Culverts
40	Subdrain
42	Silt Fence
49	Pavement Markings
M	Manholes
55	Granular Shoulders
NA	Clear and Grub
NA	Milling

HES-92-5(27)--26-91

Linear Schedule Final As-built
Warren County Iowa 92
Grade and P.C.C. Pave
January 17, 1994
Prepared By: Dave Harmelink

Figure 4 Project II

Activity Codes

1	Excavation Class 10 Rdw.and Borrow
4	Special Backfill
5	Granular Subbase
6	Removal of Pavement
14	4" Longitudinal Subdrain (Shoulders)
I	Intakes
32	Storm Sewer
61	10" Class C P.C.C. Pavement
62	6" P.C. Concrete Median

Figure 5 Project III

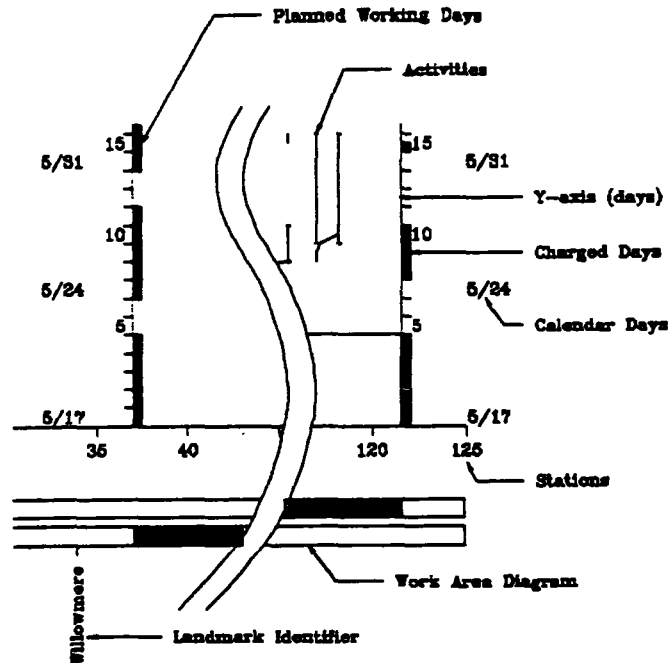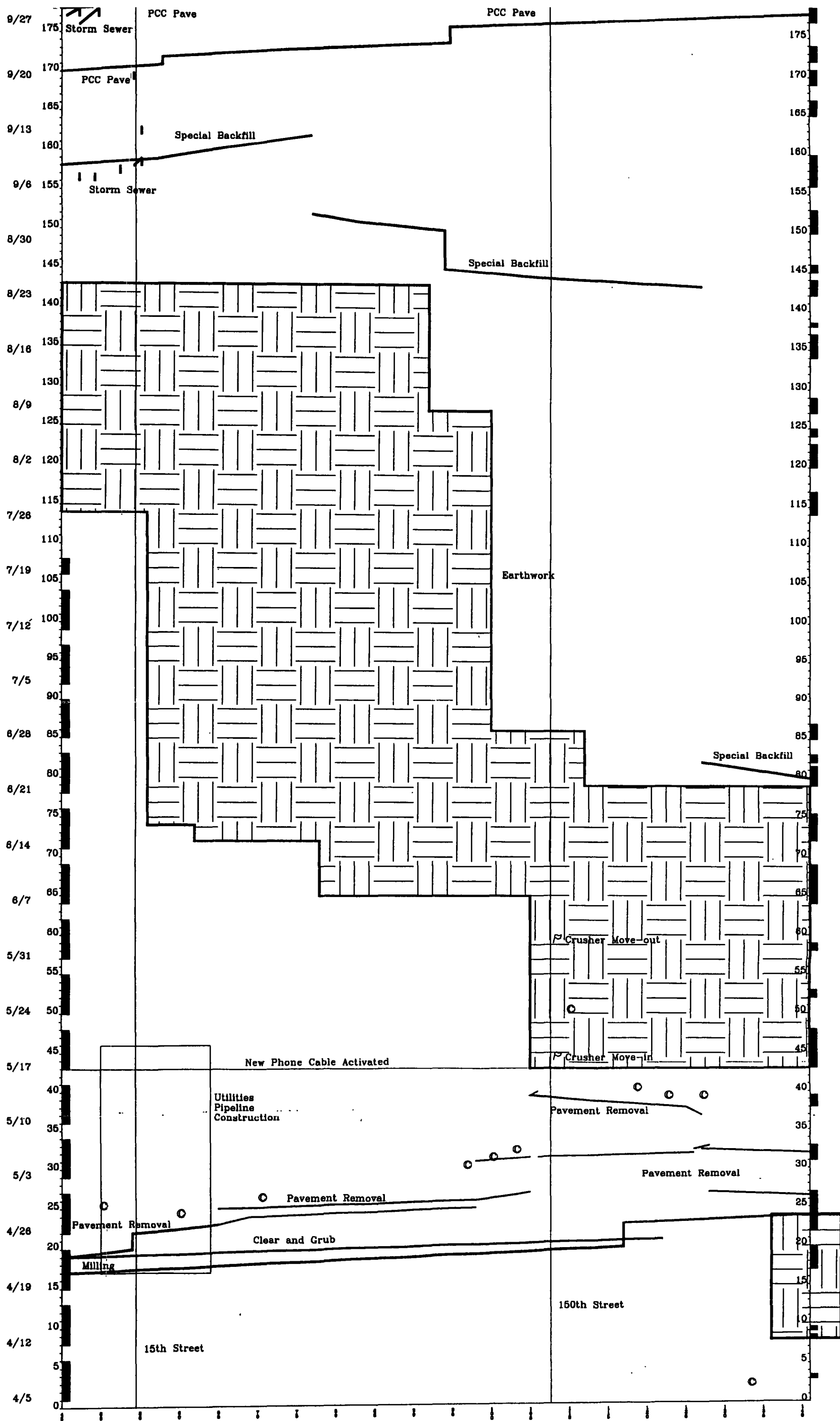|	
9/27	Storm Sewer · PCC Pave · PCC Pave
9/20	PCC Pave
9/13	Special Backfill
9/6	Storm Sewer
8/30	Special Backfill
8/23	
8/16	
8/9	
8/2	
7/26	
7/19	
7/12	
7/5	Earthwork
6/28	
6/21	Special Backfill
6/14	
6/7	
5/31	Crusher Move-out
5/24	
5/17	New Phone Cable Activated · Crusher Move-In
5/10	Utilities Pipeline Construction · Pavement Removal
5/3	Pavement Removal
4/26	Pavement Removal · Pavement Removal · Clear and Grub
4/19	Milling
4/12	15th Street · 150th Street
4/5	

Data Collection

Whenever possible, the data used in producing the as-built schedules was developed from the sheets that the inspectors use to record daily quantities for pay items. Figure 6 shows an example of one of the inspectors sheets for Subbase, Granular. The data recorded on this sheet includes the date, the location in stations, the lane, and current and cumulative quantities. In order to plot an activity such as mainline paving, the appropriate information needs to be assembled from this sheet. The numbers in the left margin indicate the work period on which this date falls and were added by the researcher. For example, granular subbase was placed in the eastbound lane on 7/22/93, work period # 113, from station 126+00 to station 146+09. This information would plot as a line on the as-built schedule with endpoint coordinates in the form of (x,y). The starting point of the line would be (126.00,113) and the endpoint of the line would be (146.09,114). The next time this activity progresses is on 7/26/93; work period #117. During the three day interval between the end of day 113 and the start of day 117, there was no progress recorded for the granular subbase activity. This plotted as a vertical line from (146.09,114) to (146.09,117).

A vertical section in an activity shown as a line on these schedules, usually indicates that there was not any progress recorded on this activity during this period of time. If the line indicating an activity was anything but vertical, progress has occurred, and the nearer the slope of this line is to horizontal, the higher the rate of progress. Keep in mind, however, that this measure of progress is in terms of stations along the roadway, and that actual quantities may vary along the course of the project.

29

Form 15

Item Description: _Subbase, Granular_   Page _2_ of ____

Item # _0090_   Project #_IM-80-7(59)247-13-52_ Contract # _3514?_

	Date	Location Sta to Sta		Side	Length	Width	Sq. Yds. Today	Sq. Yds. To date	Div.
84	6-23-93	1338+45	1359+85	EB Mainline	2140.0	27.67	6579.31	77797.83	II
	6-24-93	1359+85	1365+00	EB Mainline	515.0	27.67	1583.34	79381.22	I
87	6-26-93	1365+00	1384+89	EB Mainline	1989.0	27.67	6115.07	85496.29	I
90	6-29-93	1384+89	1394+01	EB Mainline	912.0	27.67	2803.89	88300.18	I
94	7-03-93	1394+01	1407+147	EB Mainline	1313.7	27.67	4038.90	92339.08	I
103	7-12-93	Equ. 147+14.7 Bk =0+00 Ahead 0+00	16+50	Mainline	1650.0	27.67	5072.83	97411.91	II
109	7-13-93	EB 16+50	EB 31+00	EB Mainline	1450.0	27.67	4457.94	101869.95	II
105	7-14-93	31+00	40+00	Mainline	900	27.67	2767.00	104636.85	II
106	7-15-93	40+00	46+50	EB Mainline	650	27.67	1998.39	106635.24	II
106	7-15-93	48+00	67+50	EB Mainline	1950	27.67	5995.17	112630.41	II
107	7-16-93	EB Exit Ramp West 41+50	47+50	Exit Ramp (See Page _ of _ Pavt Removal)			.229.17	113859.58	II
107	7-16-93	68+25	79+00	Mainline	1075	27.67	3305.03	117164.61	II
110	7-19-93	80+00	90+00	Mainline	1000	27.67	3074.44	120239.05	II
111	7-20-93	90+00	110+00	Mainline	2000	27.67	6148.89	126387.94	II
112	7-21-93	110+00	126+00	Mainline	1600	27.67	4919.11	131307.05	II
113	7-22-93	126+00	146+09	EB Mainline	2009	27.67	6176.56	137483.61	II
117	7-26-93	146+09	155+01	EB Mainline	892	27.67	2742.40	140226.01	II
117	7-26-93	EB Entr. Ramp West 68+03	76+50	Branch (See Page _ of _ Pavt Removal)			2395.89	142621.90	II

Quantity Awarded: _324852.000 Sq. Yds._

Quantity Authorized: _____

Quantity Paid: _____

Meas. & Insp. By: _Larry Hotz_   Method of Measurement _2111.08_

Checked By: _____   Basis of Payment _2111.09_

Figure 6 Sample Inspection Progress Log

Research Observations

The process of preparing the initial linear schedules, maintaining current as-built schedules and attending the projects' progress meetings led to several observations that were influential in the development of the Linear Scheduling Model.

The preparation of the initial linear schedules from the CPM data, produced some interesting results. If there were any logic errors, conflicts in location, or omissions in the original CPM schedule they became immediately obvious in the linear schedule. In the preparation of the CPM schedule, the scheduler had to mentally visualize the relationships between activities. For the schedule to be accurate, all of the possible relationships had to be established. These relationships were, at times, very difficult to describe using CPM techniques. For example, consider a paving operation that required five days of cure time before subsequent activities could proceed. This was typically modeled as a start-start relationship between paving and cure time in the CPM. Let's assume that the paving operation covered several miles and occurred over a period of several days. The cure activity relationship said that subsequent activities could start five days after paving started, but since paving could span many days, the true relationship was that subsequent activities could not start until five days after paving had occurred at any point along the course of the paving operation. CPM was unable to correctly model an activity that occurred at any location other than the same point at which paving started or an activity that progresses at a faster rate than that of the paving operation. In either of these cases, CPM would allow subsequent activities to occur before the five days of cure had elapsed.

The contractor and the researcher spent a considerable amount of time and effort in the development of the CPM schedule for Project II, and only after it appeared to be accurate was the linear schedule produced. A simple visual check of the linear schedule indicated that there was a problem with activities occurring within the cure phase of the paving activity. This problem was nearly impossible to detect from the original CPM schedule.

Output from CPM schedules can, at best, be difficult to understand. The contractor typically delivers a logic diagram and a tabular report showing activity dates. From this information, IDOT personnel try to discern if the contractor's plan is accurate. As the previous discussion indicates, this was difficult even for those that were very intimate with the schedule. The format of the linear schedule seemed to overcome many of these problems. A user of the schedule can visualize the project construction plan. Apparent on the schedule was the sequence of activities at any particular location on the project, the starting and ending date as well as the starting and ending location of all activities, and an indication of production for linear activities in terms of stationing. Any logic errors in the plan were clearly evident as physical interferences between the graphic entities depicting the activities on the schedule.

Presentation of the schedule at progress meetings supported the suggestion that it was much easier to convey the contractors plan for accomplishing the work using linear schedules than CPM schedules. The primary reason for this was that linear schedules add a very important dimension to the scheduling process-the location. In linear scheduling, *"where* an activity occurred" is equally as important as *"when* it occurred". This is evidenced by the

assignment of the axes on the diagram. In projects with predominantly linear type activities, this is an extremely important attribute.

Research Recommendations

The research indicated that there was strong potential for linear scheduling to have an impact on highway construction planning and scheduling. However, for linear scheduling to gain acceptance as a bonafide scheduling tool, would require that linear scheduling contain more rigorous analytical capabilities similar to those in CPM. Specifically, linear scheduling needs to include the following capabilities:

1. Determine the controlling activities and controlling activity path.

2. Reconcile the various calendars common to highway construction projects.

3. Identify non-controlling activity segments and establish a measure of float for the activities.

4. Provide a means to status activities and update the project to determine start and completion dates of future activities.

5. Capture as-built data and construct as-built project schedules.

CPM scheduling techniques have evolved into highly developed computer applications. Subsequently, CPM has become a widely used and accepted method for scheduling a wide variety of project oriented activities. Linear scheduling, on the other hand, has not gained wide acceptance as a planning or scheduling technique for two reasons:

1. As mentioned previously, an analytical model capable of providing necessary scheduling functions has not been developed, and

2. Until such a model is developed, computerization of the linear scheduling technique will remain unpropitious.
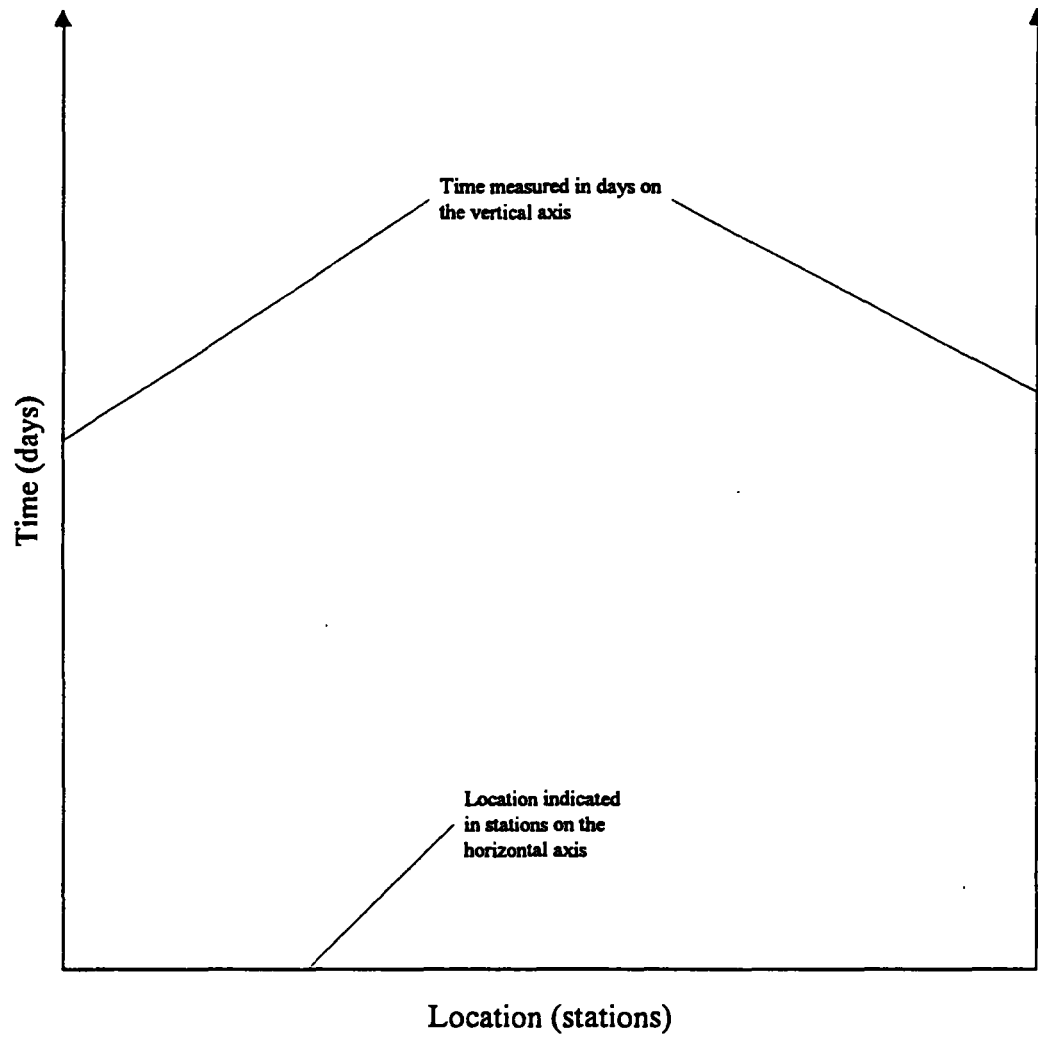
The following section describes the Linear Scheduling Model that has been developed to provide some of the basic scheduling functions outlined by the research effort.

LSM Graphical Entities

This section will describe the graphical constructs that are used in the Linear Scheduling Model. The graphical constructs fall into three categories: axes, activities, and other. The entities in each category will be described and illustrated.

Axes

A linear schedule has a horizontal axis and a vertical axis. The vertical axis is allocated some measure of time and the horizontal axis is related to a physical location or distance from a reference point for the project. The units for time are days and can be measured in several calendars which will be discussed in the section on calendars. For highway construction projects, the units for location are usually stations (see Figure 7). Scales need to be selected for the axes that allow the schedule to be legible on whatever output device or media is used. Output devices can range from video monitors to laser printers to large color plotters with media ranging from letter size to E-size paper and video resolution from stand VGA to high-resolution graphics. The area enclosed by the axes creates a visual planning field for the person creating the schedule. Each activity placed on this field has dimensions of time and a physical location.

Time measured in days on
the vertical axis

Time (days)

Location indicated
in stations on the
horizontal axis

Location (stations)

Figure 7 Linear Schedule Axes Definitions

Activities

All of the work on a particular highway construction project can be divided up in to specific activities. For the purpose of linear scheduling, these activities can be grouped into three general types:

1. linear activities,

2. block activities, and

3. bar activities.

Each of these activity types and there specific sub-types will be described and graphically represented in the following sections.

Linear Activities

A general definition of a linear activity is an activity that progresses along a physical path. This path is represented by the location (horizontal) axis on the linear schedule. At any point of progress along this path, the activity is complete up to that point. For example, consider an activity that involves the removal of old pavement prior to the replacement of the road. As pavement is removed along the course of the road, the activity is complete up to that point. Once the pavement has been removed at any location, there is never a need to go back and remove pavement that is no longer there. Any point along the path behind the current work location is now potentially clear for other activities to occur. This describes the nature of a linear activity.

Linear activities are represented by lines on a linear schedule as shown in Figure 8. A linear activity originates at some location, at some point in time, called the origin, and
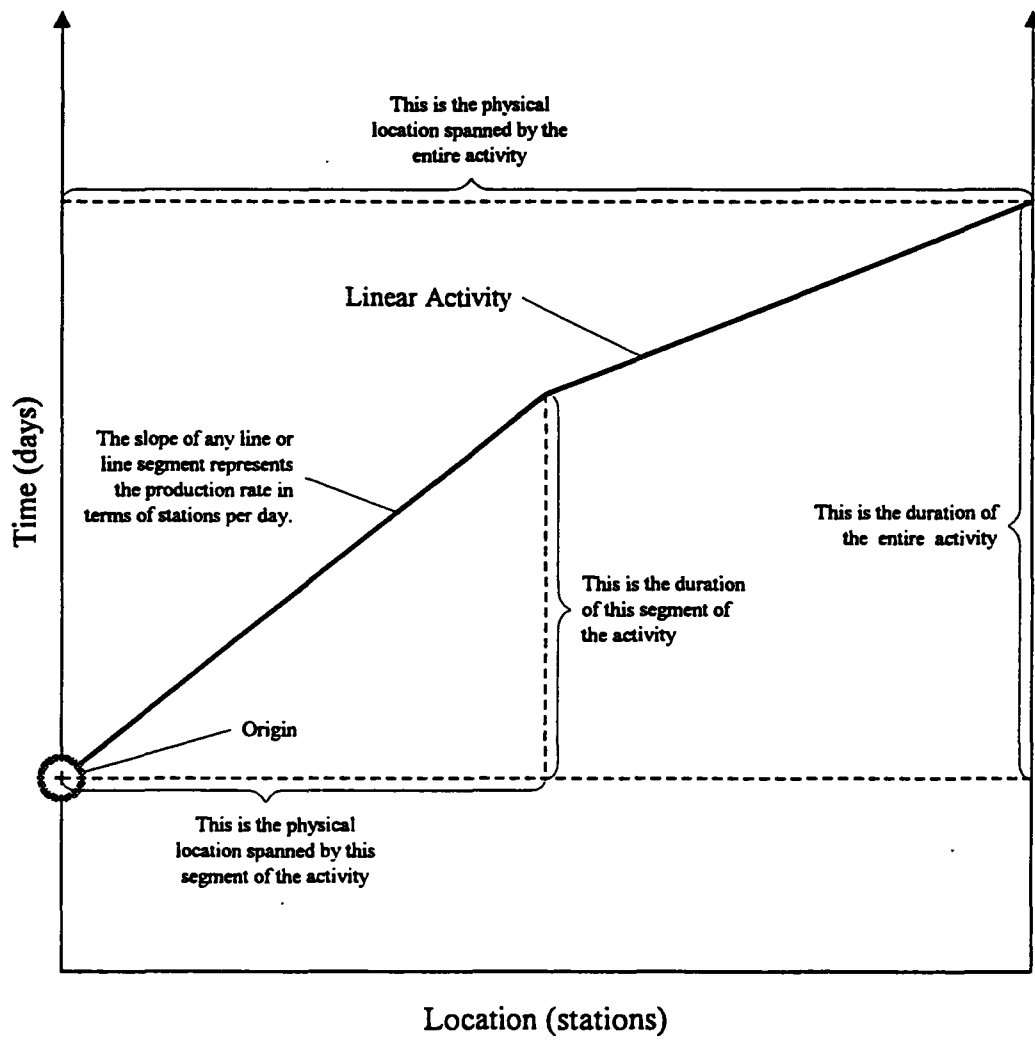
Figure 8 A Linear Activity

progresses to another location at some later point in time. These two points define the endpoints of a line. The slope of this line is a measure of the rate of production for this activity and can be expressed in terms of x-axis units, typically stations, per day. An activity can have varying production rates as indicated in Figure 8, but each individual segment must have a fixed, or linear rate.

The general linear activity type has four specific sub-types. These sub-types relate to whether or not the activity spans the entire location of the project and whether or not the activity is in continuous or intermittent operation. The distinction between continuous and intermittent activities will be made later. Table 2 shows the three main activity types, linear, block, and bar, and their sub-types. For linear activities, the four sub-types are:
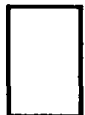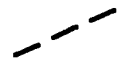
1. Continuous Full-span Linear - CFL,

2. Intermittent Full-span Linear - IFL,

3. Continuous Partial-span Linear - CPL, and

4. Intermittent Partial-span Linear - IPL.

The figure also shows the symbols used to depict the various activity types. Continuous linear activities are represented by a solid line while intermittent activities are represented by a dashed line. A full-span activity covers the entire project location while a partial-span activity only covers a portion of the project. The reason for the different activity types will be explained as the LSM model is described.

Block Activities

A block is used to represent an activity that occupies an area on the project for a
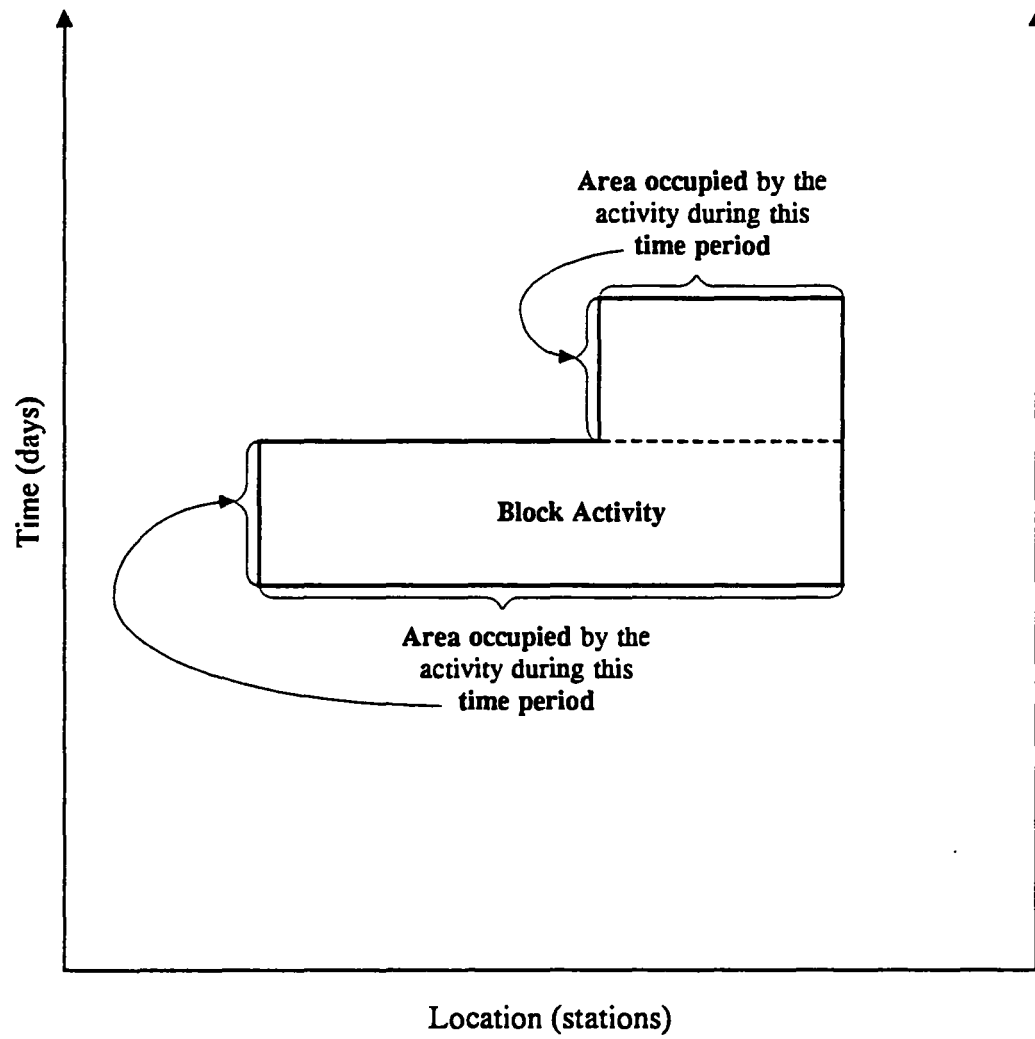
Table 2 Activity Types

		Linear	Block	Bar
Continuous	Full-span	CFL	FB	B
Intermittent		IFL		
Continuous	Partial-span	CPL	PB	
Intermittent		IPL		

period of time. During the duration of a block type activity, work is being performed throughout the occupied area. Other activities cannot be performed in this area until the entire activity is complete. A good example of a block type activity would be an earthwork operation, since, the work covers a significant area and the work activity moves back and forth along the area instead of along a path as in a linear activity. Figure 9 shows an example of a block activity. Block activities can be either full-span or partial-span, as suggested by Table 2, depending upon the area of the project occupied. Block activities are represented graphically as rectangles.
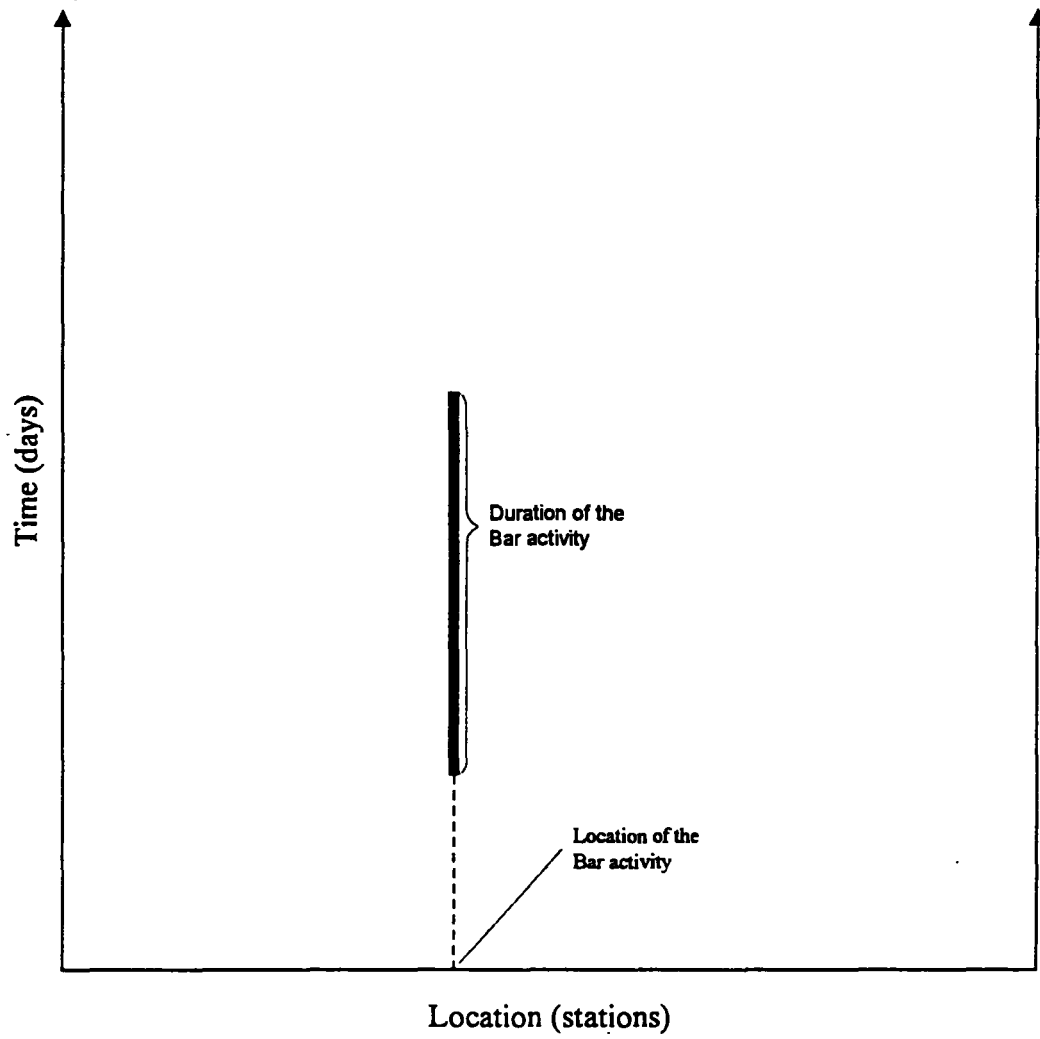
Bar Activities

A bar activity, represented by a vertical bar on the linear schedule as shown in Figure 10, represents an activity or group of activities that occur at a particular location. The area occupied by the bar activity is small with respect to the area of the project. Typical activities represented by bars on a linear schedule include bridges, culverts, and storm sewer intakes. For example, a bar activity could represent the construction of a bridge on a highway construction project. The construction of the bridge could very well be scheduled using another method such as CPM. The length of the bar on the linear schedule would indicate the time during which this activity affected activities on the highway construction project. There may be subsequent activities on the bridge project that do not affect the highway construction and would not be included in the length of the bar activity on the linear schedule. A bar activity could also represent the intersection of activities on another linear schedule.
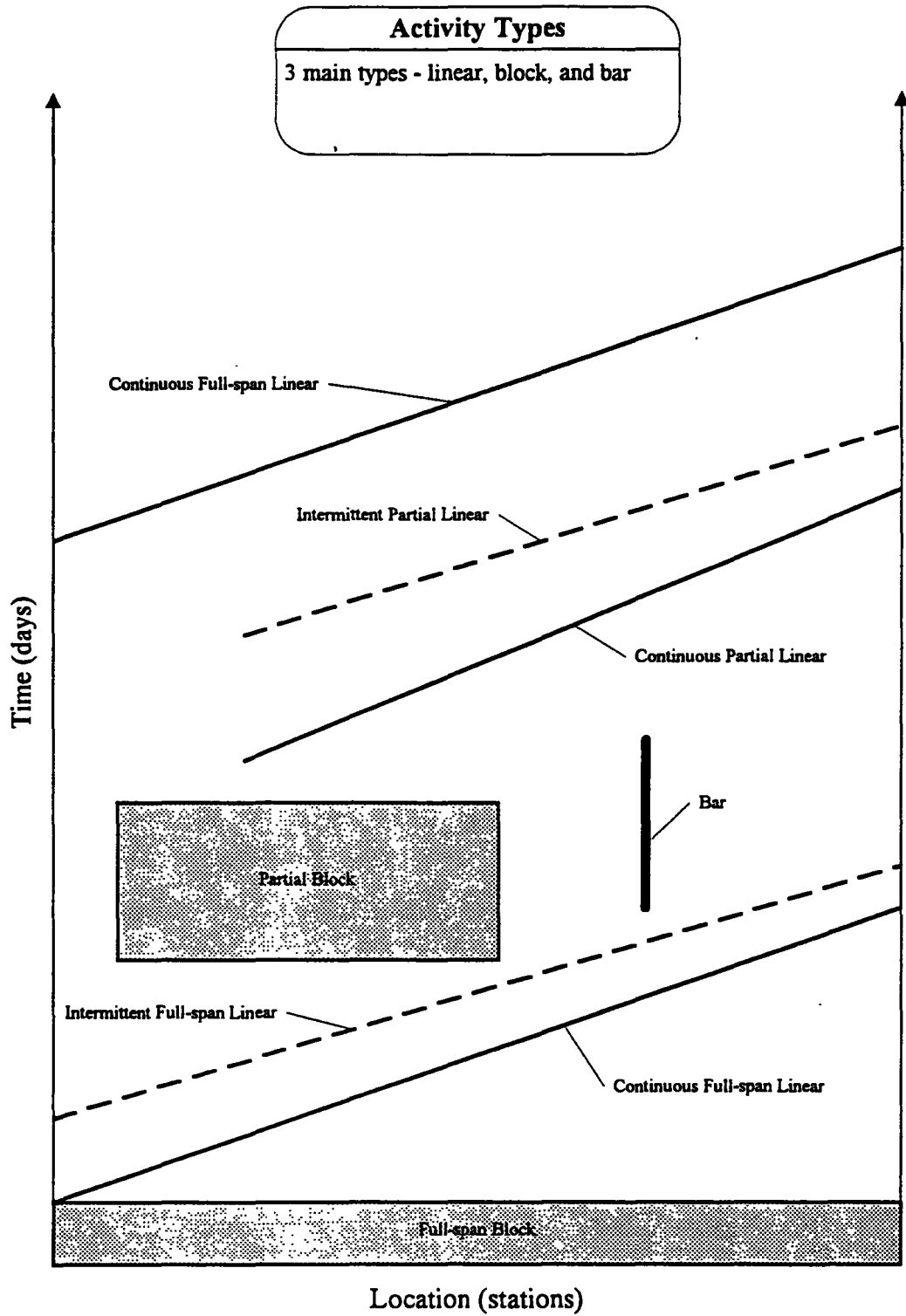
Figure 11 shows an example of all seven activity types and how they would appear on

Figure 9 A Block Activity

Figure 10 A Bar Activity

Figure 11 Seven Activity Types

a linear schedule.

Symbols and Milestones

A linear schedule can contain symbols or milestones. These are not usually thought of as activities as described earlier. A milestone does not have a duration but rather marks the time, and possibly the location, of a specific event. For example, a horizontal line across a section could indicate the beginning or end of a lane closure. A symbol, like a milestone, will place some event in time. A symbol can also indicate the location of an event. An example of a symbol, would be a small object, maybe a flag, at a location on the linear schedule to represent time and location of the arrival at the site of the concrete batch plant.

Other Graphical Constructs

Besides axes and activities, linear schedules may include other graphical entities such as a title block, a legend, a project plan view, or a mass diagram. These entities provide information to help the user understand the linear schedule or they provide landmarks to help the user orient the linear schedule with the physical characteristics of the project. Figure 12 shows an example of a project plan showing physical landmarks on the project and a mass diagram for an earthwork diagram. This helps the user of the linear schedule link work activities on the schedule to physical characteristics of the project such as landmarks and staging without referencing station numbers. For example, on a complex project with several stages, individual schedules could be created with a project plan view identifying the work involved in each stage. Visual cues such as these are important in helping field personnel and others not involved in the planning process understand the contractor's plan.
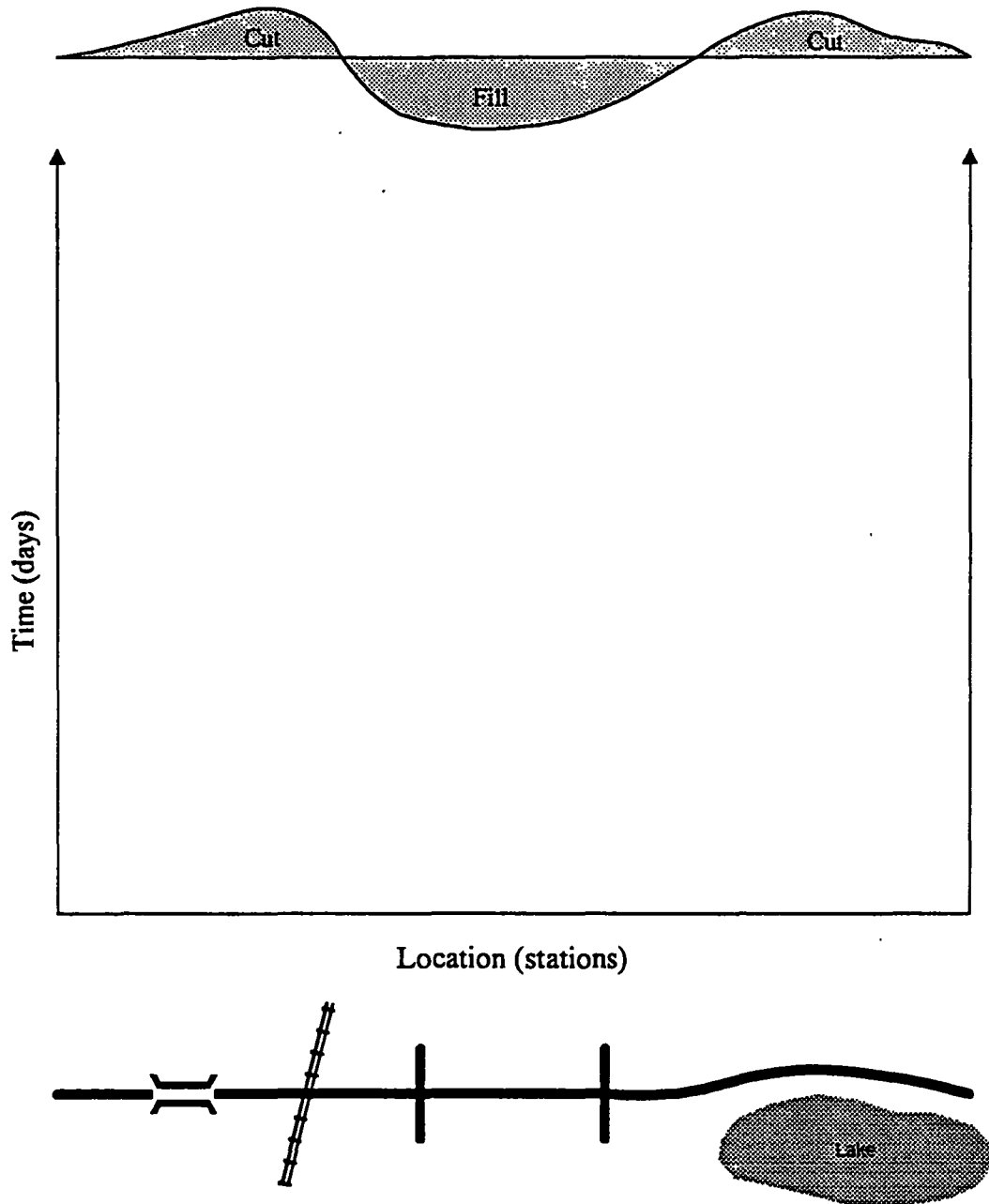
Time (days)

Location (stations)

Figure 12 Other Graphical Constructs

Developing the Project Plan

To construct a plan for a project using a linear schedule, the planner first establishes the planning field. This is accomplished by allocating the x-axis some measure of location on the project and the y-axis the time during which the project is to occur. This marks out a two dimensional visual planning area into which all of the activities involved in the completion of the project will be placed. Each activity placed on the linear schedule, physically occupies a space determined by the activities location and time attributes. The planner begins by placing activities on the linear schedule to represent how the project is expected to be built. As the plan develops, adjustments to activities based on location or time constraints can easily be made since all of the necessary information is visually available to the planner. The planning process, using a linear schedule, simply involves visually placing the activities onto the planning field to find the most efficient organization of activities to complete all of the work included in the project.

Complex Project Scheduling

Many highway construction projects today, especially those that require schedules, are complex projects which involve multiple stages. In an effort to reduce the impact of highway construction on the motoring public, projects are typically constructed while maintaining a limited flow of traffic through the project. The maintenance of traffic flow requires multiple stages of the work to make specific lanes and intersections available to traffic. The goal of the planner, either in the design phase or the construction phase of a project, is to develop a plan that balances the need to accommodate traffic and the need to efficiently construct the project.

The capabilities of the linear schedule to visually represent the project plan, greatly enhances the planner's ability to efficiently develop plans for highway construction work involving multiple stages or phases. The planner, however, must be creative in how the stages of the work are represented on the linear schedule or schedules. The as-built schedule in Figure 8 shows a schematic representation, below the x-axis, of the lanes under construction in this stage of the project.

Multiple linear schedules could also be used on highway construction projects that have intersecting work paths. The activities on a linear schedule that intersect another work path would appear as bars on the intersected linear schedule. In this manner, the activities in each schedule will reflect the influence of the activities in the other schedule at the point of intersection.

Controlling Activities

A critical element in the acceptance of linear scheduling as a viable tool in project planning and management, is the ability to determine a set of controlling activities. The research conducted with IDOT shows that the ability to determine controlling activities is essential for evaluating "charged" working day claims. The determinant factor in whether the contractor is charged a day is the contractor's ability to perform work on activities on the controlling activity path. The identification of a set of controlling activities is also important for the acceptance of linear scheduling in the construction industry and academia. At the heart of CPM scheduling, is the ability to determine a critical path through a logical set of activities on a project. To compete in the construction industry, linear scheduling must be able to

provide a synonymous set of activities as those calculated by CPM scheduling techniques. This section will present a method for determining controlling activities from a graphically represented linear schedule.

The *controlling activity path* for a linear schedule will be developed through a series of cases and examples. Terminology specific to linear scheduling will be defined, and key terms will be italicized. A dictionary of linear scheduling terms is included in Appendix A.

Activity Sequence List

In CPM scheduling, a network of activities is established by creating logical relationships between the activities on the project. A large building construction project can have several hundred activities. Although, there are certain activities which must occur in a logical sequence (e.g. the footings need to be constructed before the foundation walls, which rest on the footings, can be constructed) there are a significant number of activities that the planner can choose the order in which they occur. For example, do the windows and the roof have to be completed before the drywall activity can start? In a climate where it is likely to rain, the scheduler would usually agree that this is the proper sequence to reduce the risk of water infiltration causing damage to the drywall. However, in an arid climate, the scheduler may decide that the risk of starting the drywall activity before the building is fully closed is low, and subsequently base the activity logic on this decision. The order in which activities are scheduled to occur can be dependant on a number of factors other than the hard logic construction sequence such as external conditions, amount of resources, or material availability.

Unlike typical building construction projects, the sequence in which activities must occur on a highway construction project is much less flexible. The reasons for this are:

1. A typical highway construction project has a limited number of activities, usually less than 20.

2. The logical relationship between activities is quite rigid, or hard. For example, the old concrete must be removed before the grade can be prepared, before the subbase can be laid, before the new pavement can be poured.

3. The hard logic activities make up the majority of the work on a typical highway construction project.

These facts simplify the logical relationships between activities on a highway construction project. The planner is more involved in determining the location where work will occur rather than the sequence in which it will occur.

A sequence in which the activities will occur must first be established to determine the controlling activities in a linear schedule. This ordering of activities defines the *activity sequence list*. The *activity sequence list*, while not location dependant, must describe the order in which activities will occur at any location on the project Figure 13 shows an example of a linear schedule comprised entirely of *continuous full-span linear* (CFL) activities The order in which these activities are scheduled to occur, at any location on the project, is simply A, B, C, D, E, F, G, as shown in Figure 14.

Although, the activity sequence for the linear schedule in Figure 13 is easy to describe, this would be a very unusual highway construction project, since most projects will have

Sample Linear Schedule #1

All activities are continuous full-span (CFL) activities

Time (days)

G

F

E

D

C

B

A

Location (stations)
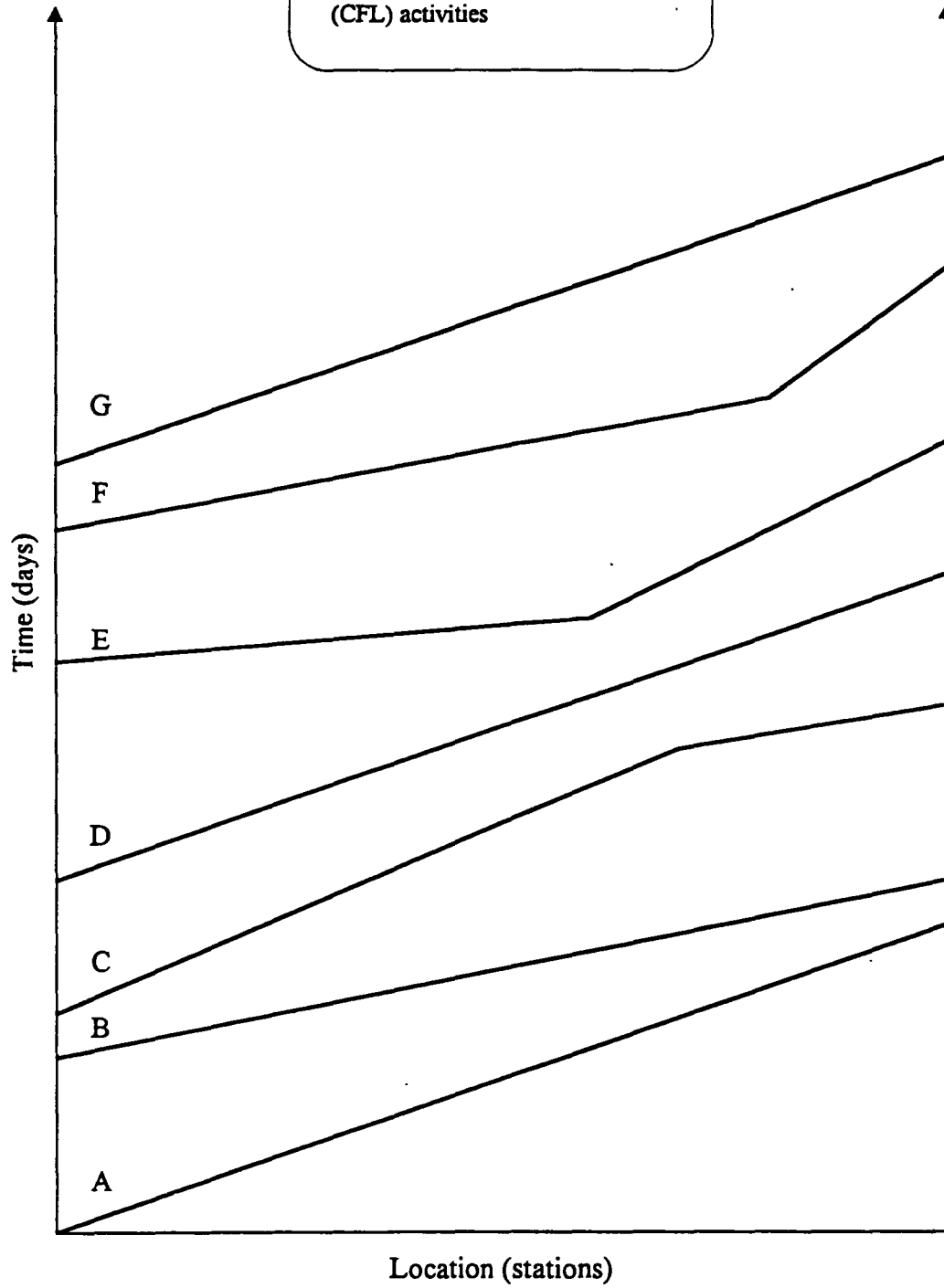
Figure 13 Example Linear Schedule

```
┌─────────────────────────────────┐
│      Activity Sequence #1       │
├─────────────────────────────────┤
│ All activities are continuous full-span │
│    (CFL) activities             │
└─────────────────────────────────┘
```

| Start | → | A | → | B | → | C | → | D | → | E | → | F | → | G | → | End |
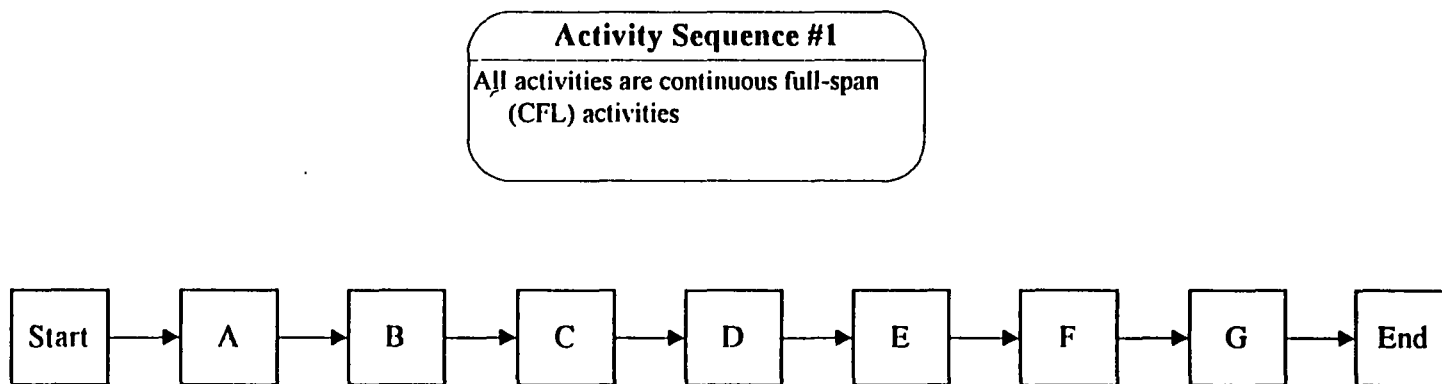
Figure 14 Activity Sequence

several other activity types besides just CFL activities. Figure 15 shows an example of a more

typical highway construction project.linear schedule. This project has three CFL activities, A,

C, and G. Also assume that there is a "0" time CFL activity at the beginning and the end of

the project. These activities are called "Start" and "End." *Intermediate activities*, any

activities that are not CFL activities, will always lie between two CFL activities. For example,

activity B, a partial-span block (PB), and activity C, two bars (B), lie between CFL activities

A and D. The *activity sequence list* must describe the activity sequence through these

activities, regardless of location.

Figure 16 shows the logical sequence of activities for this project. Notice, that for the

CFL activities **Start, A, D, G,** and **End,** there cannot be multiple logical paths since these

activities span the entire project. At intermediate locations between CFL activities, however,

there may be multiple paths. The possible activity sequences can be determined by examining

the order of activities at any possible location (vertical line) between the CFL activities. The

vertical dashed lines on Figure 15 show the possible activity sequences through this sample

linear schedule. The five possible paths are schematically represented in Figure 16 and the

activities involved in each path are listed. As in CPM scheduling, the goal is to find the

longest continuous path through this sequence of activities. This path will define the

controlling activity path and determine when and where activities are controlling.

Continuous Full-Span Linear Activities

The next six cases will deal only with *continuous full-span linear* (CFL) activities,

since these are the primary activities that define the method of scheduling. The technique to

52

Figure 15 Example Linear Schedule

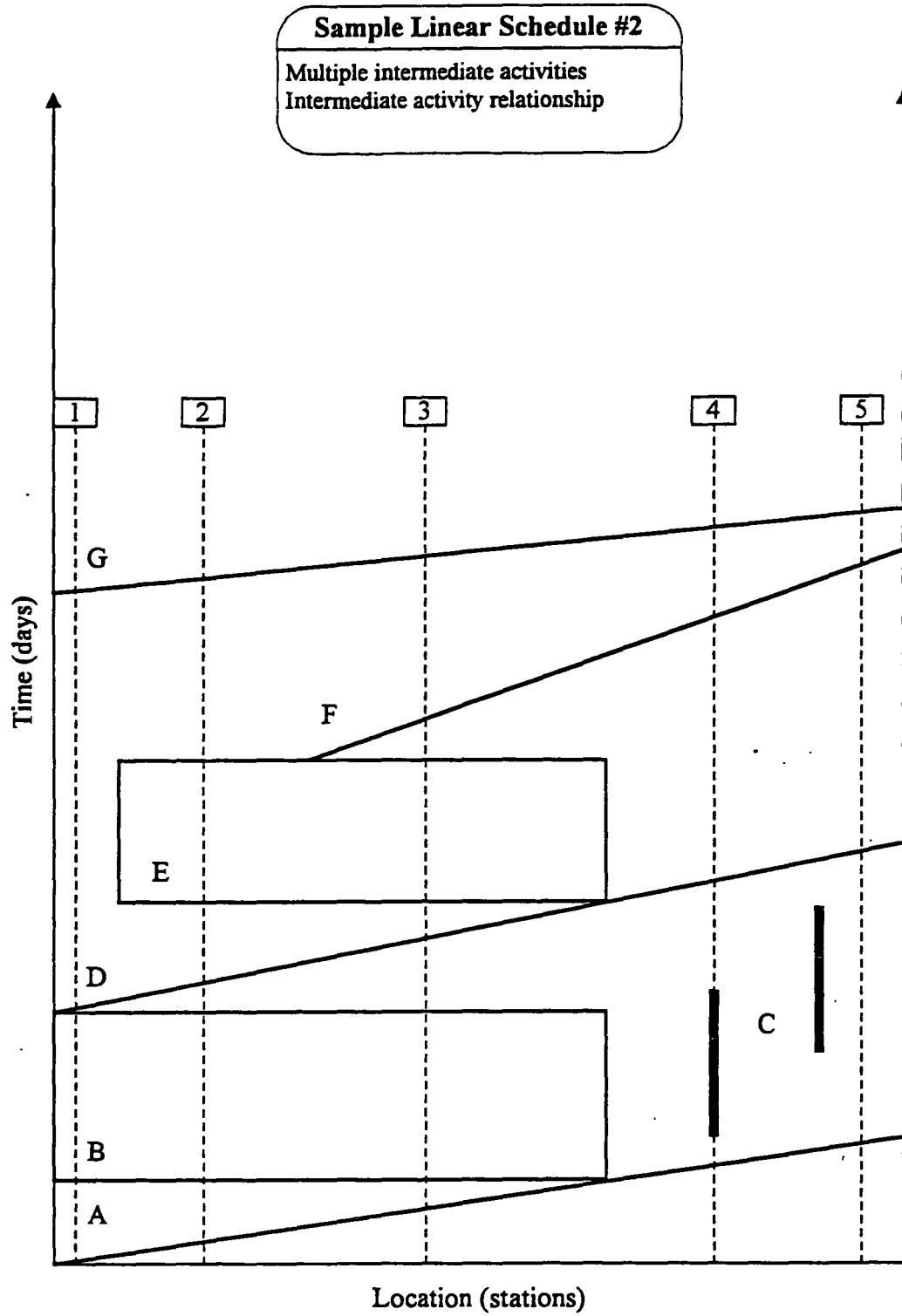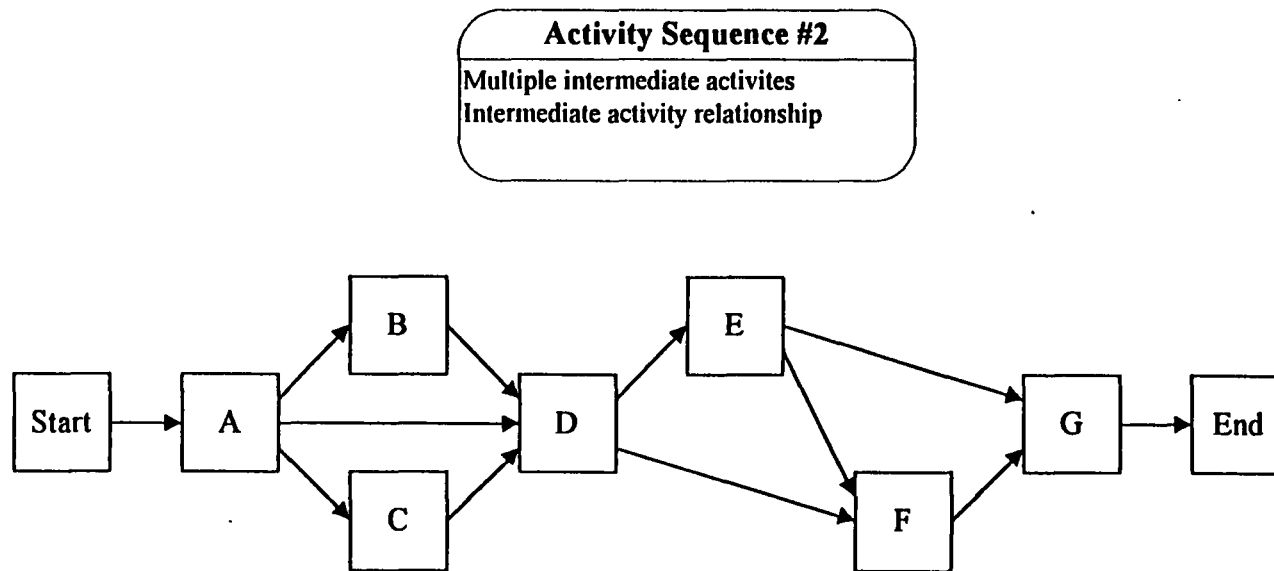**Activity Sequence #2**

Multiple intermediate activites
Intermediate activity relationship

Possible Activity Sequences

Sequence 1 - Start-A-B-D-G-End
Sequence 2 - Start-A-B-D-E-G-End
Sequence 3 - Start-A-B-D-E-F-G-End
Sequence 4 - Start-A-C-D-F-G
Sequence 5 - Start-A-D-F-G

Figure 16 Activity Sequence

identify *controlling activity segments* and the *controlling activity path* will first be developed using only CFL activities. After the technique has be fully explained with CFL activities, other types of activities, defined earlier as *intermediate activities*, will be added to the schedule. By the end of the cases, the *controlling activity path* will be identifiable using any combination of the activity types described in the previous section.

<u>Case 1</u>

Each case presented here will have a corresponding set of figures. Figure 17 is the initial figure for the first case. Notice that there are two *continuous full-span linear* (CFL) activities represented on the simplified *linear planning area*. The title block indicates that there are no *intermediate activities* which means that only CFL activities are included in this case. The title block also indicates that the activity defined as the *target activity* has a higher production rate than the activity designated as the *origin activity*. The slope of the target activity is flatter than that of the *origin activity*. This indicates that the *target activity* covers the same area as the *origin activity* in less time, therefore, the production rate is higher.

There are two major steps to perform in determining the controlling activities in a linear schedule. The first step is to determine which portions of each activity could potentially be controlling. In each case, the activity for which the potential controlling segment is being determined is designated the *origin activity*. The *origin activity* will always be a *continuous full-span activity* (CFL), and the earliest point in time on this activity is designated as the *origin*. The next CFL activity in the activity sequence list will be the target activity. The *potential controlling segment* of the *origin activity* can be determined by examining the
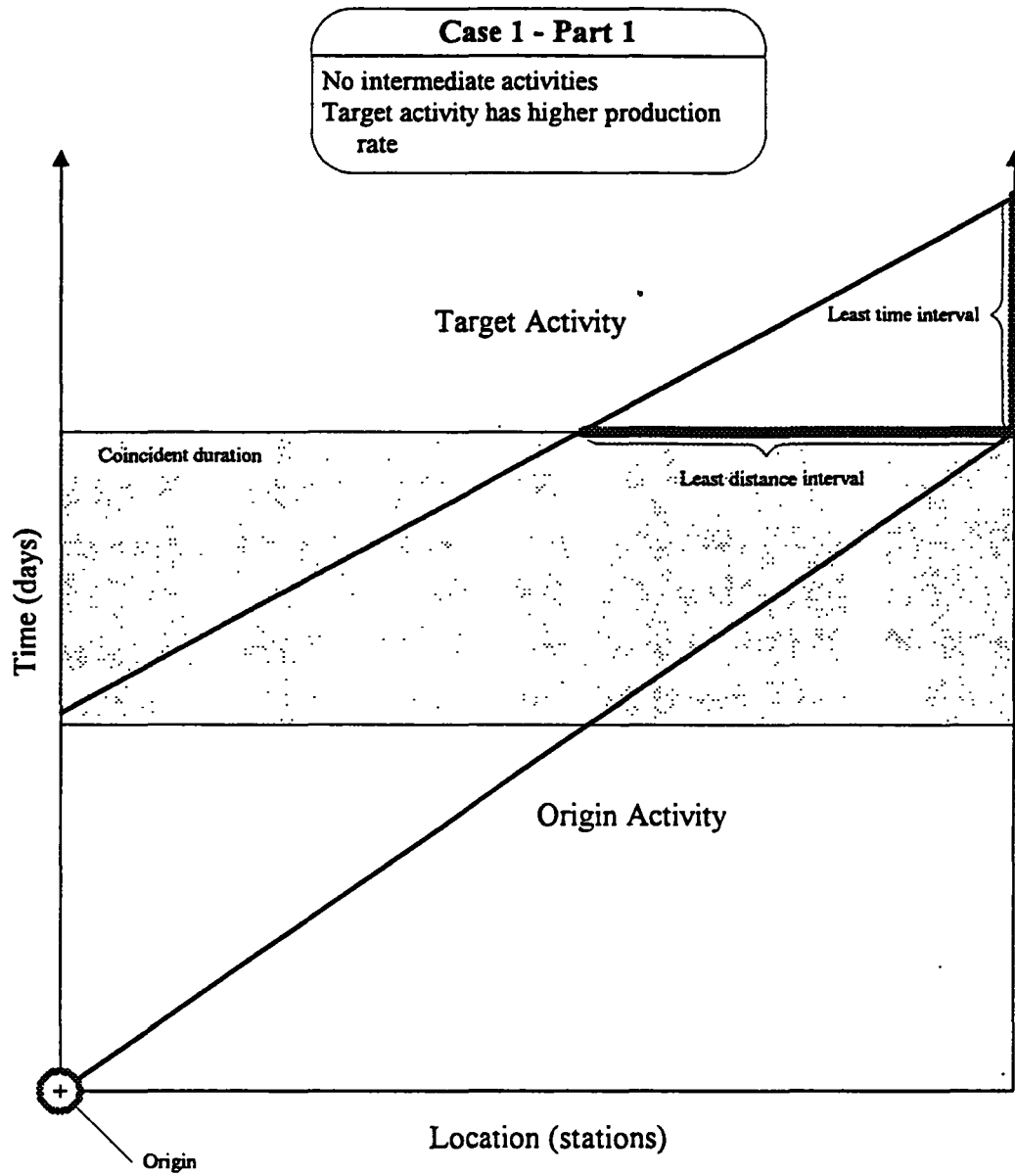
Figure 17 Case 1 - Part 1

relationship between these two activities.

Three specific elements must be determined to describe the relationship between these two activities. They are:

1. The *least time interval* (LT) - this is the shortest time interval between any two consecutive activities. *Consecutive activities* are activities that can be connected in time (vertically) without crossing another activity. The *least time interval* will always occur at a *vertex* of one of the activities in question. For a linear activity, vertices occur at the endpoints, or at any point where the rate of production changes. Vertices occur at the corners of box activities and at either end of a bar activity.

2. The *coincident duration* - This is an interval in time during which the two activities connected by the *least time interval* are both in progress.

3. The *least distance interval* - this is the shortest distance between the two activities that lies within the coincident duration interval and intersects the least time interval.

The *least time interval* will always be a distance measured vertically between activities since it is a measure of time. Because the *least distance interval* is a measure of distance, it will always be measured horizontally on the linear schedule. The *least time interval* is significant because it describes a point where the two activities are closest to each other in time. In a sense, this is a point where the activities actually touch each other. The reason that linear activities usually do not touch each other is because the production rate of any

particular activity has some amount of variability. The line indicating the linear activity is an estimate of the average anticipated production rate of that activity. The space between the activities can be viewed as a buffer which can absorb this normal variation in production rate without conflicting with subsequent activities. The larger the anticipated variation in production rate of any particular activity, the larger the buffer should be.

In this step of determining *controlling activities*, the *origin activity* is always viewed as if it were the first activity in the *activity sequence*. Therefore, the segment from the origin up to where the target activity begins will always be a *potential controlling segment*. Somewhere during the time after the target activity begins and the origin activity ends, there is a link along which the *controlling activity path* must occur. The *controlling activity path* is defined as the continuous path of longest duration through the project, and defines the sequence of activities that must be completed as planned to finish the project within the overall planned duration. It is logical to assume that this path will occur where activities are closest to each other. The *least distance interval* describes the location at which this closest point occurs.

Once the *least distance interval* has been determined, the point of intersection with the origin activity is called the *critical vertex* as shown in Figure 18. The segment of the *origin activity* between the *origin* and the *critical vertex* is then a *potential controlling segment* for this activity, and the *least distance interval* becomes a *potential controlling link* between the *origin* and *target* activities. The determination of which portion of the *potential controlling segment* is actually controlling will be described later. The actual controlling segment,
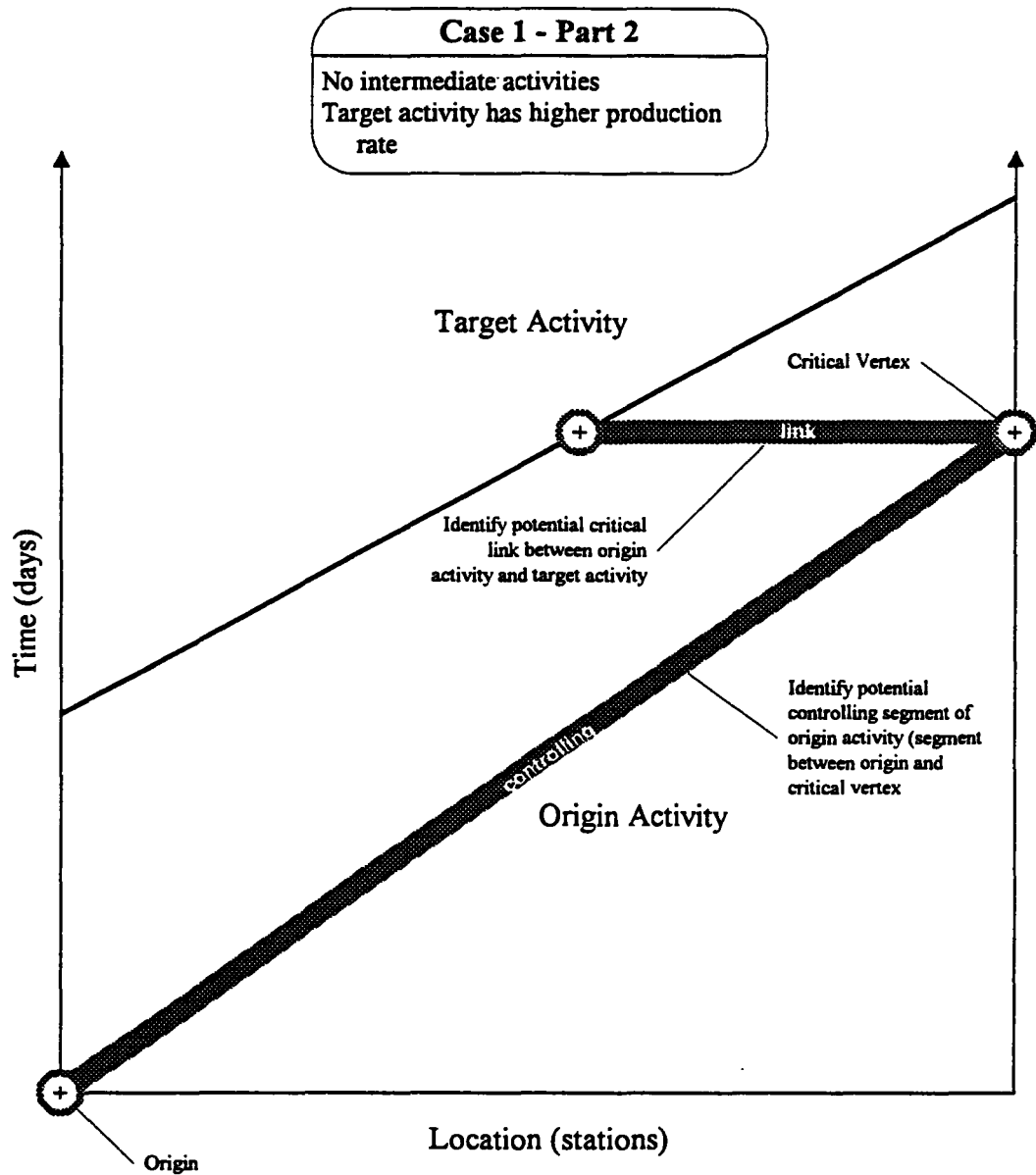
Figure 18 Case 1 - Part 2

however, will never extend beyond the segment that is determined to be potentially controlling in this step of the process.

Note, that in this case, the entire length of the origin activity has been determined to be a *potential controlling segment*. Since the *origin activity* progresses at a slower rate than the *target activity*, the end of this activity controls the end of the *target activity*. Remember that the *least time interval* can be viewed as a point where the activities touch, therefore, for planning reasons, this interval must be preserved. This means that the target activity could progress at a slower rate, if it started earlier. An example from the IDOT projects would be the concrete removal operation on the Johnson County project. Concrete removal is a low production rate activity followed by a grade preparation activity which can progress at a substantially higher rate. Therefore, while the concrete removal operation is in progress, it likely paces the subsequent activity and is probably a controlling segment.

Case 2

Case 2 is similar to the first case except that the *target activity* now has a lower production rate than the *origin activity* (see Figure 19). In this instance, the *least time interval* is at the beginning of the activities. The *potential controlling segment* is the portion of the *origin activity* in progress up until the target activity starts (see Figure 20). After the *target activity* starts, the *origin activity* could slow down without affecting the end time of the *target activity*.

Case 3

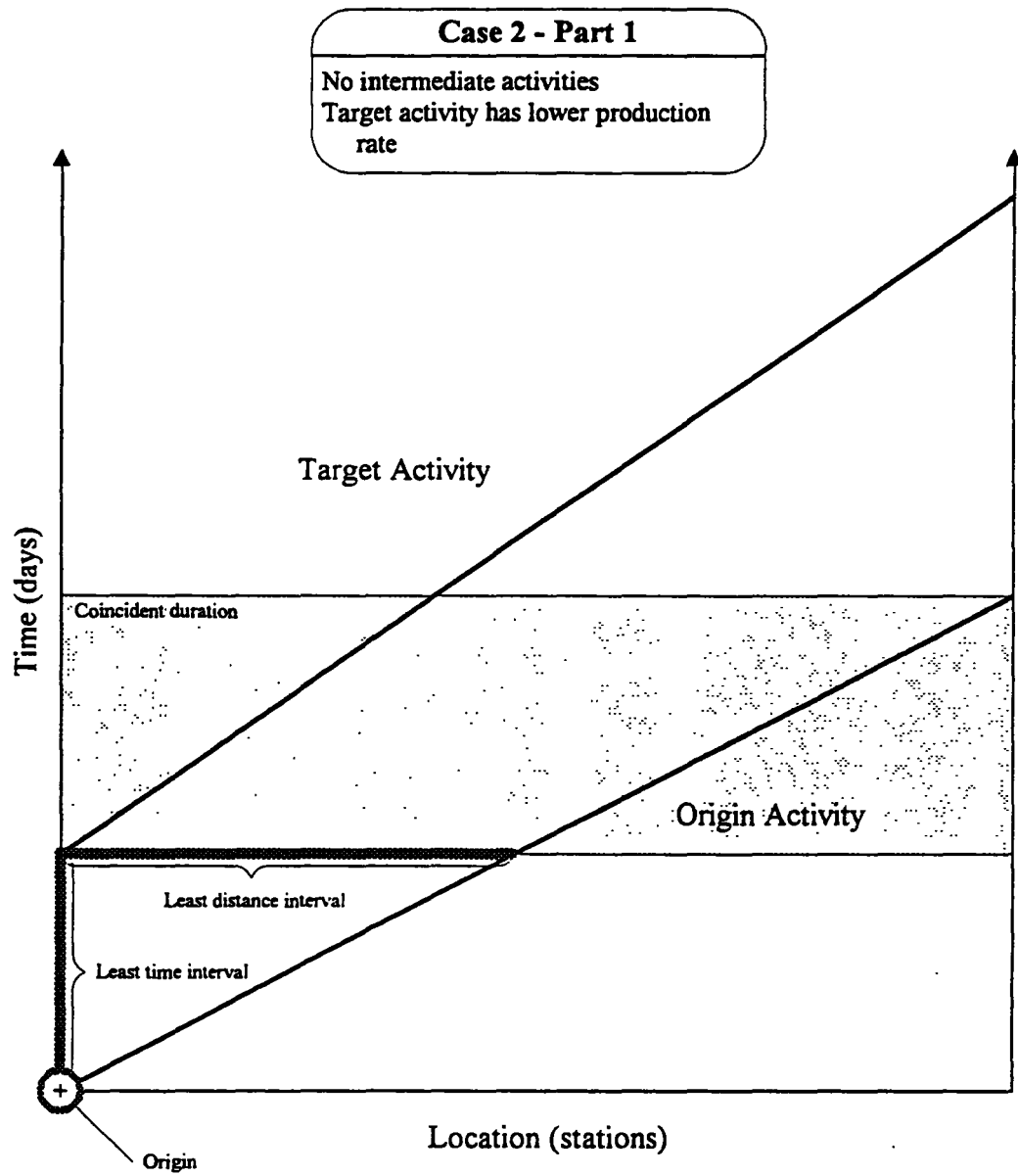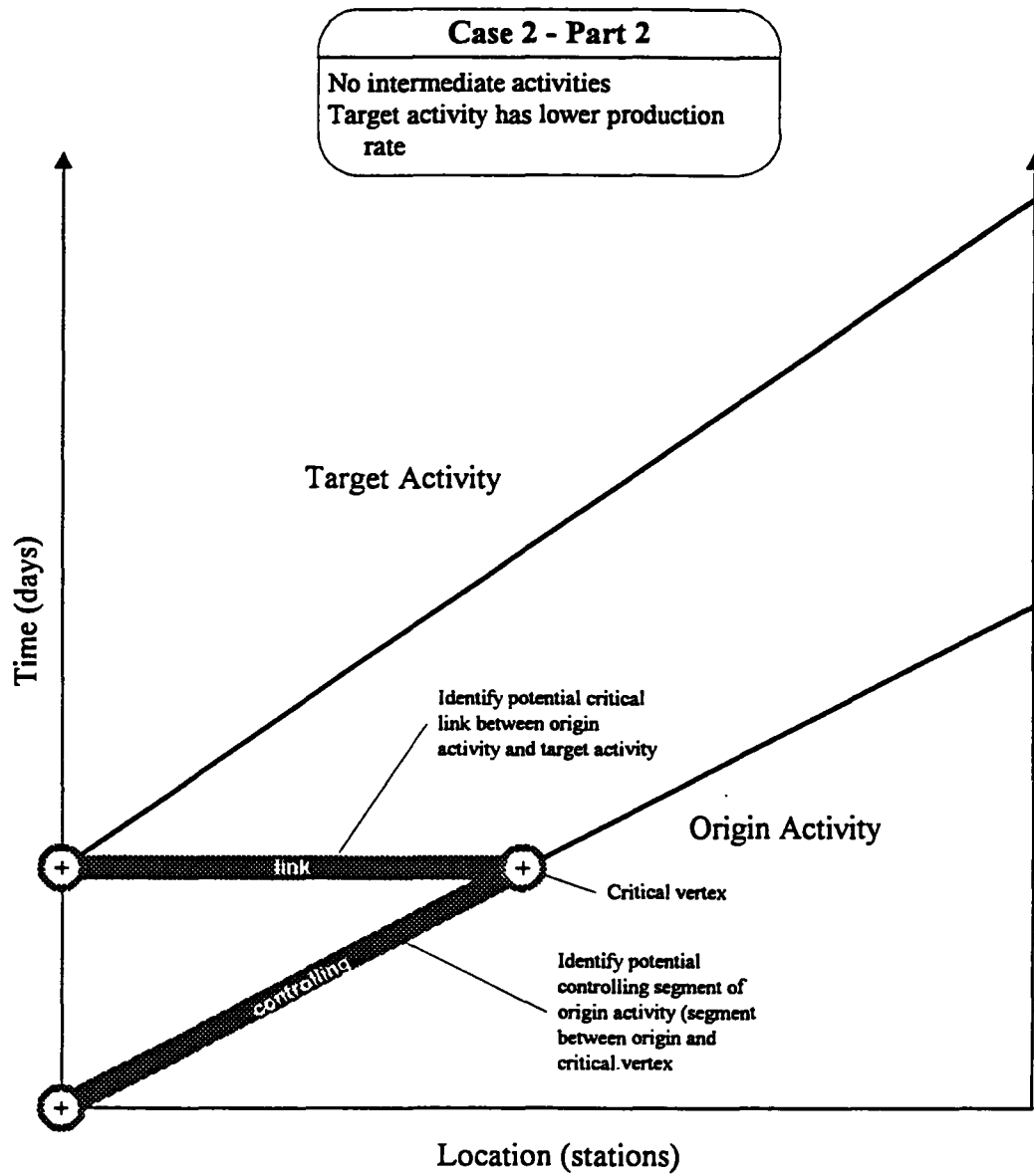In Case 3, the *target activity* has a variable rate (see Figure 21). The first portion of

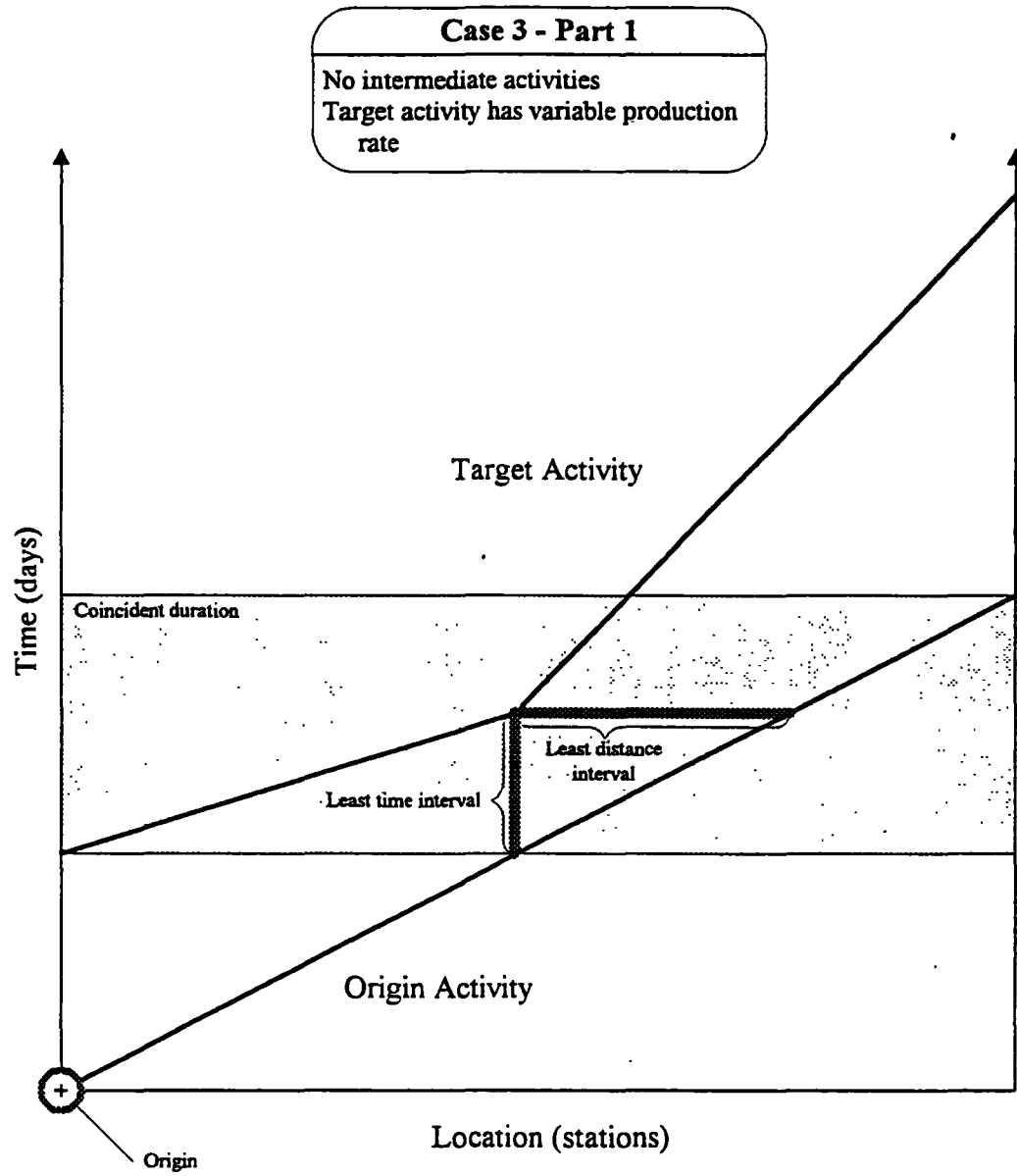Figure 19 Case 2 - Part 1

Figure 20 Case 2 - Part 2

Figure 21 Case 3 - Part 3

the activity has a higher production rate than the *origin activity*, while the second portion has a lower rate than the *origin activity*. This causes the *least time interval* to occur at the vertex formed by the change in production rate of the *target activity*. The *potential controlling segment* of the *origin activity* will then be the segment in progress up until the point that the *target activity* production rate becomes lower than that of the *origin activity* (see Figure 22).

Case 4

This case shows an example of where the *origin activity* has a change in production rate (see Figure 23). The first segment of the *origin activity* has a lower production rate than the *target activity*, and the second segment has a higher rate than the *target activity*. The *least time interval* occurs at the point where the rate changes in the *origin activity*. Notice, that the *least distance interval* does not occur at an edge of the *coincident duration* interval, as in previous cases. It does, however, fall within the *coincident duration* interval and it intersects the *least time interval*, so it meets the criteria of the definition. The *potential controlling segment* of the *origin activity* is the segment form the *origin* up to the point where the production rate increases in the *origin activity* (see Figure 24).

Case 5

This case is very similar to the previous case, except that the point where the production rate changes in the origin activity occurs prior to the start of the target activity (see Figure 25). Again, the *least time interval* occurs at the point where the production rate changes in the origin activity. In this case, however, the *least distance interval* does not intersect an endpoint of the *least time interval*, but meets the requirements of the definition by
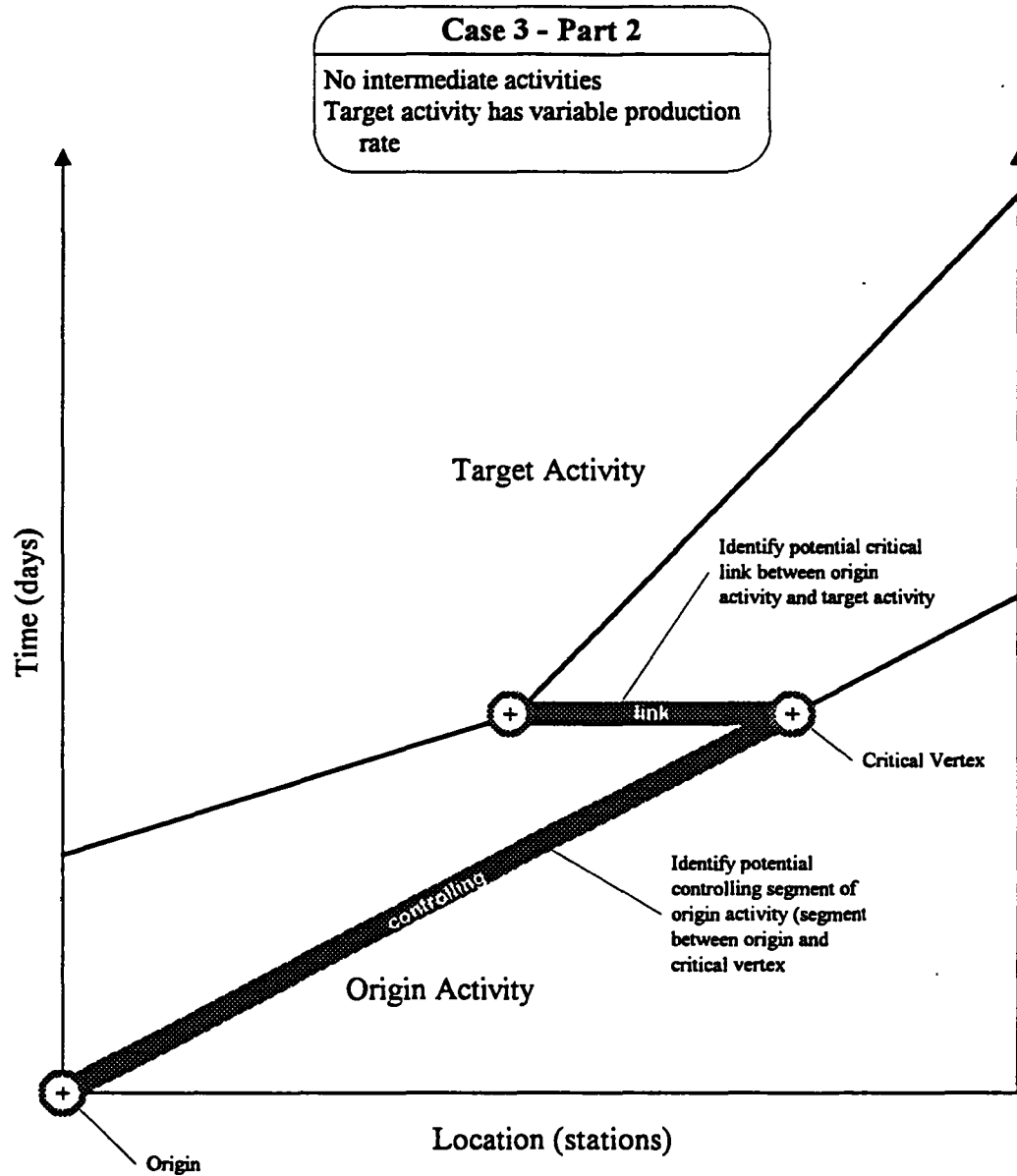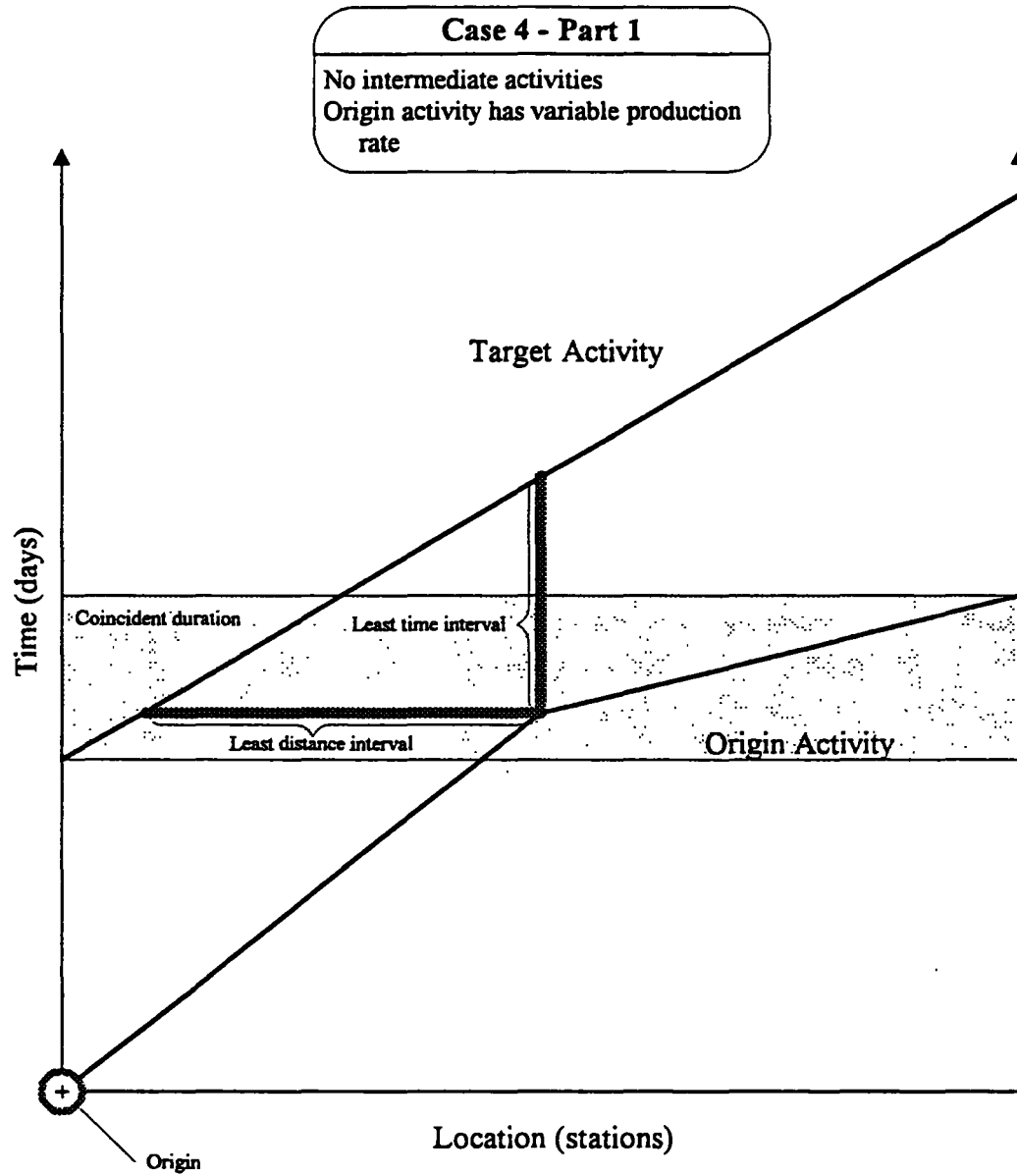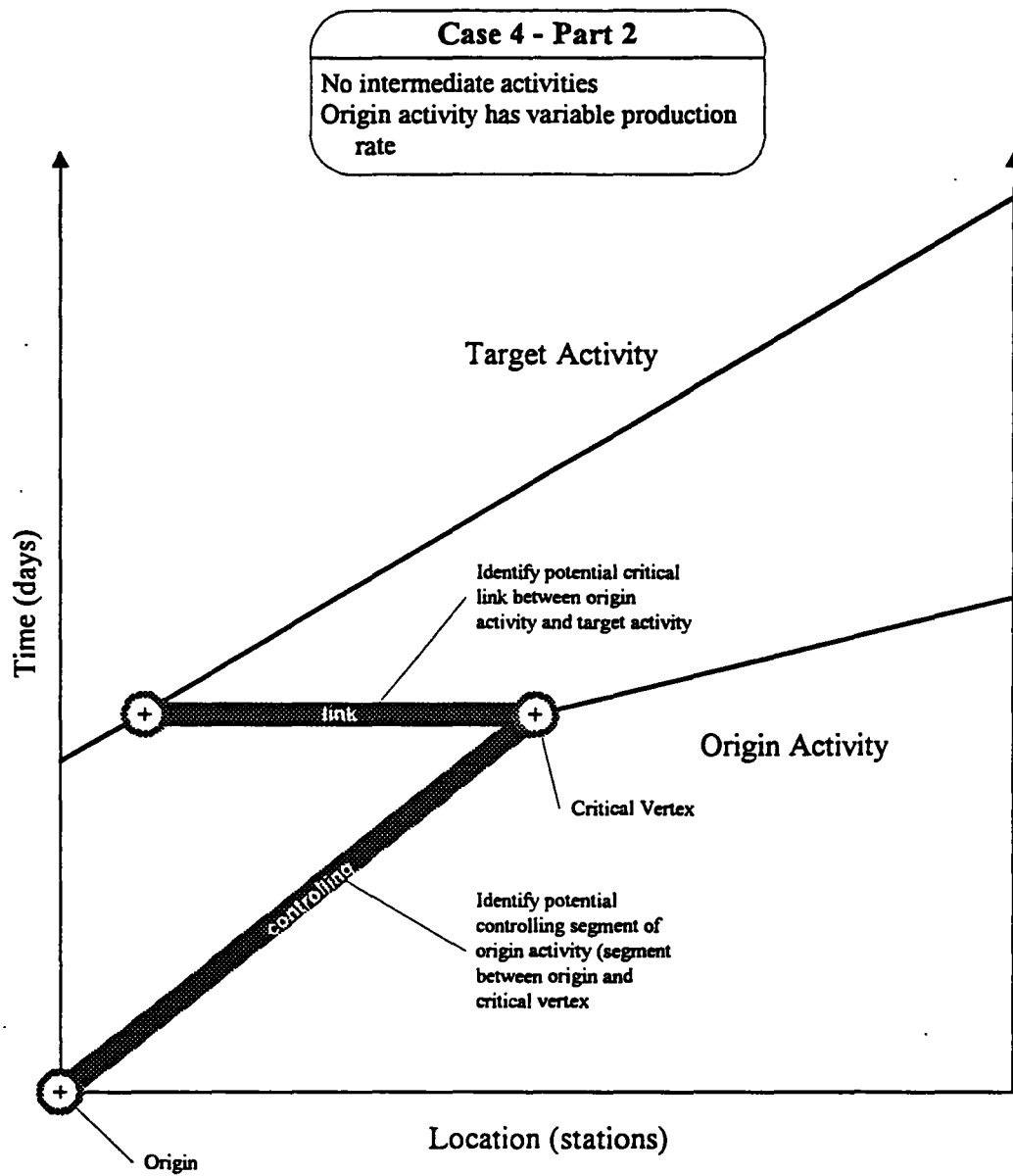
Figure 22 Case 3 - Part 2

65

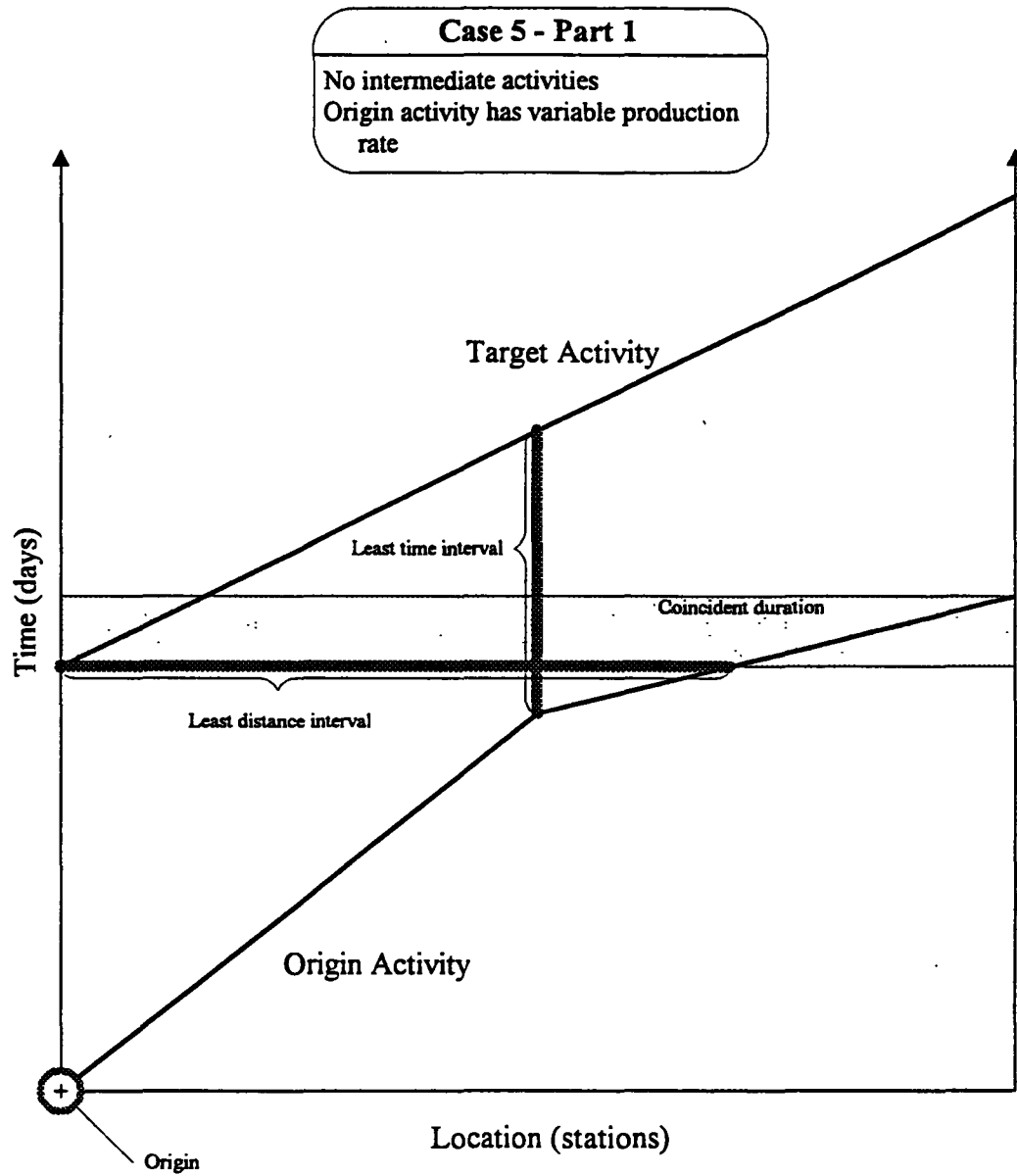Figure 23 Case 4 - Part 1

Figure 24 Case 4 - Part 2

Figure 25 Case 5 - Part 1

falling within the *coincident duration* and intersecting the *least time interval*. The *potential controlling segment* is the portion of the *origin activity* in progress up until the *target activity* begins (see Figure 26).

<u>Case 6</u>

In this case, the production rate of the *origin activity* again varies, but in the opposite direction as the last two illustrative cases. The first portion of the *origin activity* has a higher rate than the *target activity* while the second portion has a lower rate than the *target activity* (see Figure 27). The *least time interval* occurs at the beginning of the activities. The *least distance interval* does not represent the actual least distance between the *origin and target activities* within the *coincident duration* interval as it has in the previous cases. It does, however, meet the requirements of the definition by occurring within the *coincident duration* and intersecting the *least time interval*. The *potential controlling segment* is the portion of the *origin activity* in progress up until the *target activity* begins (see Figure 28) just as before.

Upward and Downward Passes

Cases 1 through 6 have shown the *potential controlling segments* possible from various combinations of origin and target activities, as long as only *continuous full-span linear* activities are involved.

The next step is to find a *controlling activity path* through a series of *continuous full-span linear* (CFL) activities. The next four figures, Figures 29 - 32, illustrate this process. Figure 29 shows an example of linear schedule using only CFL activities with fixed and variable production rates. The *activity sequence list* for this schedule is simply A, B, C, D, E,

Figure 26 Case 5 - Part 2

Case 6 - Part 1

No intermediate activities
Origin activity has variable production rate

Target Activity

Time (days)

Coincident duration

Least distance interval

Least time interval

Origin Activity

Origin

Location (stations)

Figure 27 Case 6 - Part 1

Case 6 - Part 2

No intermediate activities
Origin activity has variable production
rate

Target Activity

Time (days)

Identify potential critical
link between origin
activity and target activity

link

Critical Vertex

Identify potential
controlling segment of
origin activity (segment
between origin and
critical vertex

Origin Activity

controlling

Origin

Location (stations)

Figure 28 Case 6 - Part 2

72

Example 1 - Linear Schedule

All activities are full-span
All activities are continuous
Variable and nonvariable rates
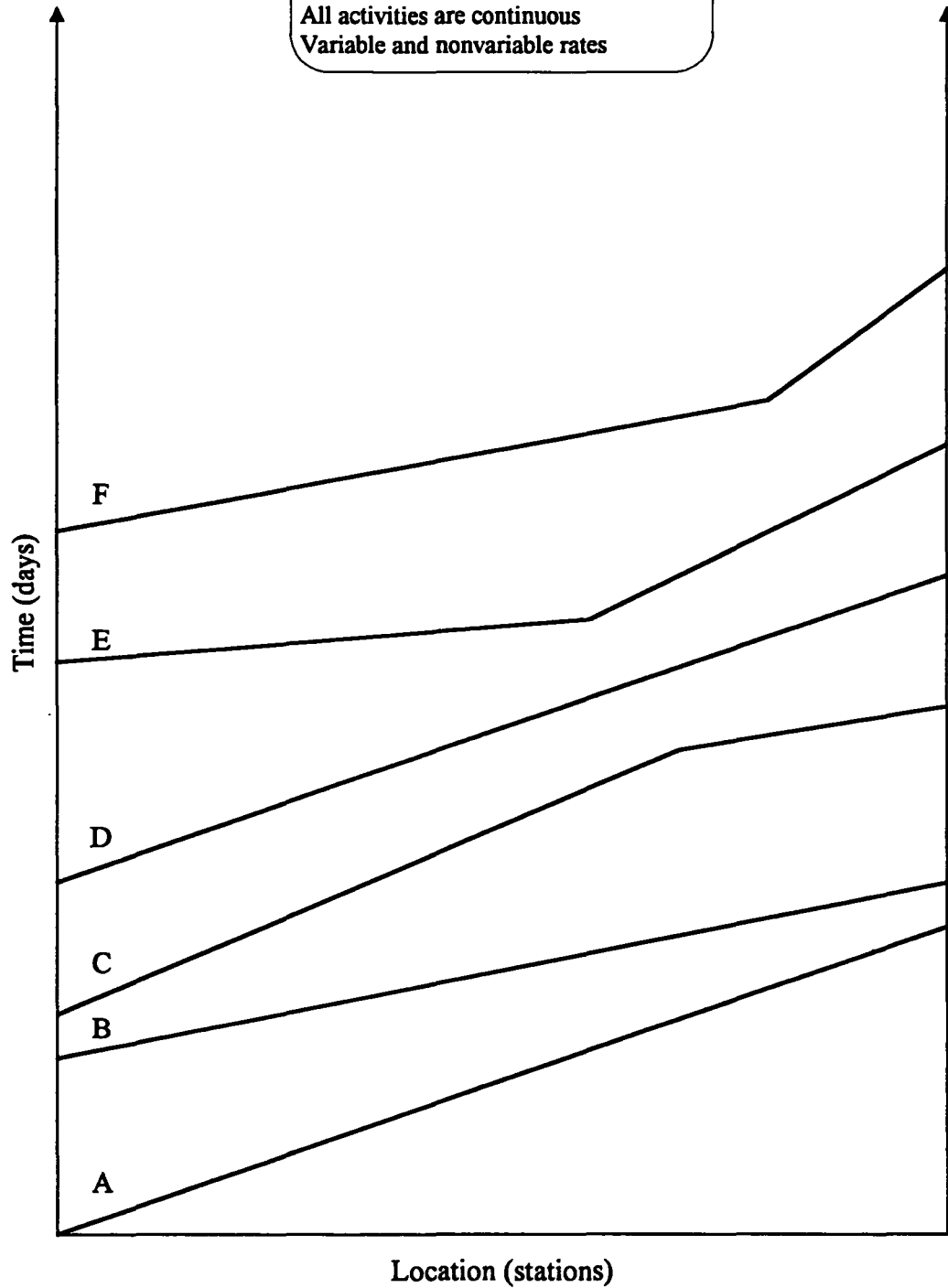
Figure 29 Example Linear Schedule

and F. The first step in finding the *controlling activity path* is to perform an *upward pass.* This is somewhat analogous to a forward pass in a CPM diagram. The term *upward pass* comes from the direction that this analysis moves through the linear schedule. The *upward pass* consists of finding the *potential controlling segments* as described in Cases 1 - 6.

Beginning with the first pair of CFL activities, **A** and **B**, the *least time interval* and *least distance interval* are determined as shown in Figure 30 The *potential controlling segment* and *potential critical link* can then be determined for the pair of activities as shown in Figure 31. This is the same process that was described in Cases 1 - 6. Repeat these steps for each pair of CFL activities in the activity sequence list **BC, CD, DE,** and **EF**.
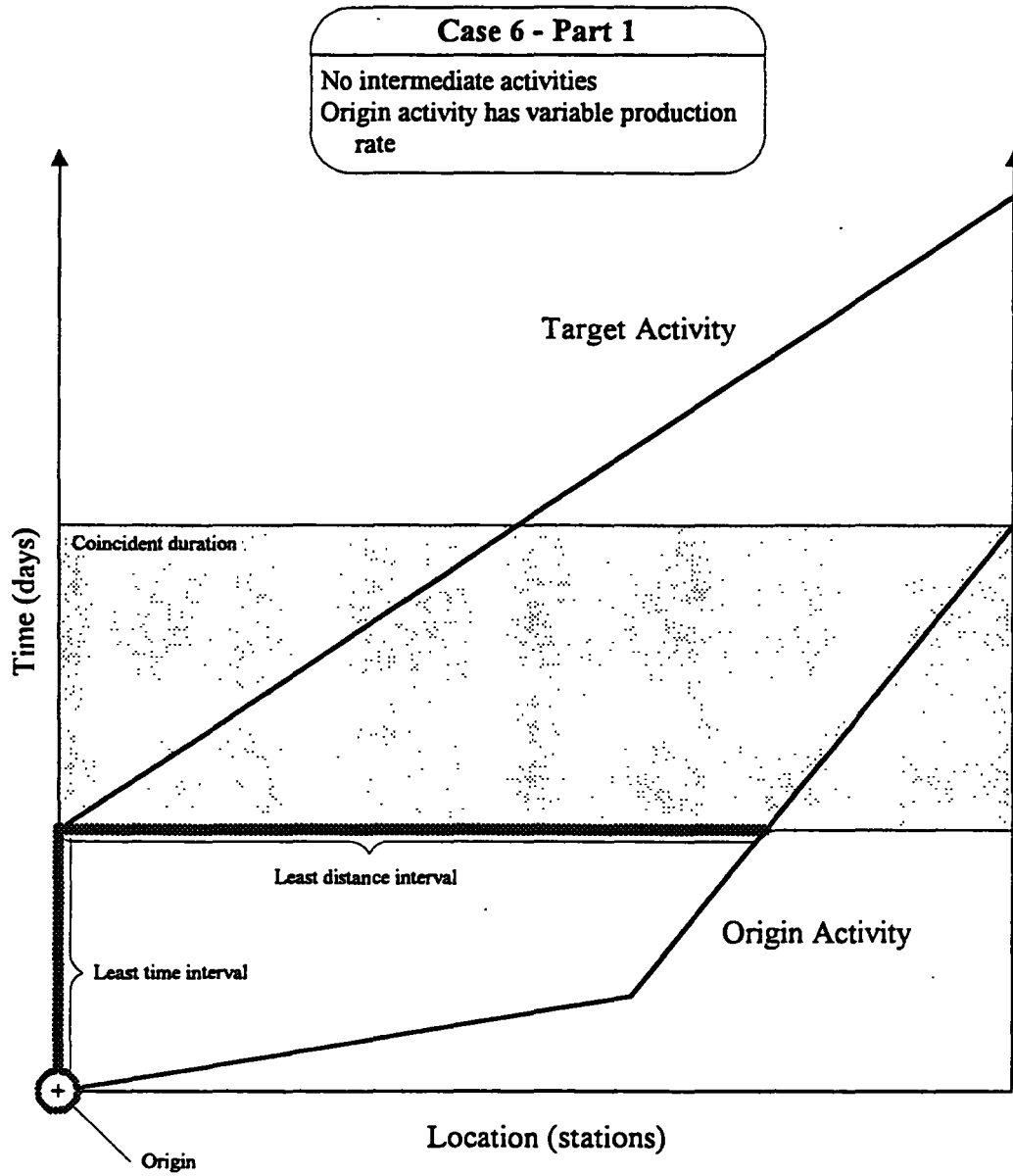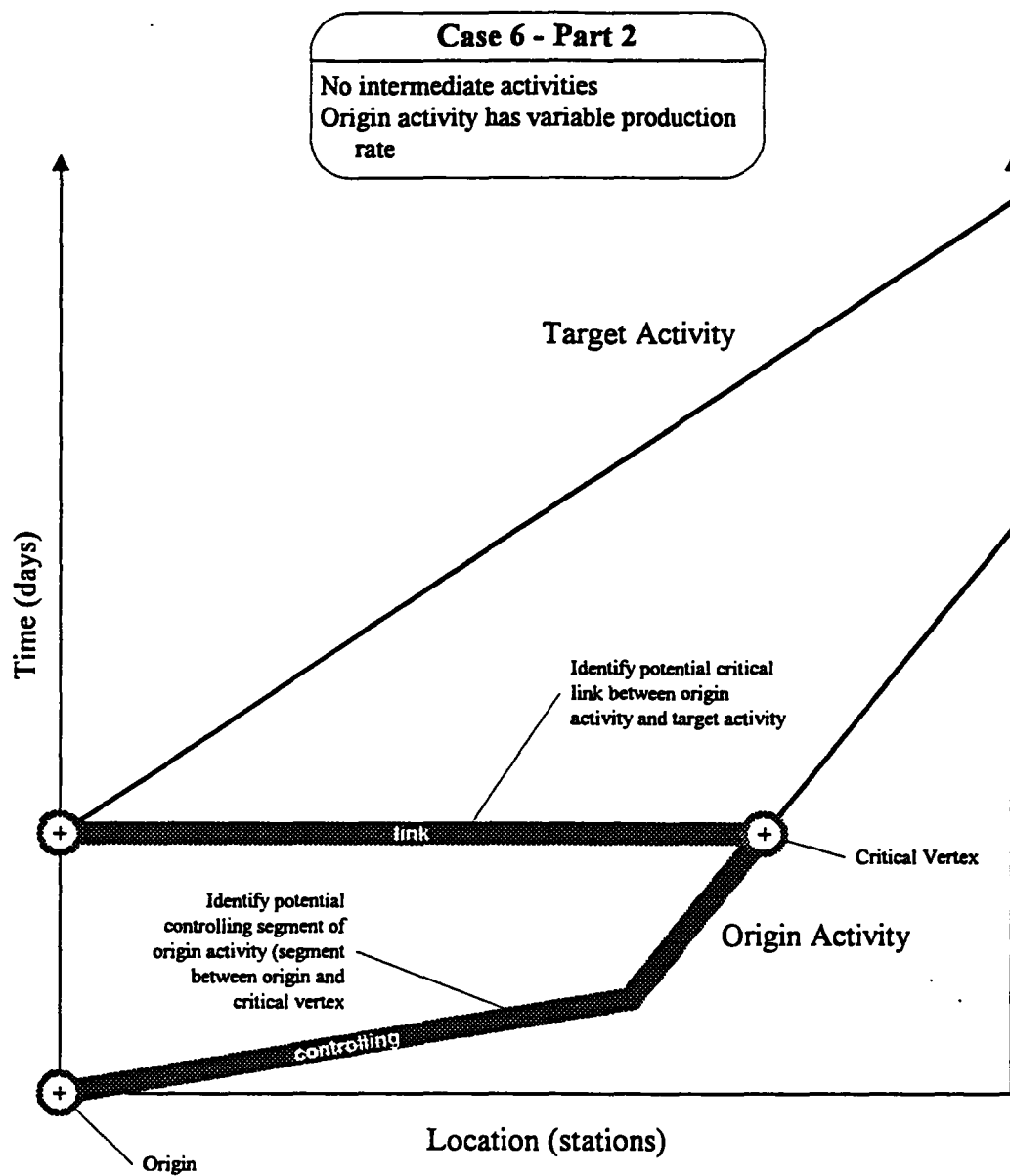
As discussed earlier, the Start and End activities are assumed to be zero time CFL activities. This means that their production rate is undefined and, therefore, they cannot be controlling segments. The Start and End activities are only used to define *potential controlling segments* of actual activities. This means that the entire length of activity **F** can be considered a *potential controlling segment* which completes the upward pass.

The next step is to perform the *downward pass.* The purpose of the *downward pass* is to determine which portions of the *potential controlling segments* are actually on the *controlling activity path*. This step can be compared to the backward pass used in CPM scheduling which identifies activities that do not have any float. In the case of linear activities on a linear schedule, the backward pass identifies segments of activities for which the production rate cannot decrease without extending the duration of the project. This also means that segments of activities not on the controlling activity path have *rate float.* An
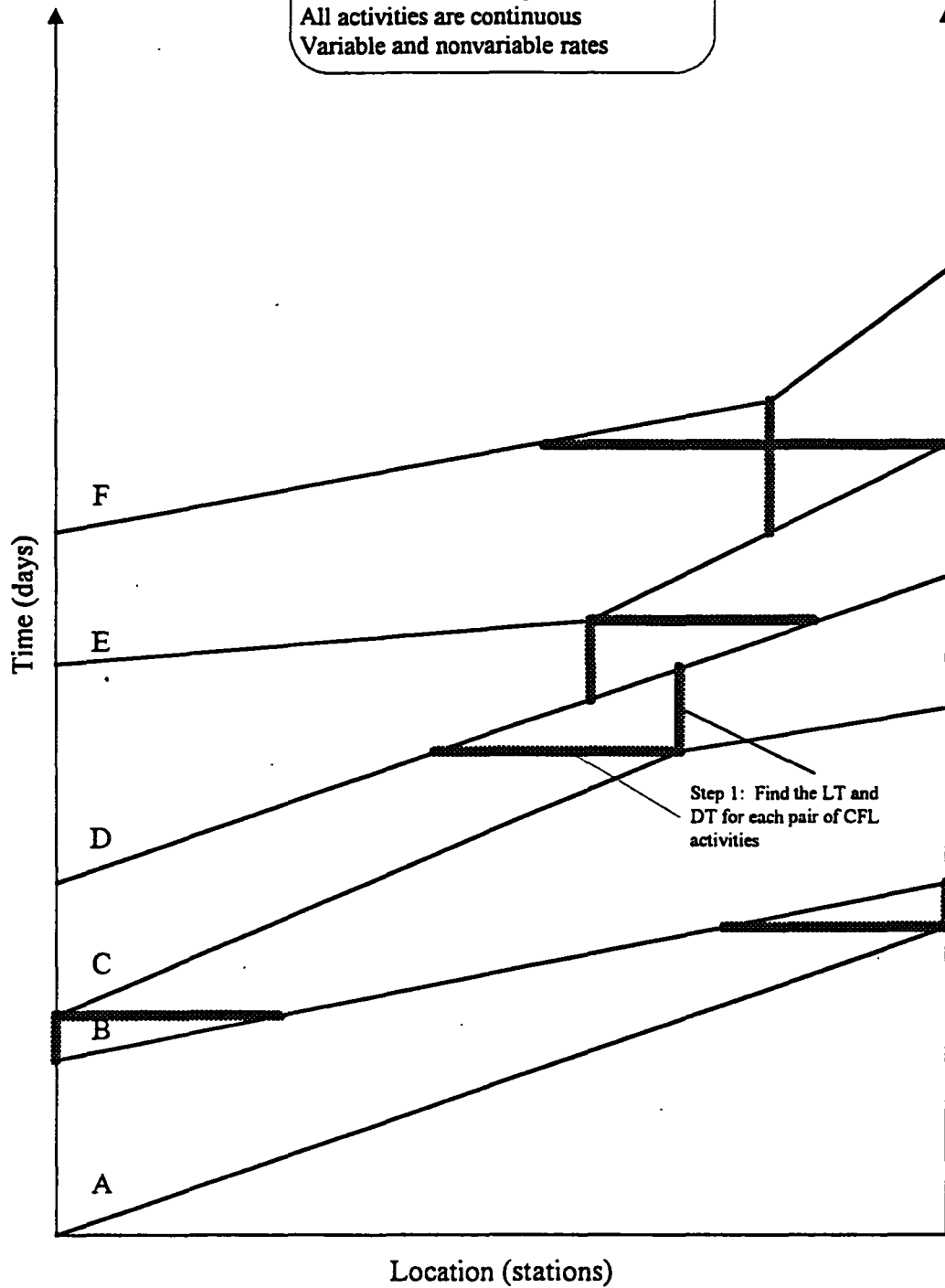
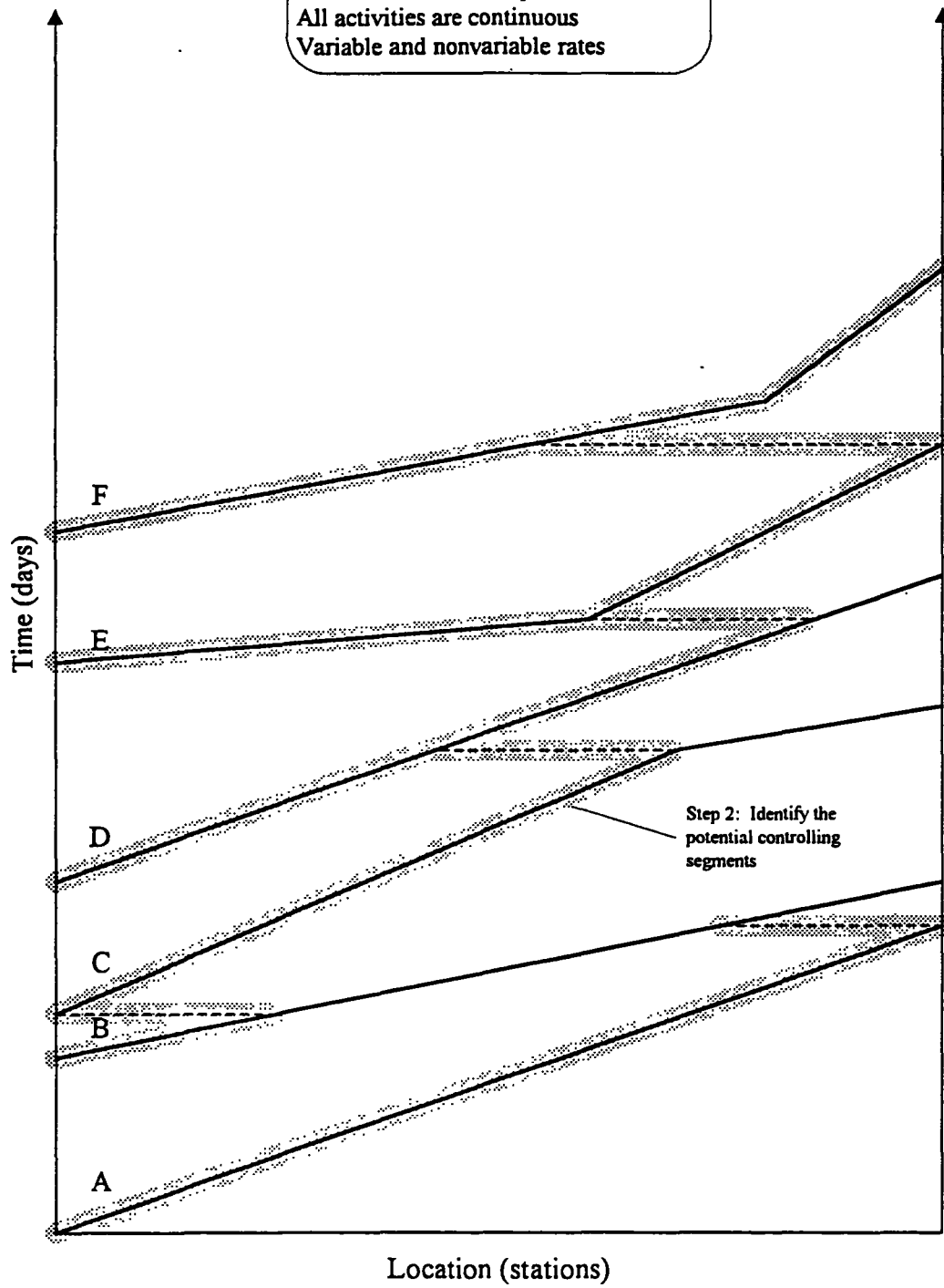**Figure 30** Upward Pass - Step 1

Figure 31 Upward Pass - Step 2

activity that has *rate float* can potentially progress at a slower rate than planned without affecting the duration of the project. A discussion of *rate float* analysis is presented in the section titled "Rate Float".

To perform the *downward pass*, start with the end of the last activity on the project. In this example, that point would be the end of activity **F**, as shown by point 1 on Figure 32. Next, follow activity **F** back in time, or downward on the linear schedule, until the *potential controlling link* with activity **E** is reached, as shown by point 2 on Figure 32. The segment identified by these two points determines the portion of activity **F** that is a *controlling segment* and the *potential controlling link* is now a *controlling link*. A *controlling link* is a point in time where the controlling activity path changes from one activity to another. Move horizontally to activity **E** along the *critical link* beginning at point 2 on Figure 32, and repeat the process performed on activity **F** to find the *controlling segment* on activity **E**. If, while moving back in time along an activity, the beginning of the activity is reached before a *potential controlling link* with a preceding activity is reached, a new *critical link* is established at the beginning of the activity. This is illustrated by the new critical link established with activity **A** at point 3 on Figure 32. Repeat the process of identifying critical segments and critical links by moving downward through the linear schedule until the start of the project is reached.

Once the *backward pass* is complete, the *controlling activity path* through the linear schedule has been identified. Notice, on Figure 32, that each *controlling segment* has endpoints identified. These endpoints describe when and where (at what time and location)

77
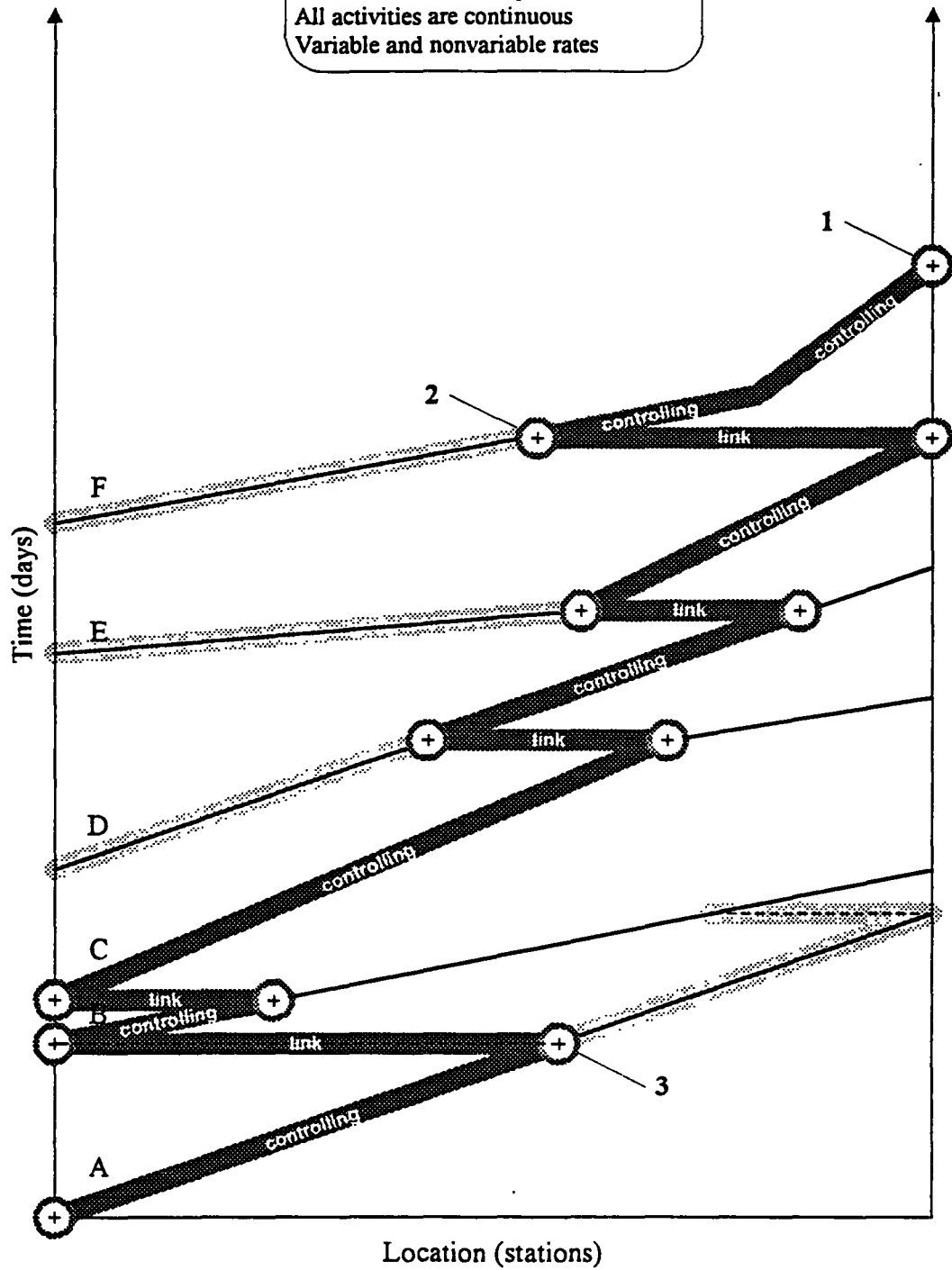
Figure 32 Downward Pass

the activity will become a controlling activity, and when and where it will cease to be a controlling activity.

Intermittent Linear Activities

Up until now, the analysis of controlling activities has dealt only with *continuous full-span linear* (CFL) activities. This section will begin the discussion of other types of activities defined earlier as *intermediate activities*. Figure 33 shows an example of an *intermittent full-span linear* activity, sandwiched between two CFL activities. In fact, to perform the analysis on any activity other than a CFL activity, or an *intermediate activity*, it will need to be sandwiched between two CFL activities which can include the CFL activities called Start and End.

An *intermittent activity* is an activity that is not expected to be in continuous operation. These types of activities are usually paced by a lower production rate activity, an activity that covers fewer stations per day. As the lower production rate activity progresses, the intermittent activity starts and stops to maintain a reasonable working distance behind the lower production rate activity. An example of an intermittent activity would be a grade preparation activity that follows a concrete removal activity on a highway reconstruction project. In this example, the concrete removal activity is a lower production rate activity than the grade preparation activity. Therefore, the grade preparation activity could start at some later point in time and still be complete at the appropriate interval after the concrete removal activity has completed. In the actual construction of the project, however, the grade preparation activity will likely start and stop so that it is never further from the concrete
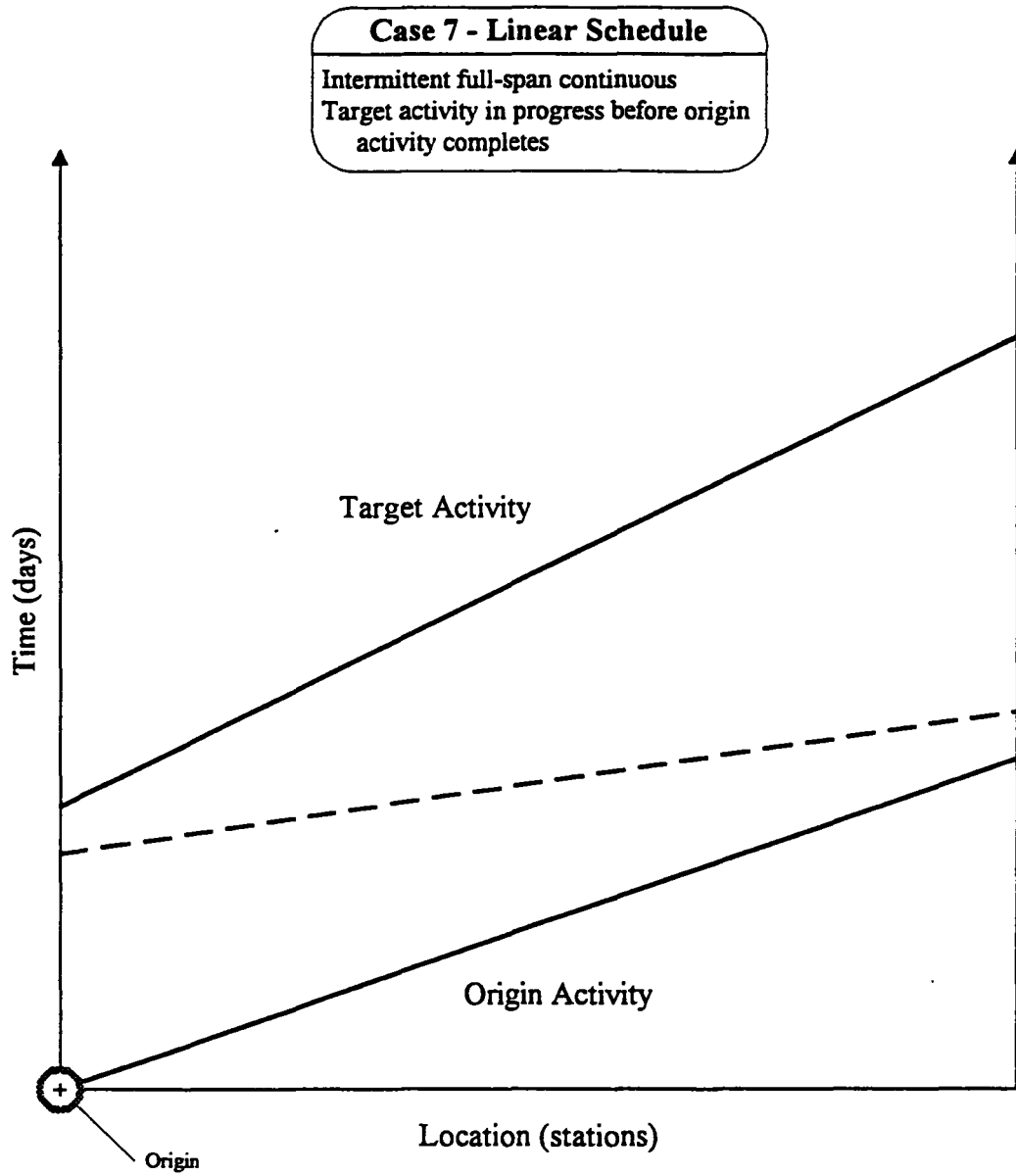
Figure 33 Case 7 - Linear Schedule

removal than some predetermined maximum distance. The close proximity of activities

minimizes the risk that the grade preparation activity could have an effect on the duration of

the project. Since an intermittent linear activity can start and stop, it will never be a

controlling activity unless it is the only activity in progress. The buffer between the

intermittent activity and the pacing activity should represent the maximum distance that the

planner expects to occur between the two activities. The following two cases illustrate the

analysis of an intermittent activity with respect to identifying controlling segments.

<u>Case 7</u>

Case 7, illustrated on Figures 33 - 36, shows an *intermittent full-span linear* (IFL)

activity that will not be on the *controlling activity path*. The *target activity* starts before the

origin activity ends. Therefore, the IFL activity is ignored and the *potential controlling*

segment of the *origin activity* is determined by the *target activity* only. Figure 36 shows the

result of the *downward pass* and identifies the *controlling activity path*.

<u>Case 8</u>

Case 8, Figures 37 - 40, shows an *intermittent full-span* (IFL) activity that has a

controlling segment because it is the only activity in progress. The *origin activity* ends before

the *target activity* begins. During this interval of time, the IFL activity must be defined as a

controlling segment since it is the only activity in progress. Figure 40 shows the result of the

downward pass and identifies the *controlling activity path*.

Intermediate Activities

The analysis of controlling activities has only involved full-span linear activities up to
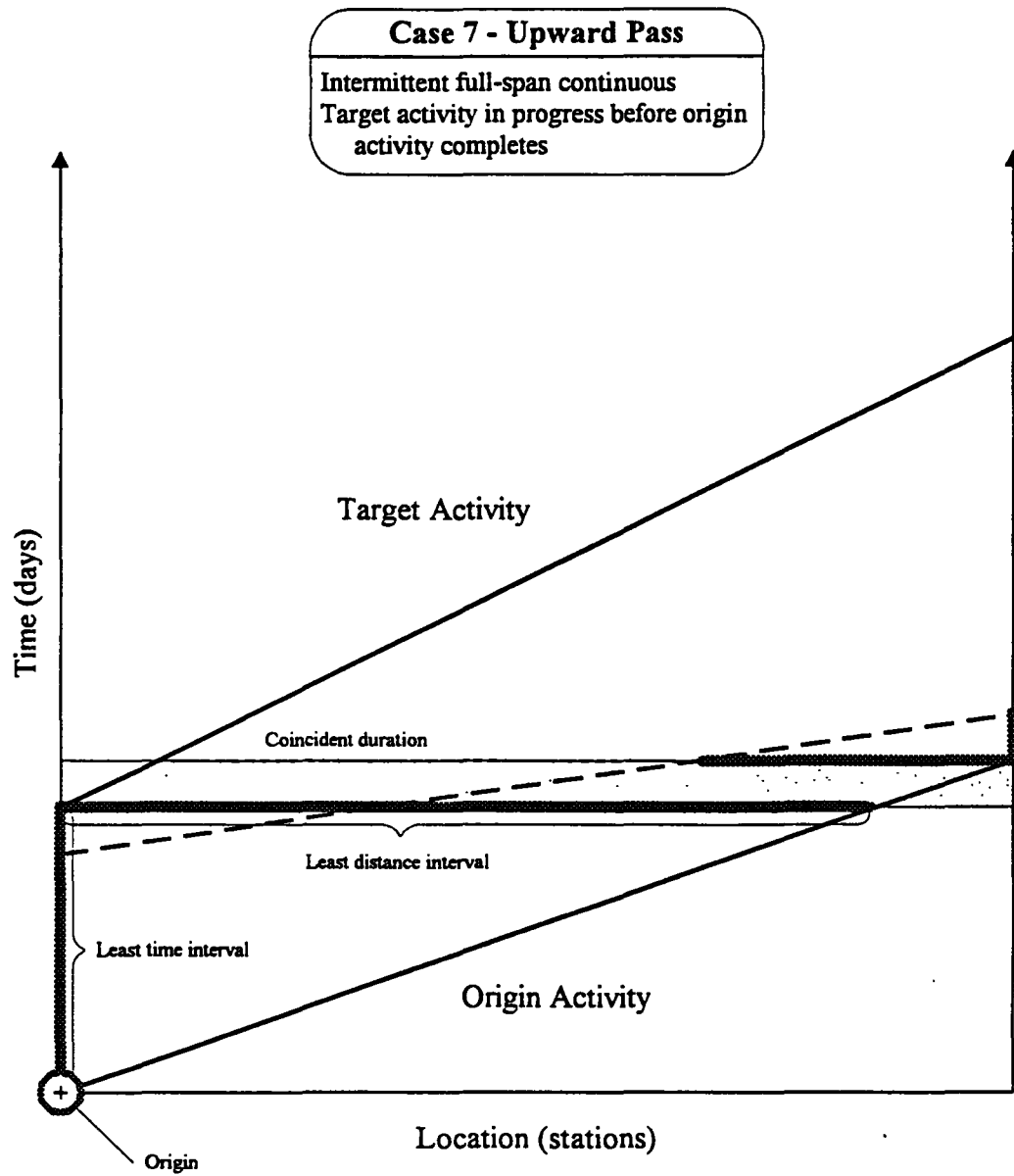
Figure 34 Case 7 - Upward Pass Step 1
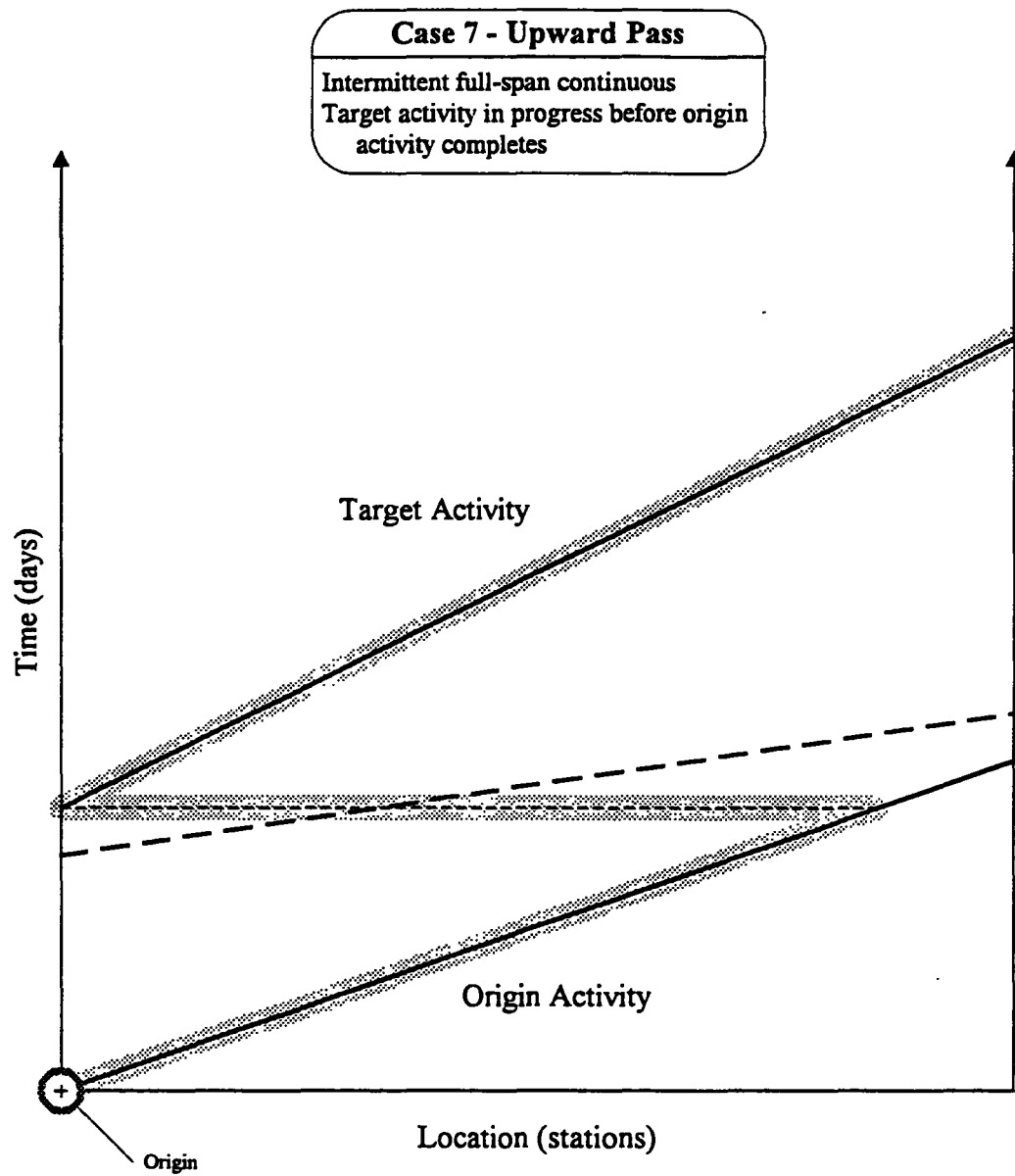
Figure 35 Case 7 - Upward Pass Step 2
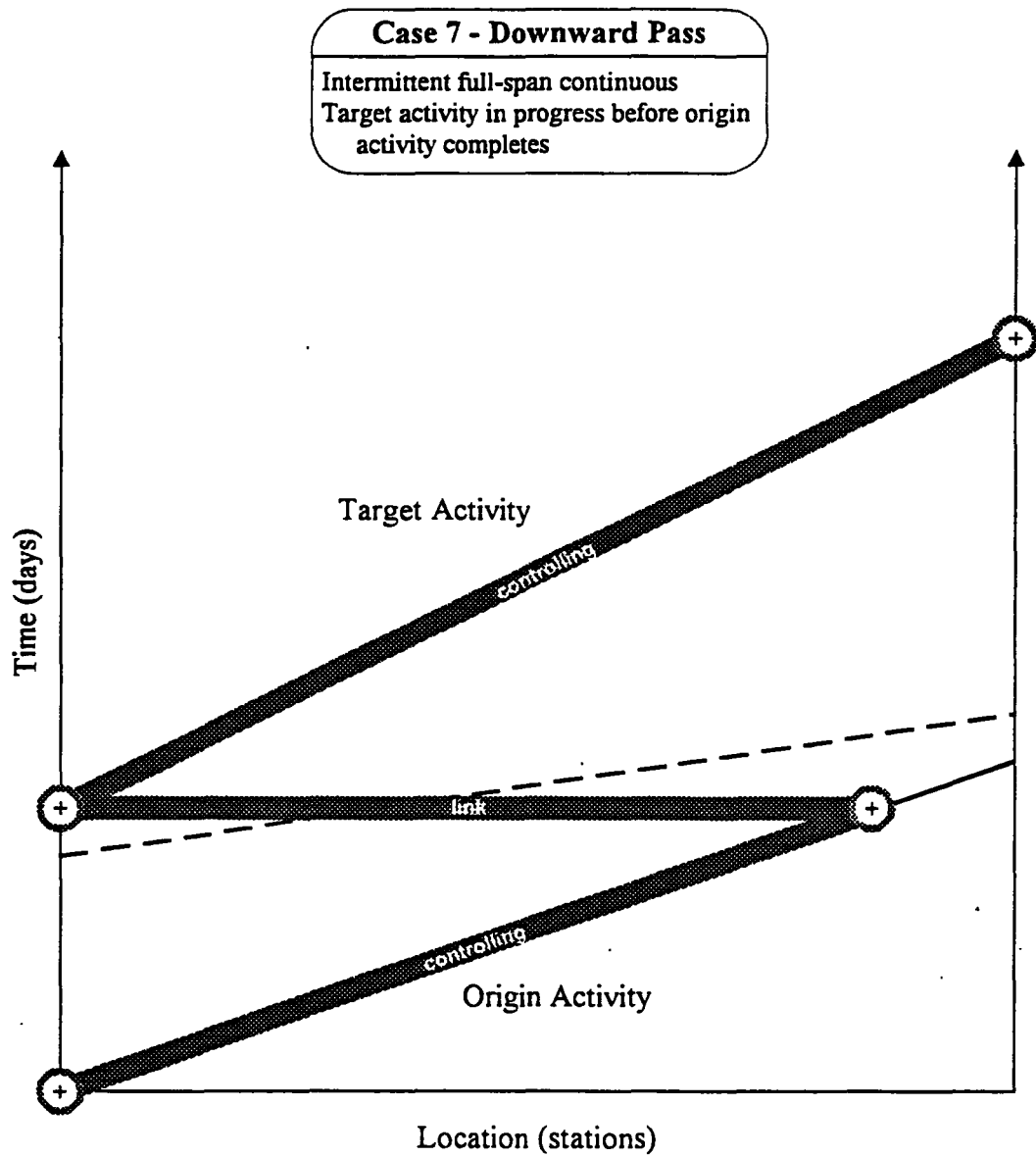
Figure 36 Case 7 - Downward Pass
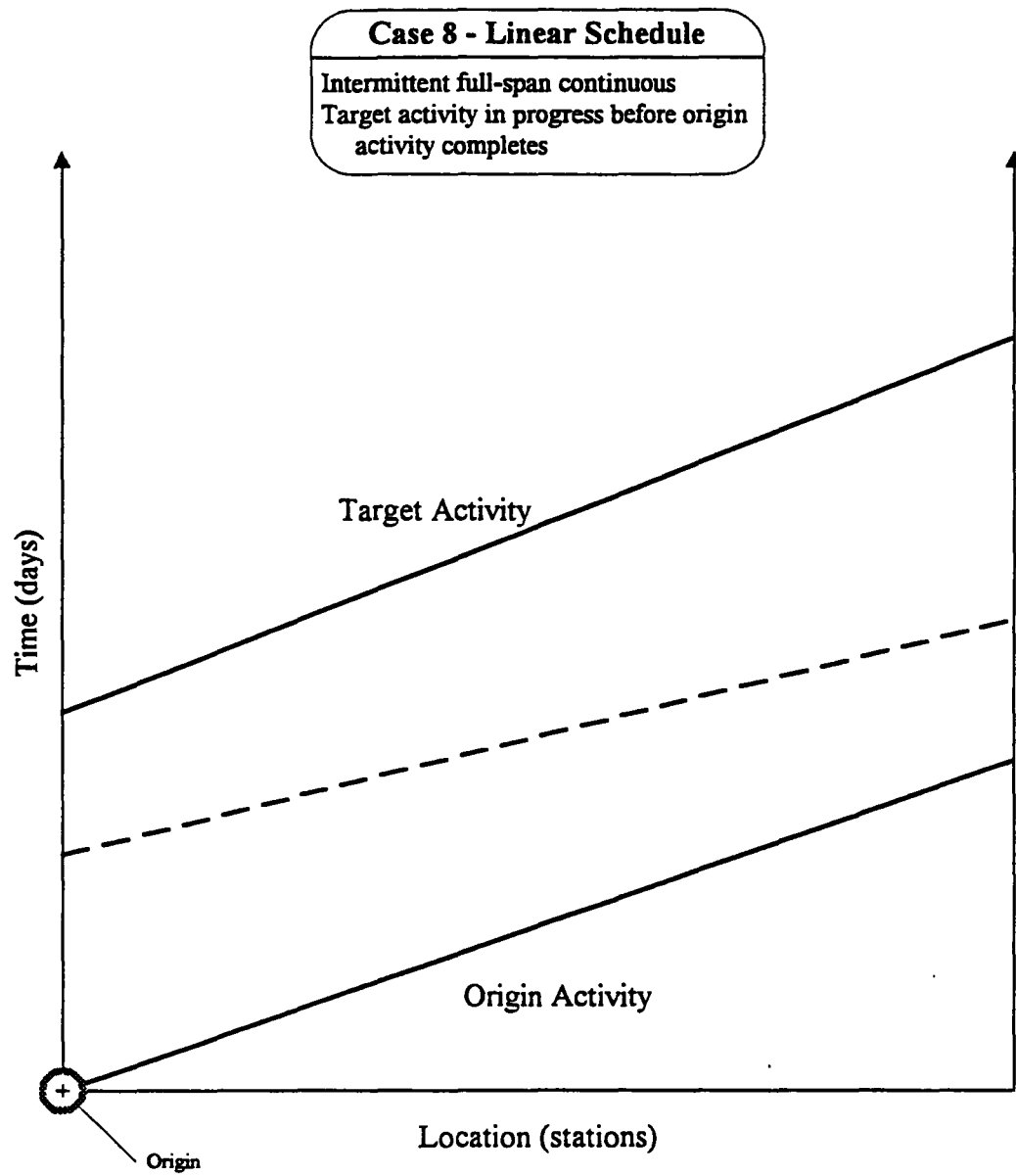
Figure 37 Case 8 - Linear Schedule

85
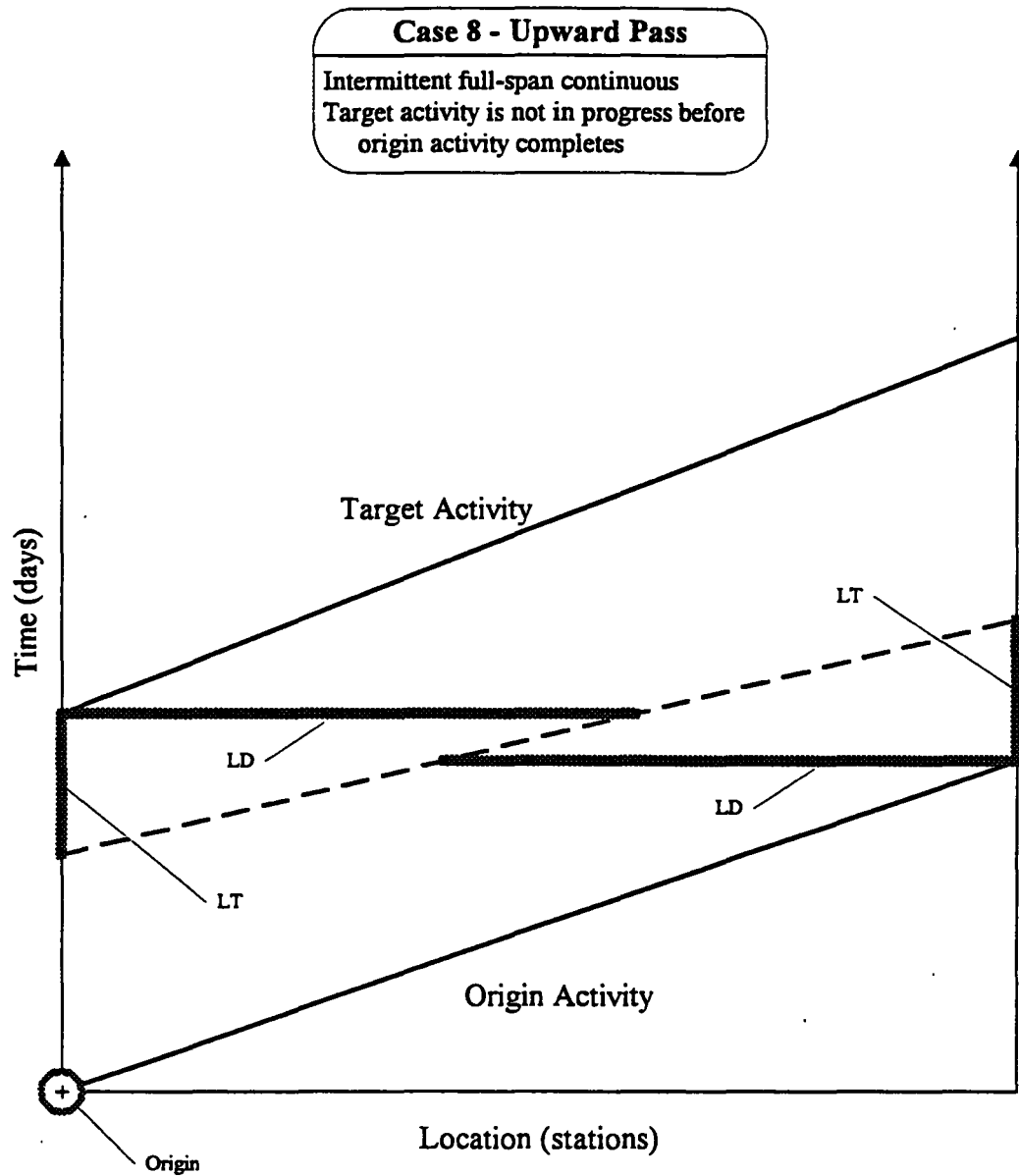
Figure 38 Case 8 - Upward Pass Step 1
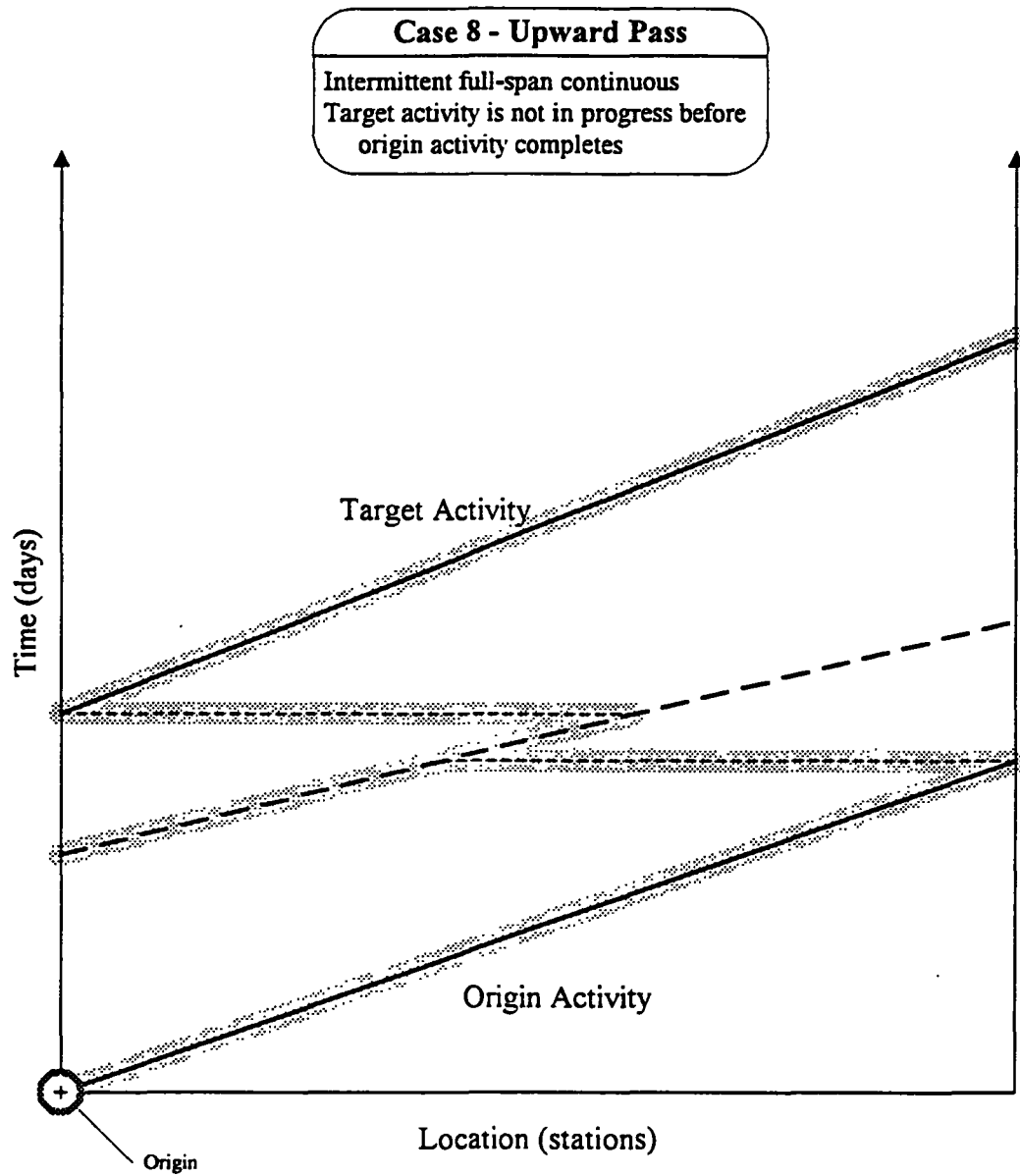
Figure 39 Case 8 - Upward Pass Step 2
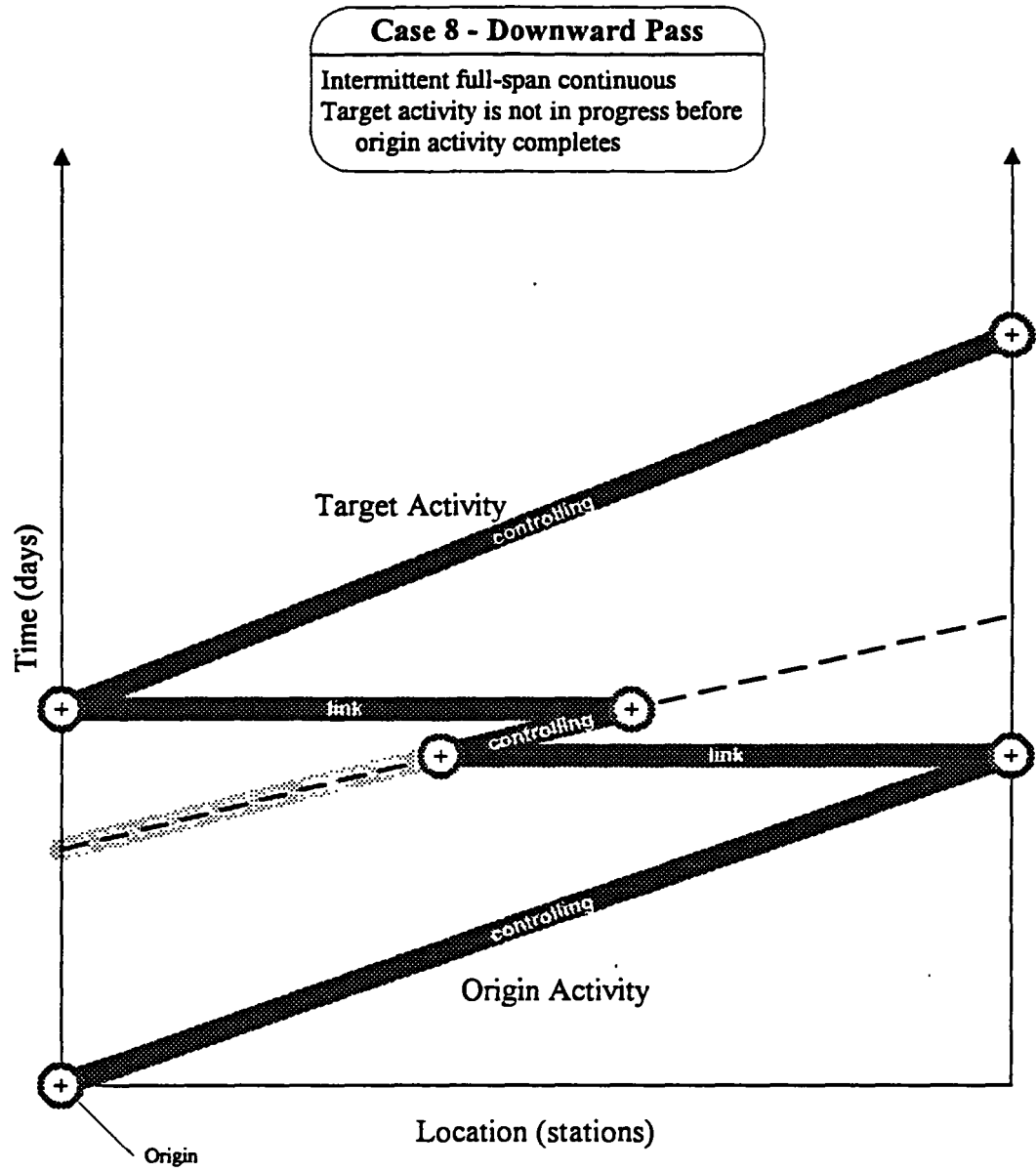
Figure 40 Case 8 - Downward Pass

this point. This section will begin the discussion of *intermediate activities*. An intermediate activity is any activity other than full-span linear activities. The reason they are termed *intermediate activities* is that their analysis involves an intermediate step in the typical analysis of *continuous full-span linear* activities presented earlier. *Intermediate activities* are always viewed as "sandwiched" between two CFL activities. There is a CFL activity preceding, and another CFL activity following every *intermediate activity* on a linear schedule. Remember that the Start and End activities are considered CFL activities in the analysis of *intermediate activities*. The cases that follow will deal primarily with block and partial-span linear activities. Bar activities will not be addressed directly in a case since a bar activity is really a very narrow block activity. Any discussion pertaining to partial-span block activities can be directly applied to bar activities as well.

Case 9

This case presents the analysis of a full-span block activity shown in Figure 41. Since a *full-span block* activity covers the entire location of the project, the *activity sequence list* is simplified. At any location on the project, the activity sequence is the origin activity, the full-span block, and then the target activity. This means that the least time and least distance intervals only need to be found between the origin activity and the block, and the block and the target activity as shown in Figure 42. The next step, is to determine the potential critical segments of the activities. Figure 43 shows the *potential controlling segments*. If, an *intermediate activity* is potentially controlling, the entire activity is assumed to be potentially controlling. For this case, the target activity is also assumed to be the last activity so that the
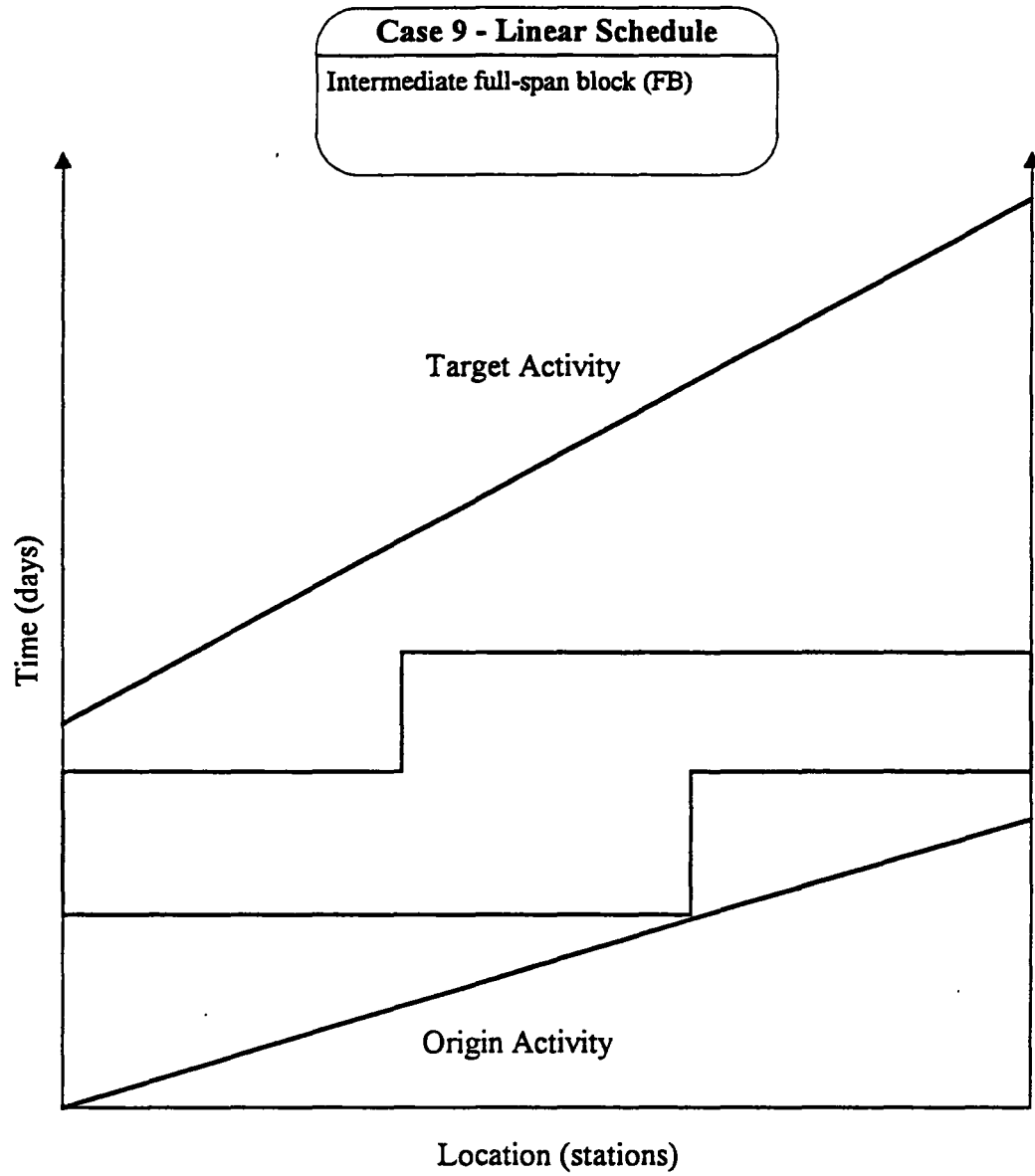
Figure 41 Case 9 - Linear Schedule
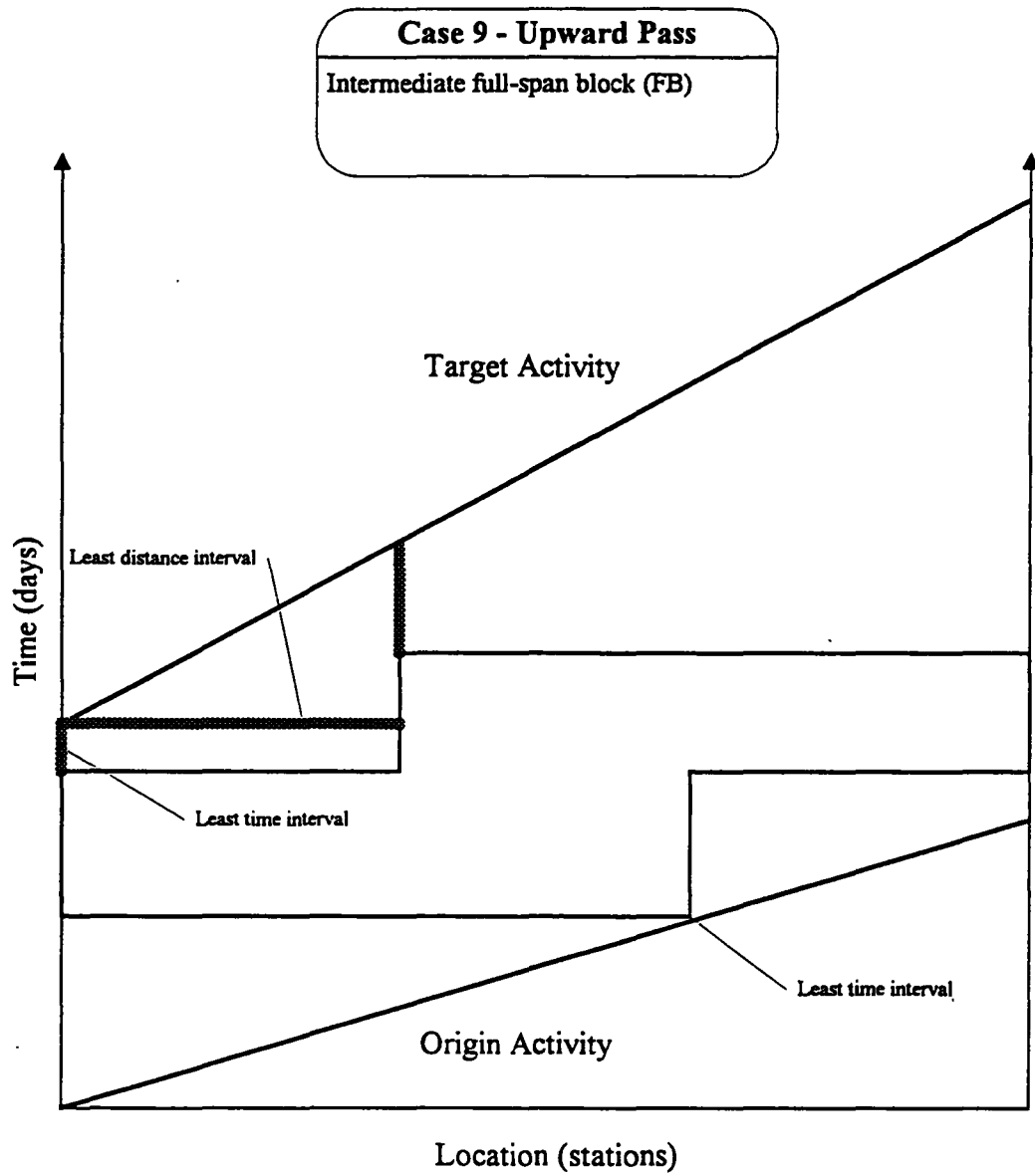
Figure 42 Case 9 - Upward Pass Step 1

Case 9 - Upward Pass

Intermediate full-span block (FB)
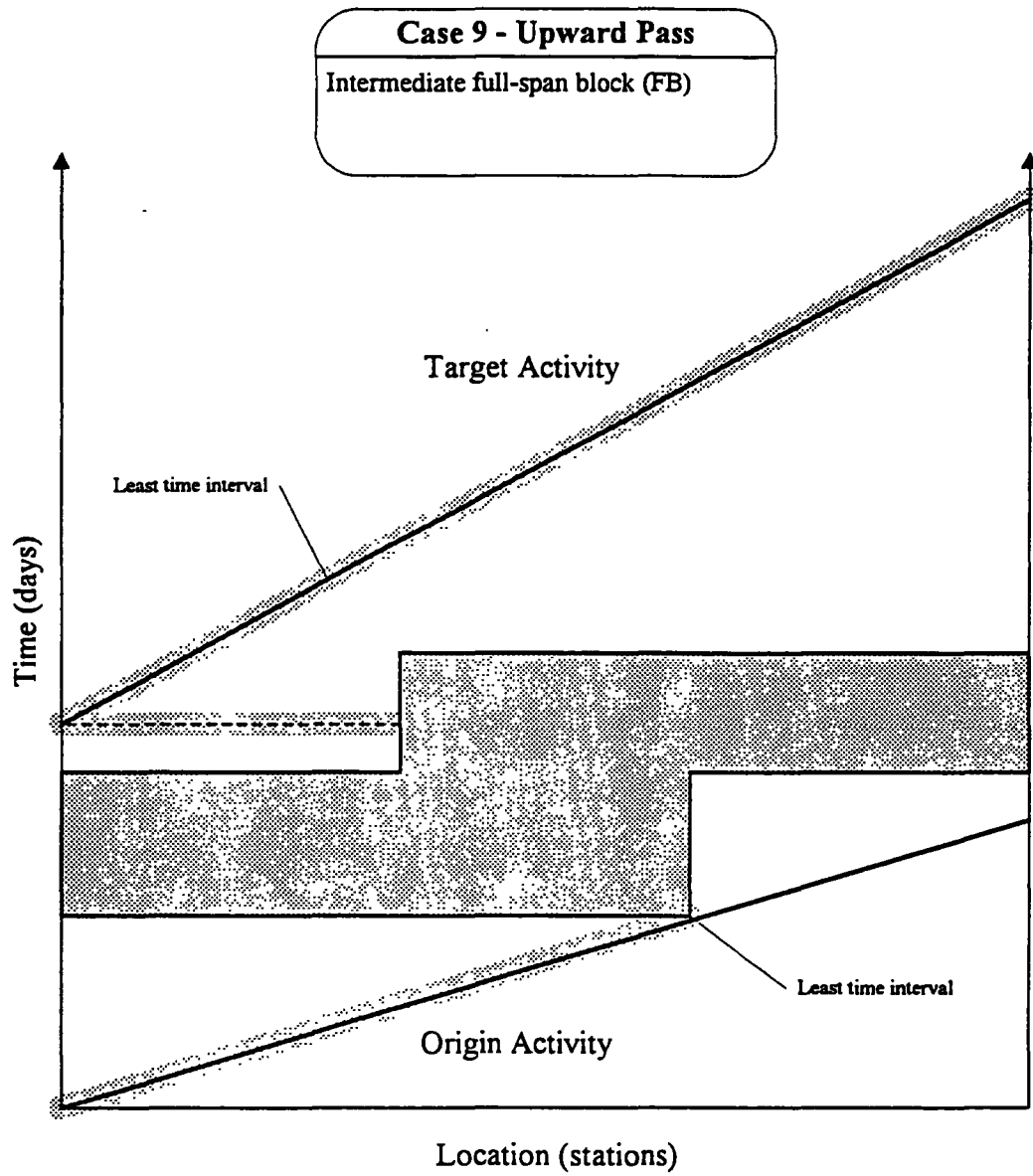
Figure 43 Case 9 - Upward Pass Step 2

effect of the *downward pass* can be illustrated. Figure 44 shows the result of the *downward pass*. Only a portion of the block activity is actually controlling. Any portion of the block activity occurring later in time than the beginning of the target activity cannot be considered controlling. Obviously, the duration of the unshaded portion of the block activity could extend upward until it reached a distance in time equal to the least time interval found earlier. At this point, the controlling activity path could change to include more of the block activity.

Case 10

Block activities can be partial-span, covering only a portion of the project location, as shown in Figure 45. A third CFL activity, activity **D**, is included to show the effects this activity can have on the controlling activity path. In this case, activity **D** has a lower production rate than activity **C**. In Case 11, this situation is reversed. To begin the analysis of this linear schedule, the activity sequence must be determined. There are two possible sequences between activity **A** and activity **C**. The activity sequence list is: **AC**, **ABC**, and **CD**. Since there are two possible routes between activities **A** and **C**, the one with the *least time interval* (LT) must be determined. Figure 46 shows that the sequence of **ABC** contains the least time interval since the sum of the LTs in this sequence is less than the least time interval between activities **A** and **C** only. The *least time interval* (LT) and the *least distance interval* (LD) for sequence **CD** is also shown on Figure 46. The least distance intervals for **AB** and **BC** are at the points of intersection between these activities.

Figure 47 shows the potential controlling segments that have been established from the least time and least distance intervals. Note, as in Case 10, since the intermediate activity is a
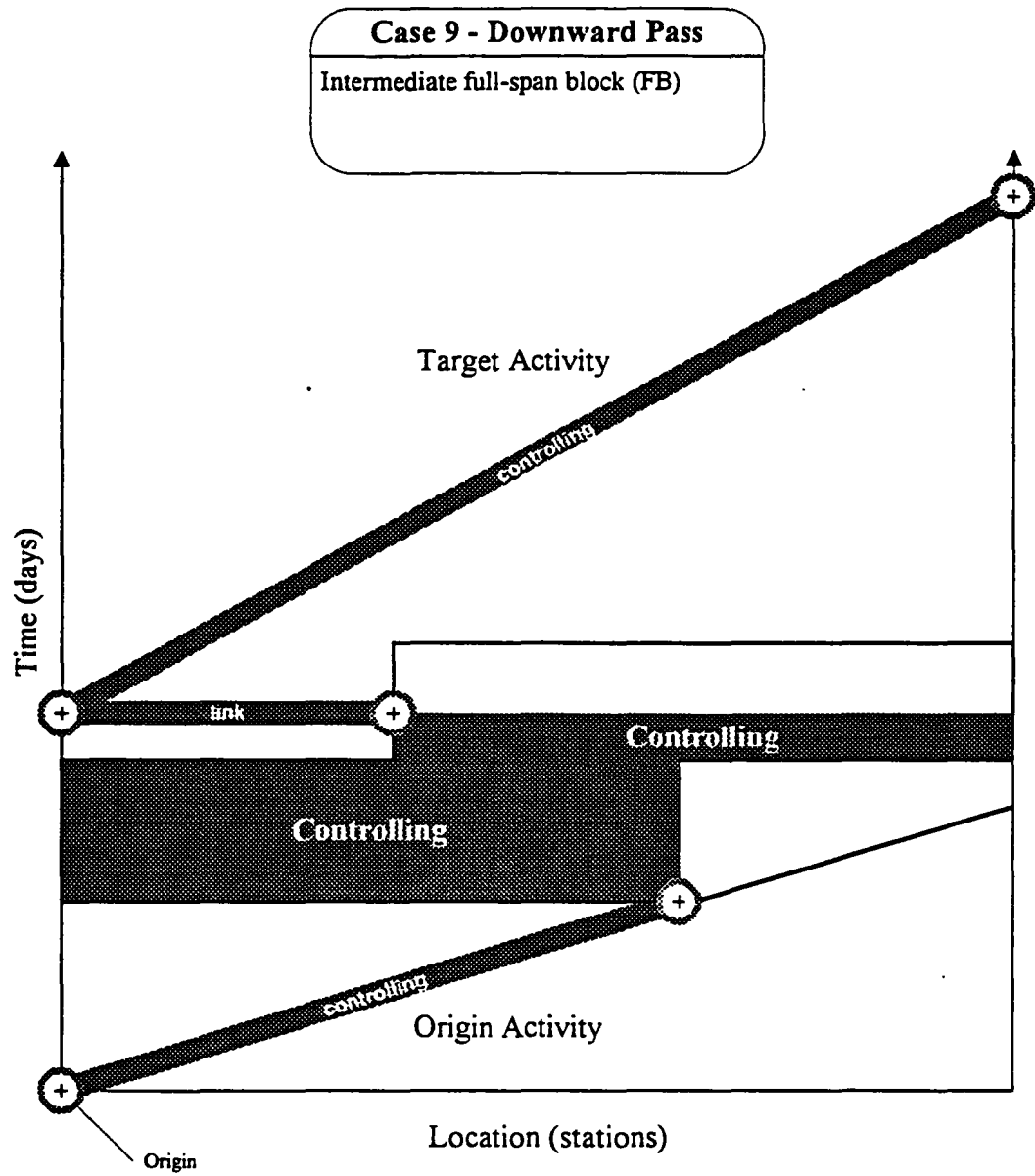
Figure 44 Case 9 - Downward Pass

Figure 45 Case 10 - Linear Schedule

Case 10 - Upward Pass

Intermediate partial-span block (PB)

Time (days)

D

LD

LT

C

LT

A

B

LT

Location (stations)
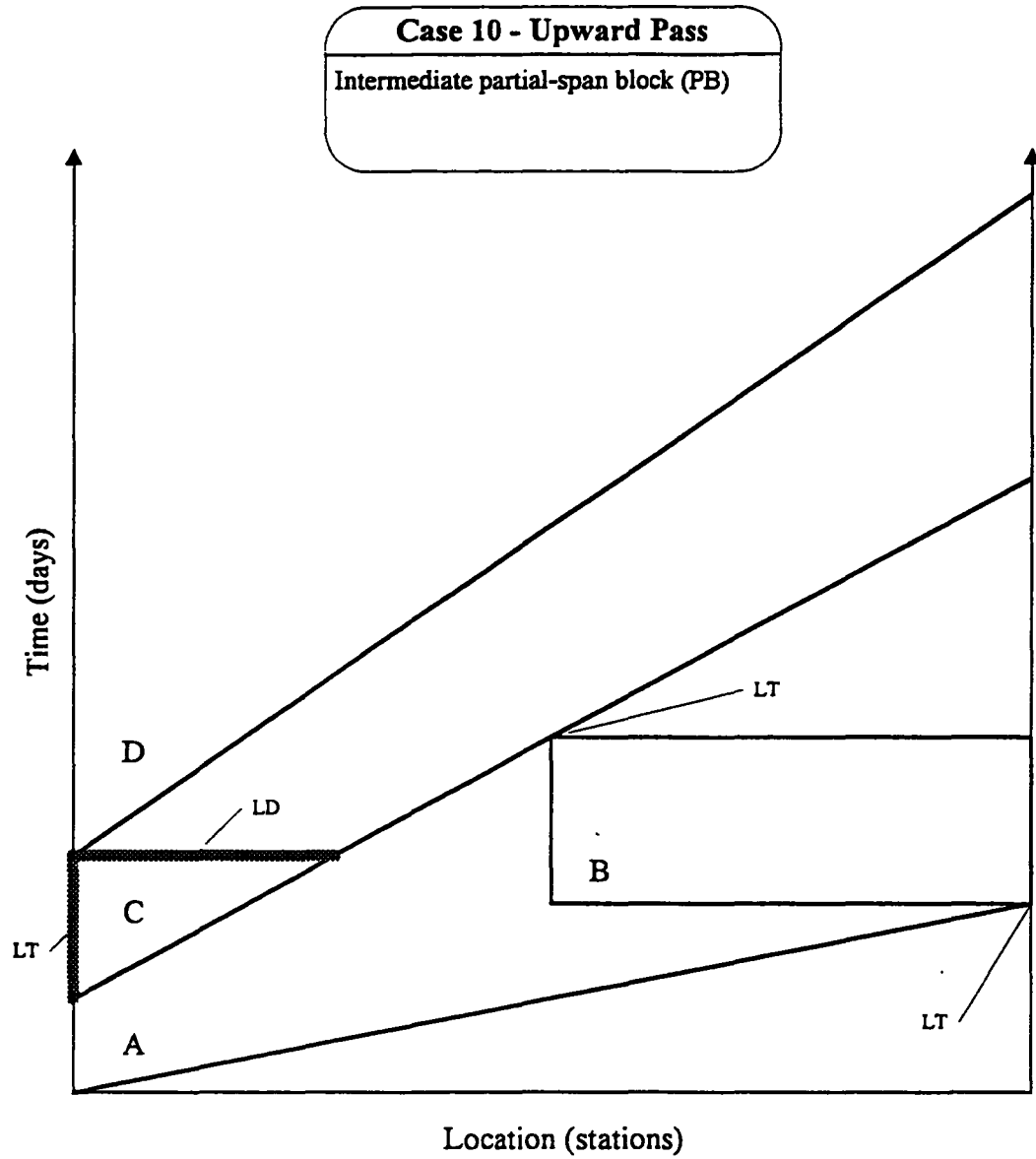
Figure 46 Case 10 - Upward Pass Step 1
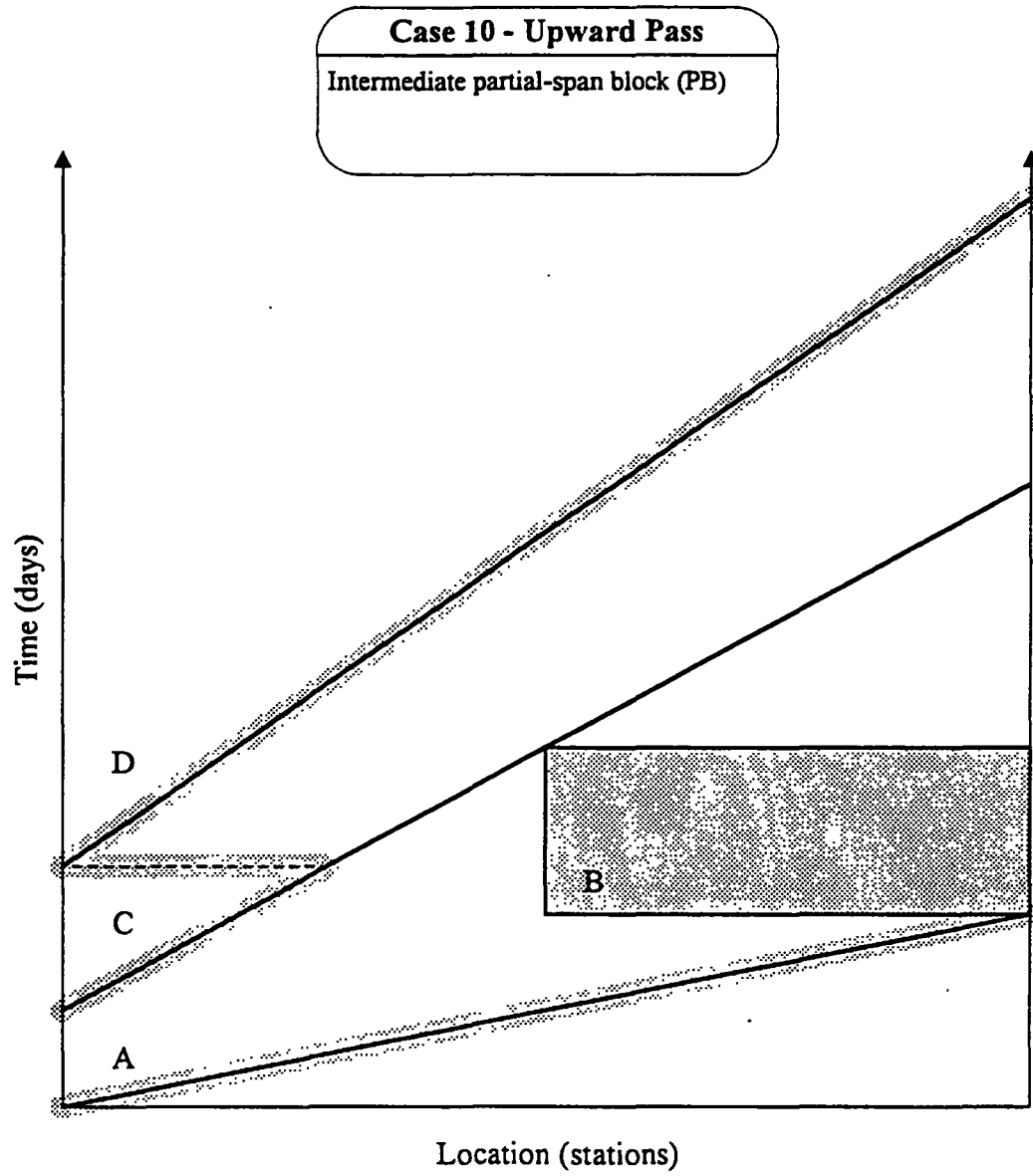
Figure 47 Case 10 - Upward Pass Step 2

potential controlling segment the entire activity is considered potentially controlling. The downward pass analysis is shown on Figure 48 The effect of activity **D** on the controlling activity path can be observed. Even though activity **B** was a potentially controlling segment, it is not on the controlling activity path.

Case 11

Case 11 is identical to activity Case 10 except that the production rate of activity **D** is now higher than that of activity **C**, as shown in Figure 49. Figures 50 and 51 show the results of the *upward pass*. The least time interval between activity **C** and activity **D** has shifted to the end of activity **C**. The result of this shift can be seen in the *downward pass* presented in Figure 52. The *controlling activity path* now includes activity **B**.

Case 12

This case is similar to the previous two cases, except that the duration of activity **B** has been decreased as shown in Figure 53. This causes the *least time interval* to occur in sequence **AC** rather than **ABC** as before. Figure 54 graphically shows the *least time interval* analysis for activities **A** and **C**. Figure 55 shows the *potential controlling segments* developed during the *upward pass* for this schedule. Activity **B** is not a *potential controlling segment* as in the last two cases. The *downward pass* for this situation is shown in Figure 56.

As discussed earlier, the analysis for a *bar* activity is identical to a *partial-span block* activity. Therefore, the only activity type left is the *continuous partial-span linear* activity. Case 13 discusses the analysis of this type of activity.

Case 10 - Downward Pass

Intermediate partial-span block (PB)
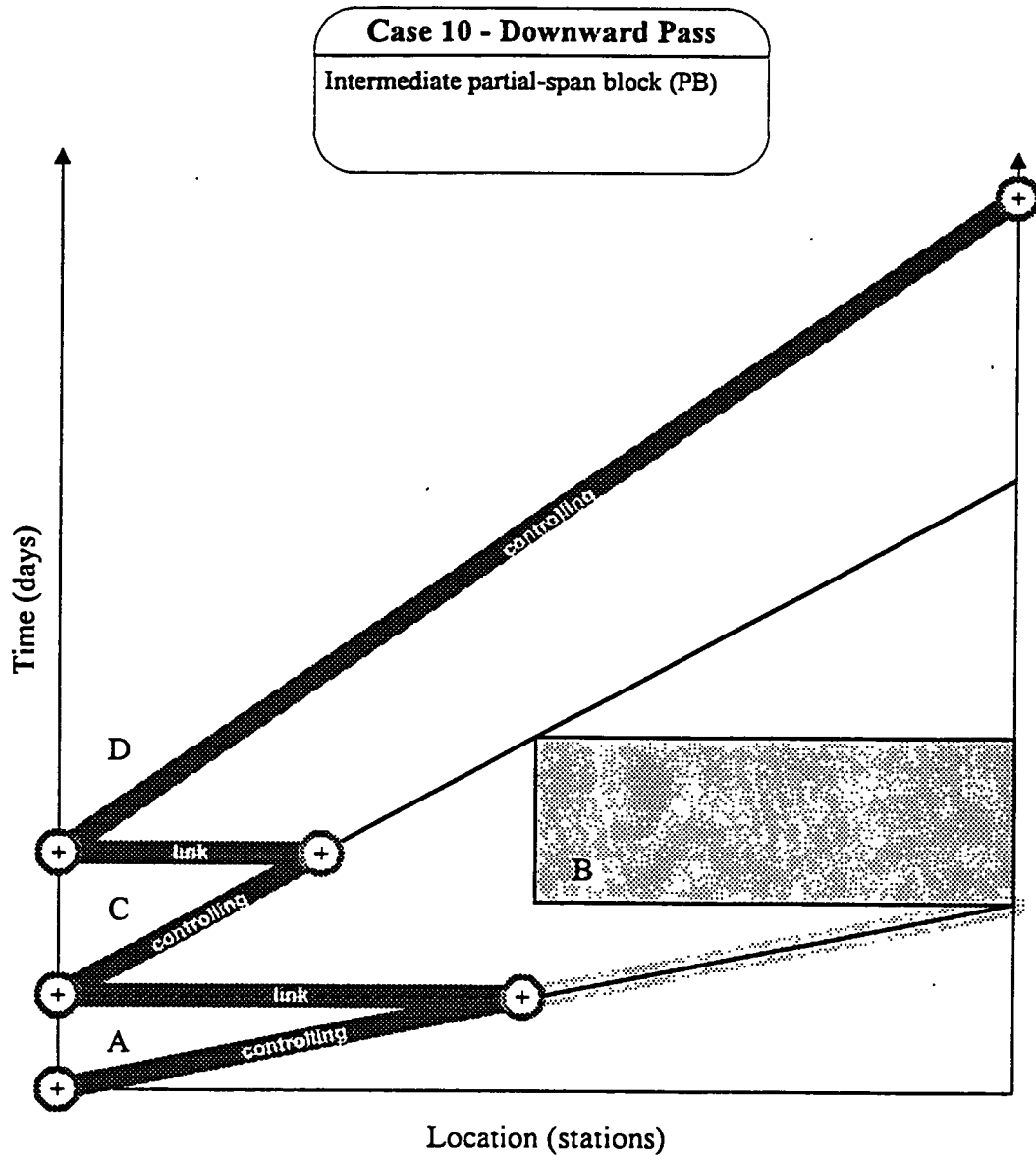
Figure 48 Case 10 - Downward Pass
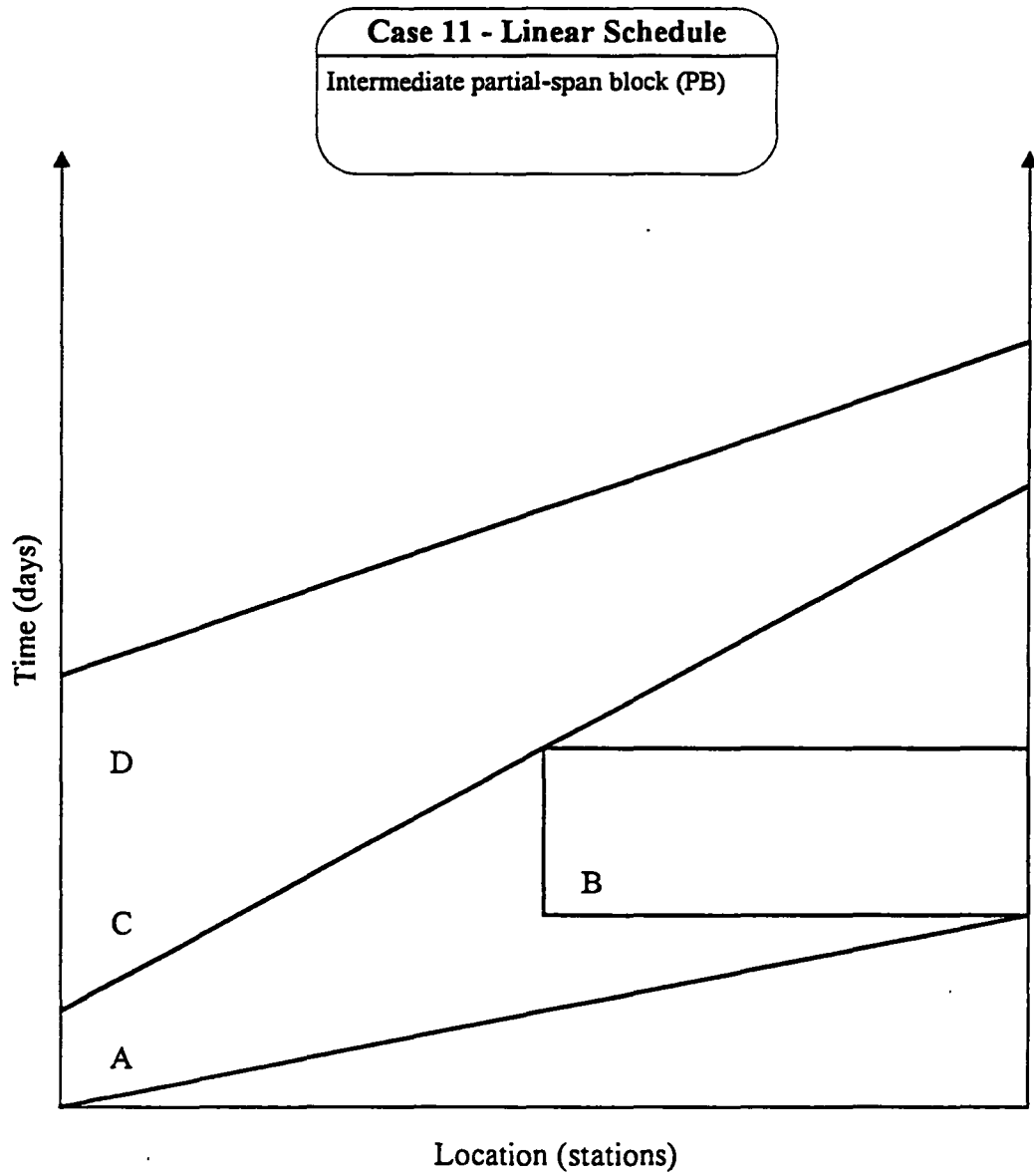
Figure 49 Case 11 - Linear Schedule
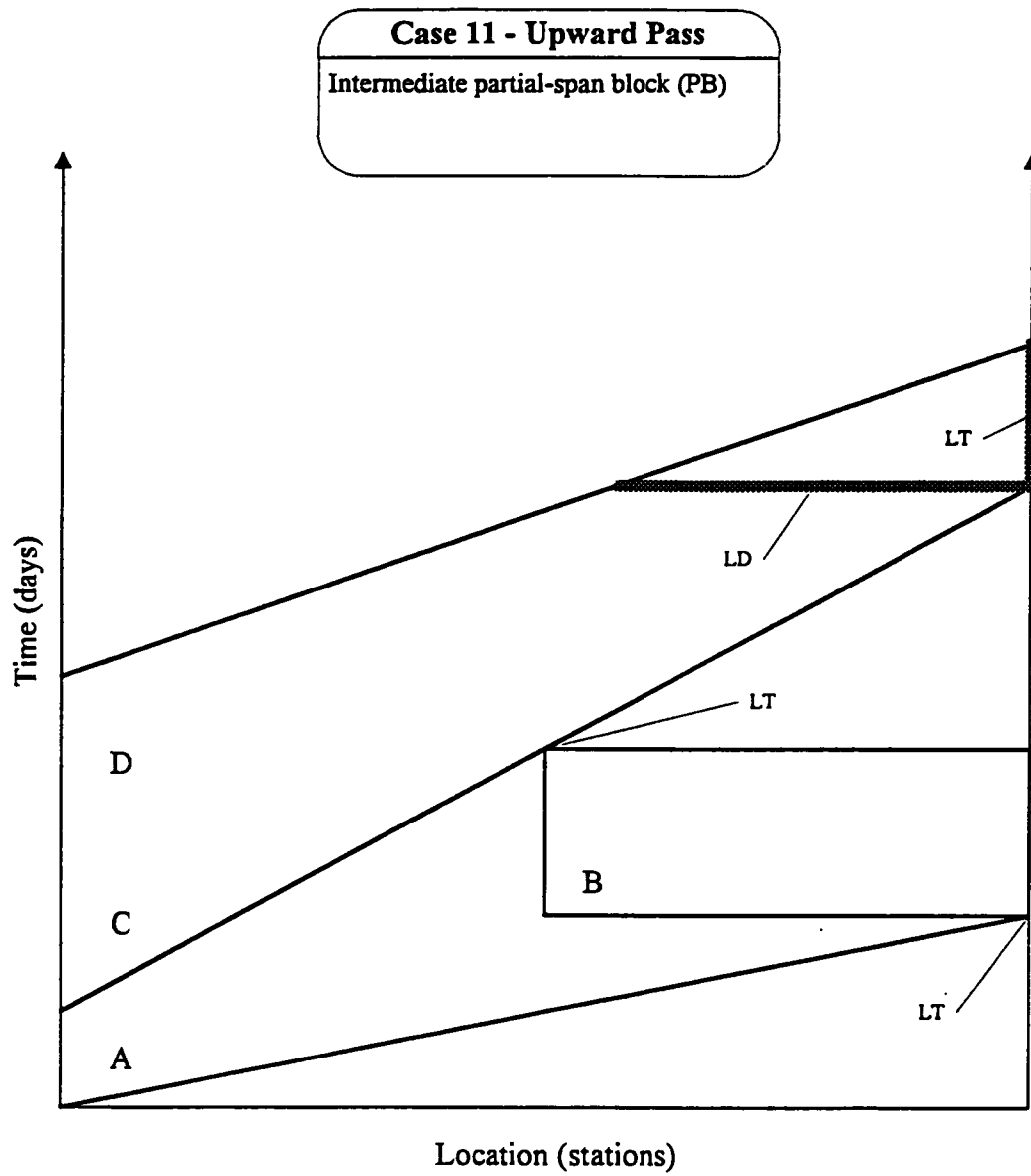
Figure 50 Case 11 - Upward Pass Step 1
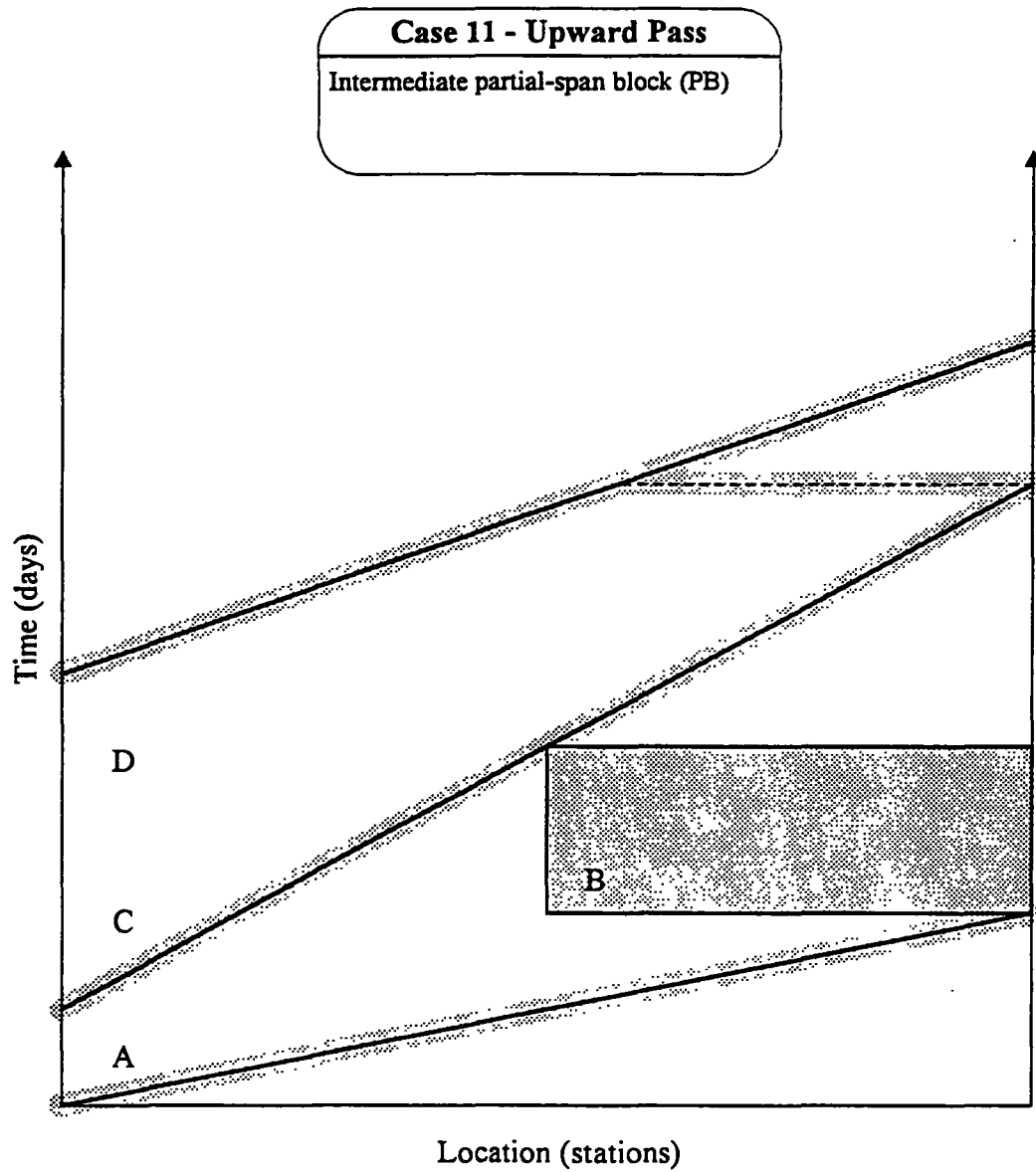
Figure 51 Case 11 - Upward Pass Step 2

102
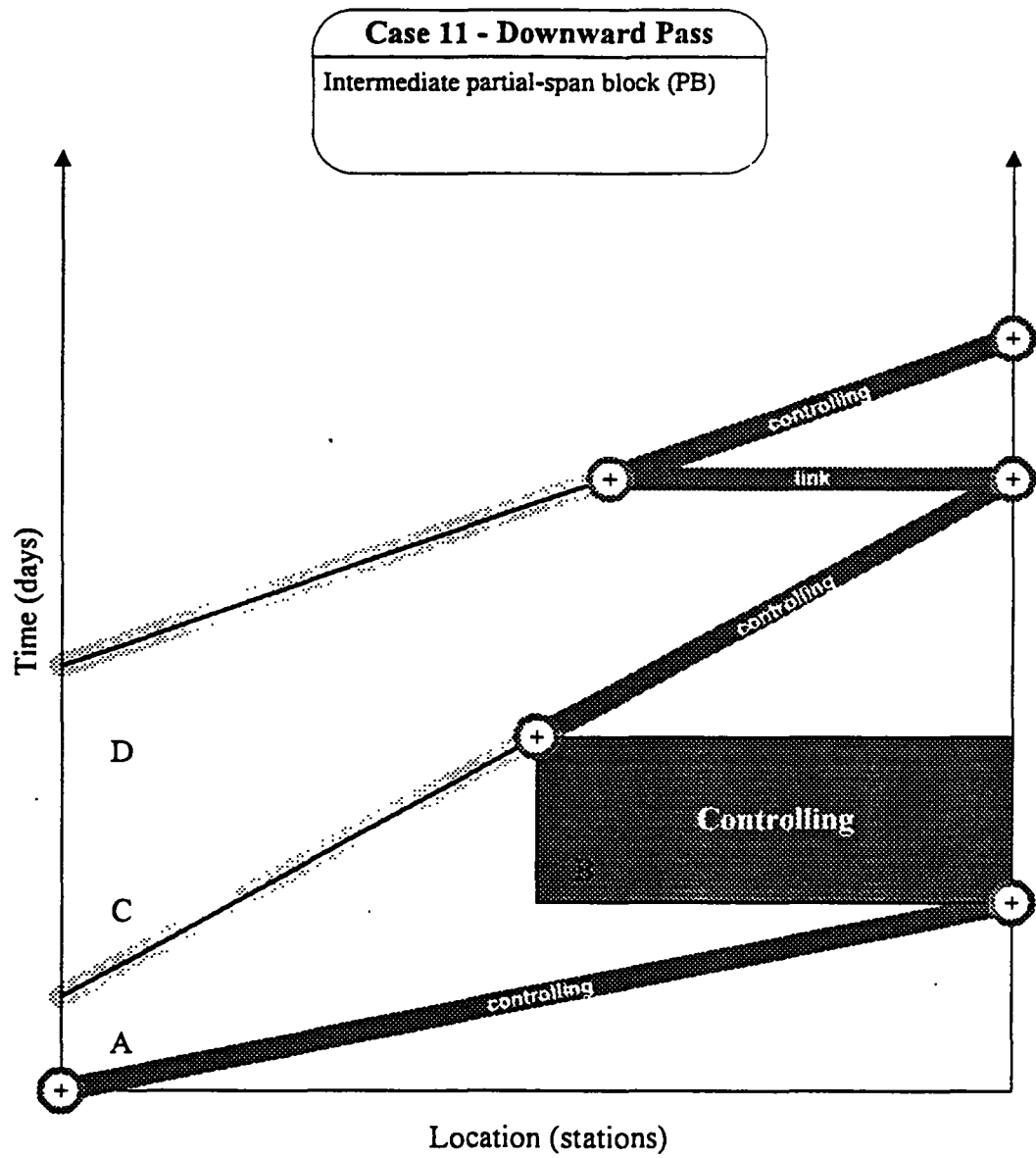
Figure 52 Case 11 - Downward Pass
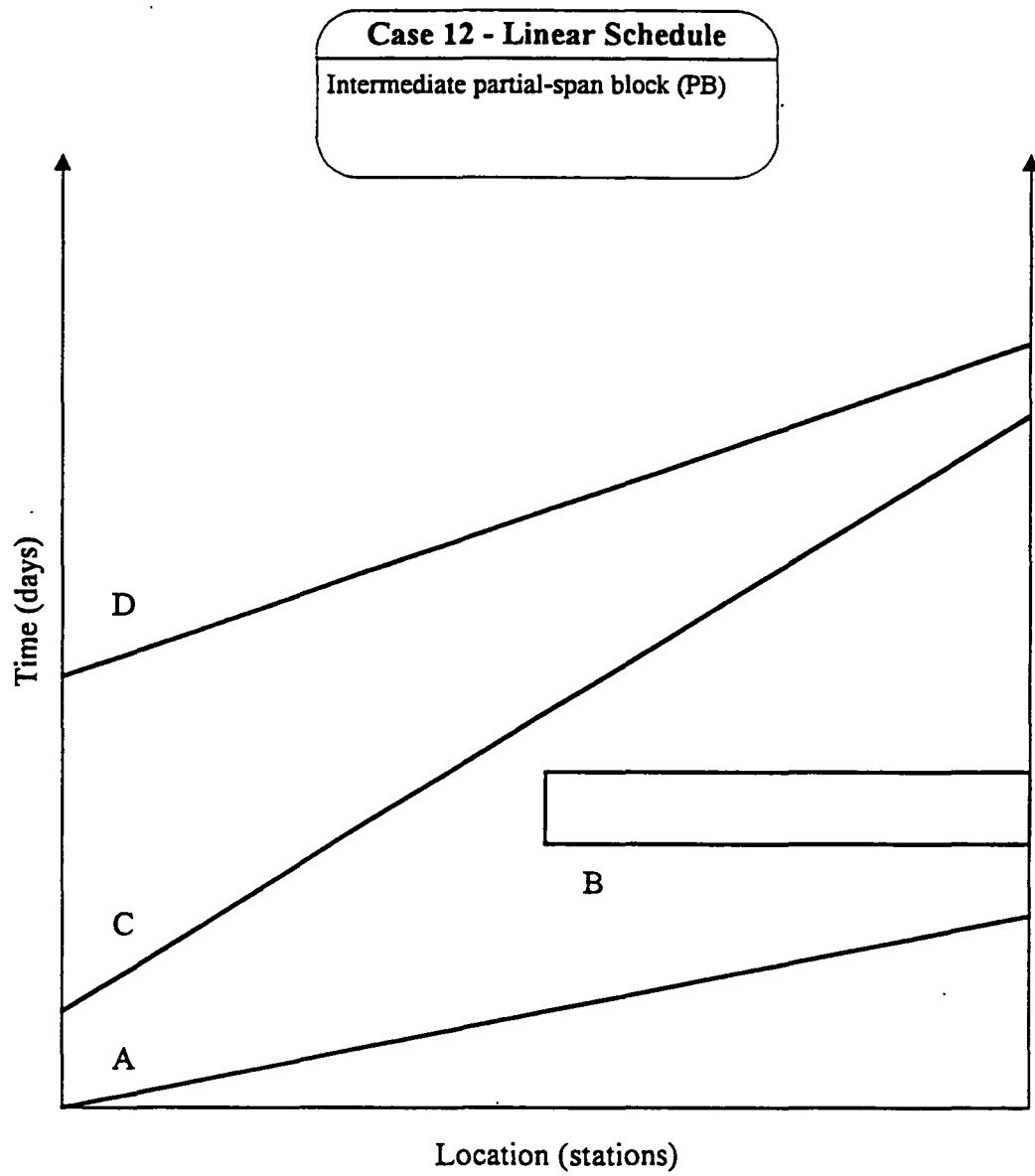
Figure 53 Case 12 - Linear Schedule
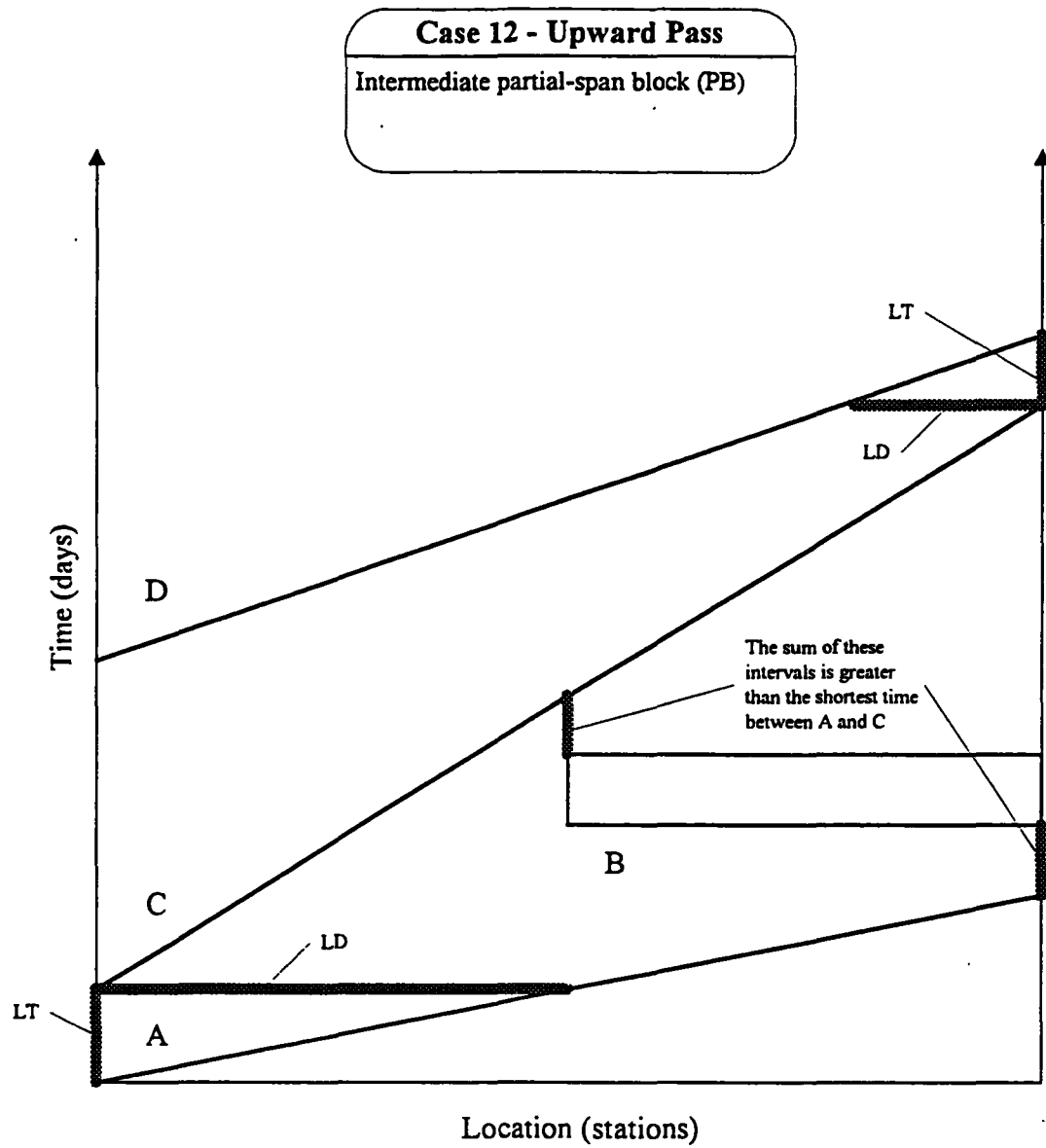
Figure 54 Case 12 - Upward Pass Step 1
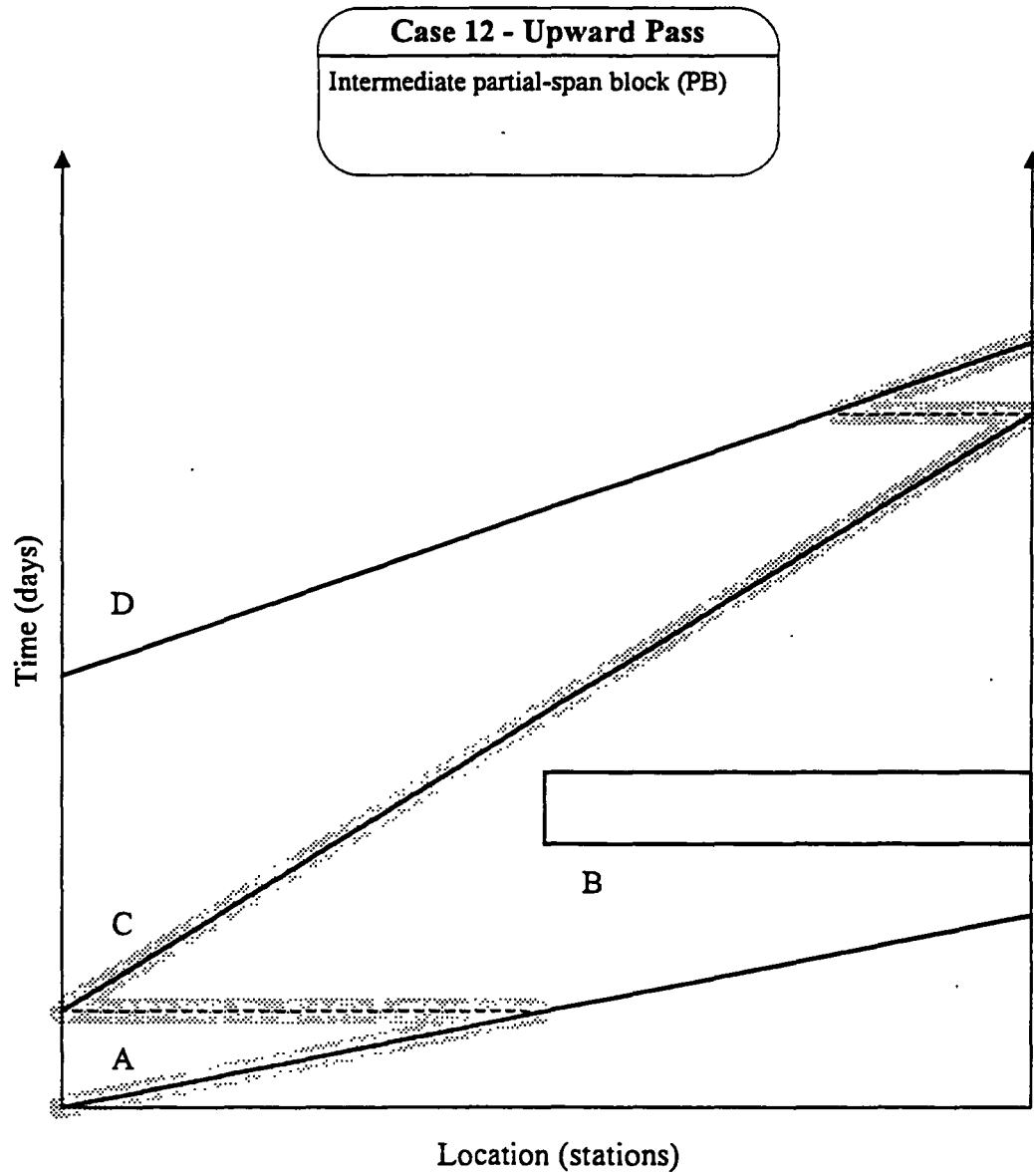
Figure 55 Case 12 - Upward Pass Step 2

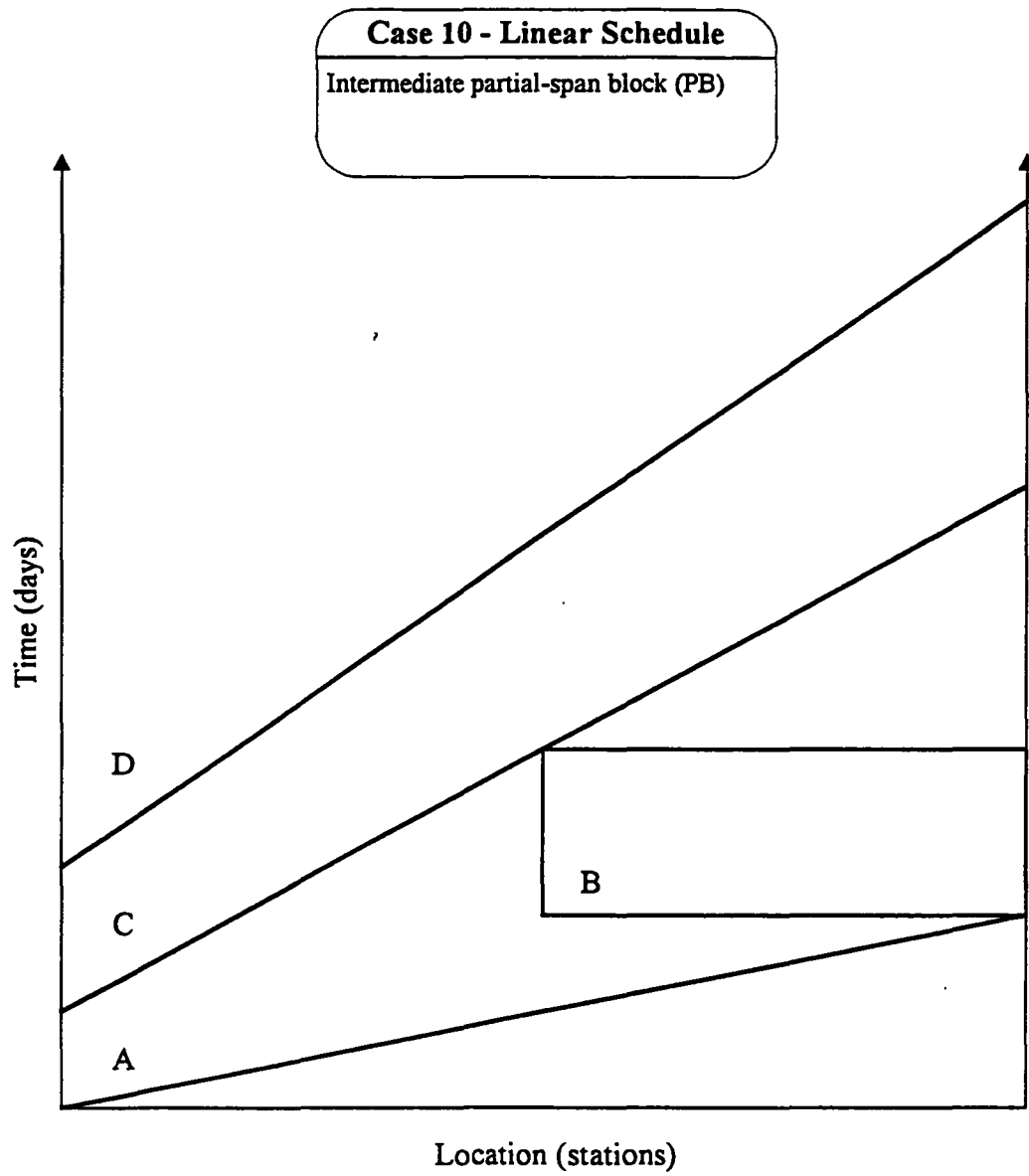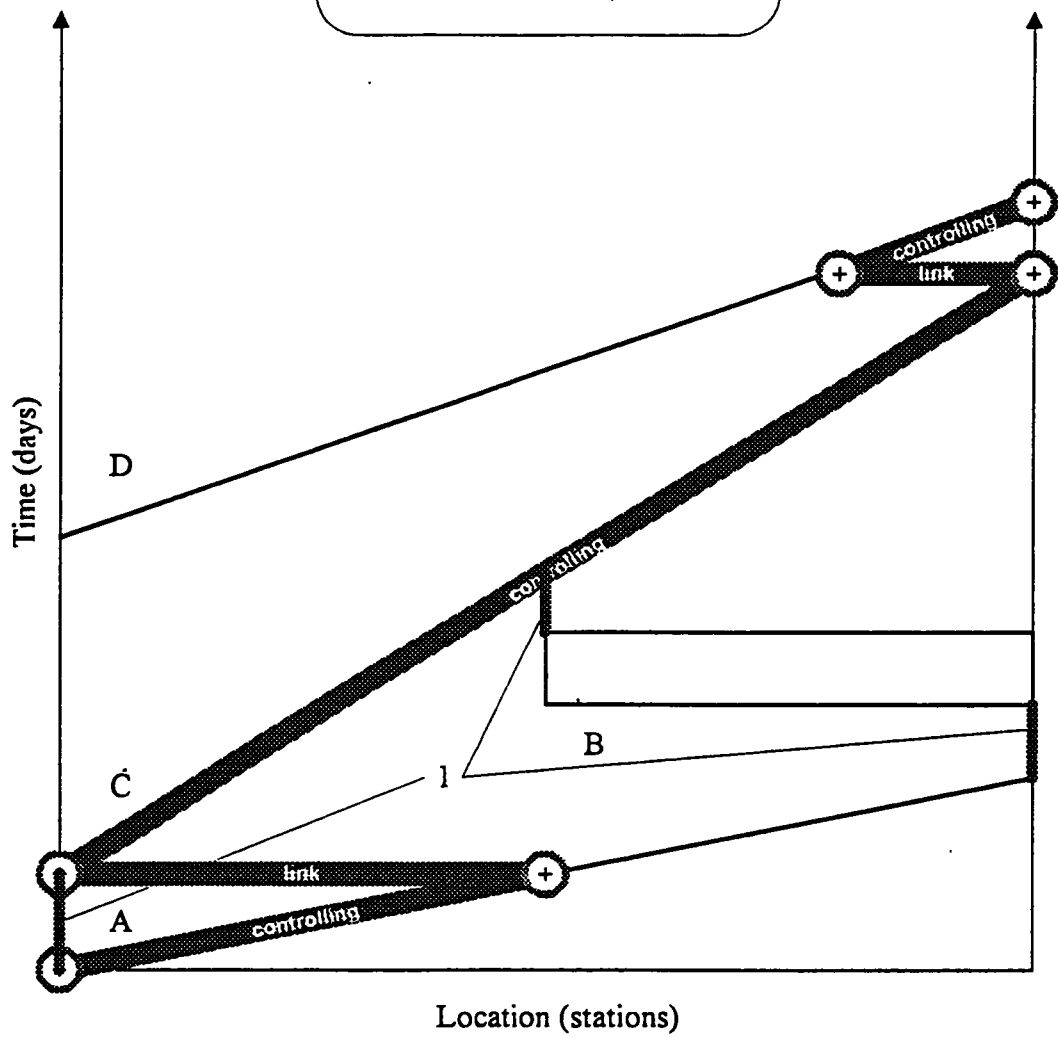**Figure 56** Case 12 - Downward Pass

<u>Case 13</u>

A linear schedule including an *intermediate activity* that is a *continuous partial-span linear* activity is shown in Figure 57. The *activity sequence list* for this schedule is the same as the previous three cases **AC, ABC, CD**. As in the analysis of a *partial-span block* activity, the *least time interval* along the possible paths between activities **A** and **C** must be determined. Figure 58 shows that the *least time interval* between activities **A** and **C** is along the sequence **ABC**. Once it has been determined that the *continuous partial-span linear* (CPL) activity is on the potential controlling activity path, the analysis proceeds as if the CPL activity was actually a CFL activity. Figure 59 shows the determination of the *potential controlling segments* for each activity. The *downward pass* produces a *controlling activity path* very similar to one produced when only CFL activities are involved in the analysis as shown in Figure 60.

The analysis of a *continuous partial-span* activity is a hybrid process. The determination of whether or not the activity is potentially on the controlling path is made like that for a *block* or a *bar* activity. The analysis of the *controlling segment*, however, is made as if the partial-span activity was actually a full-span activity.

Start and End Activities

It was stated earlier that the start time and end time of a project are treated as if a horizontal line across the project at these points in time represented *continuous full-span linear* (CFL) activities. The *Start* and *End* activities do not have a duration. They are zero time activities, and consequently, the production rate (stations/day) is undefined. The only
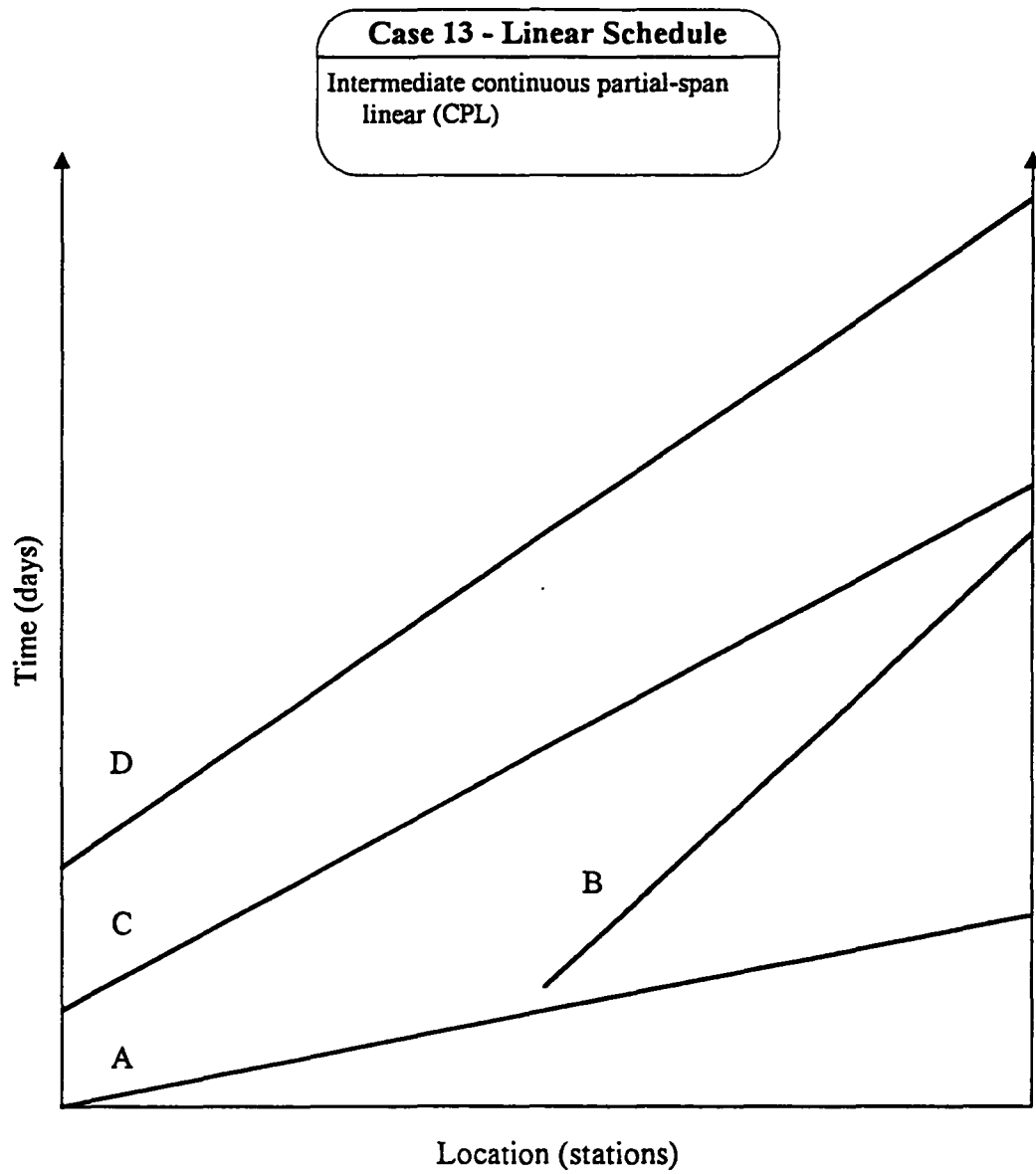
Figure 57 Case 13 - Linear Schedule

Case 13 - Upward Pass

Intermediate continuous partial-span
linear (CPL)

Time (days)

D

C

A

LT

B

LD

Location (stations)
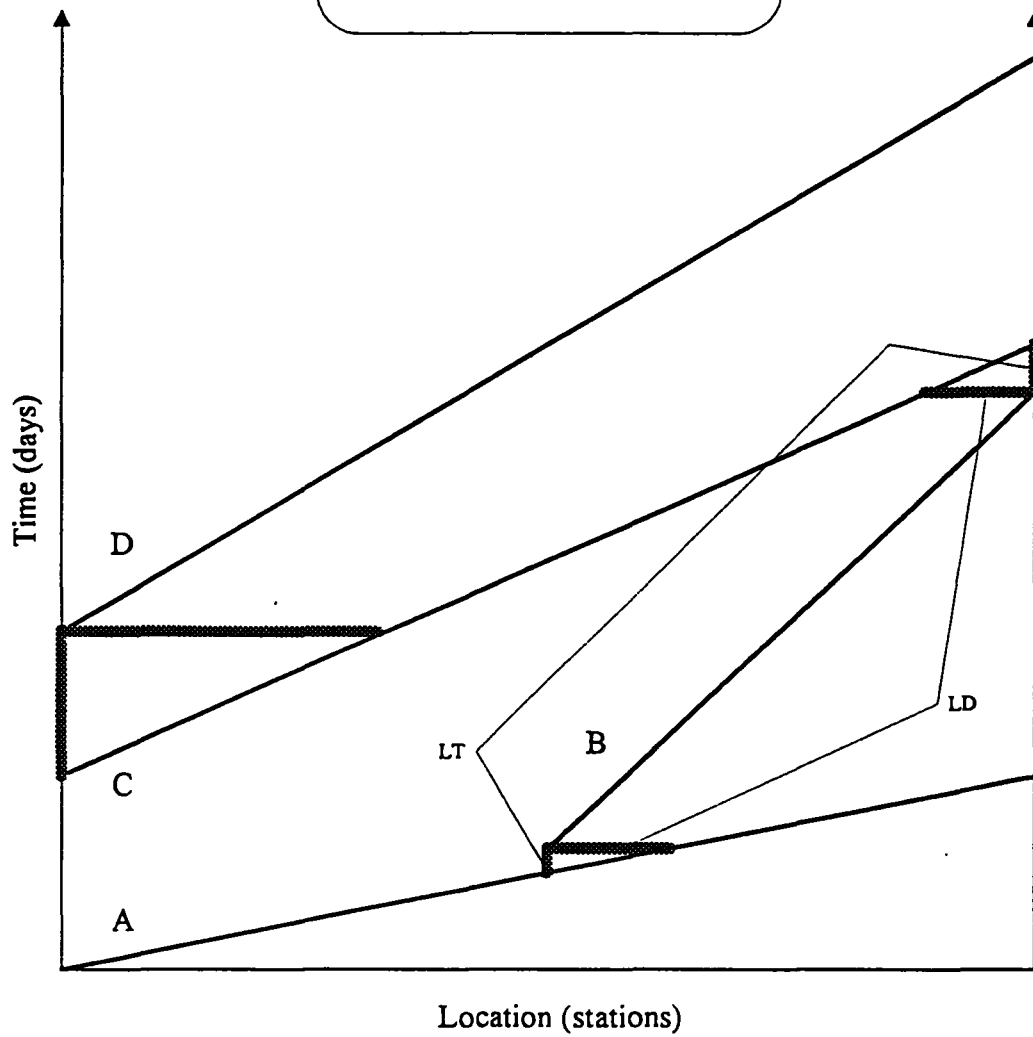
Figure 58 Case 13 - Upward Pass Step 1
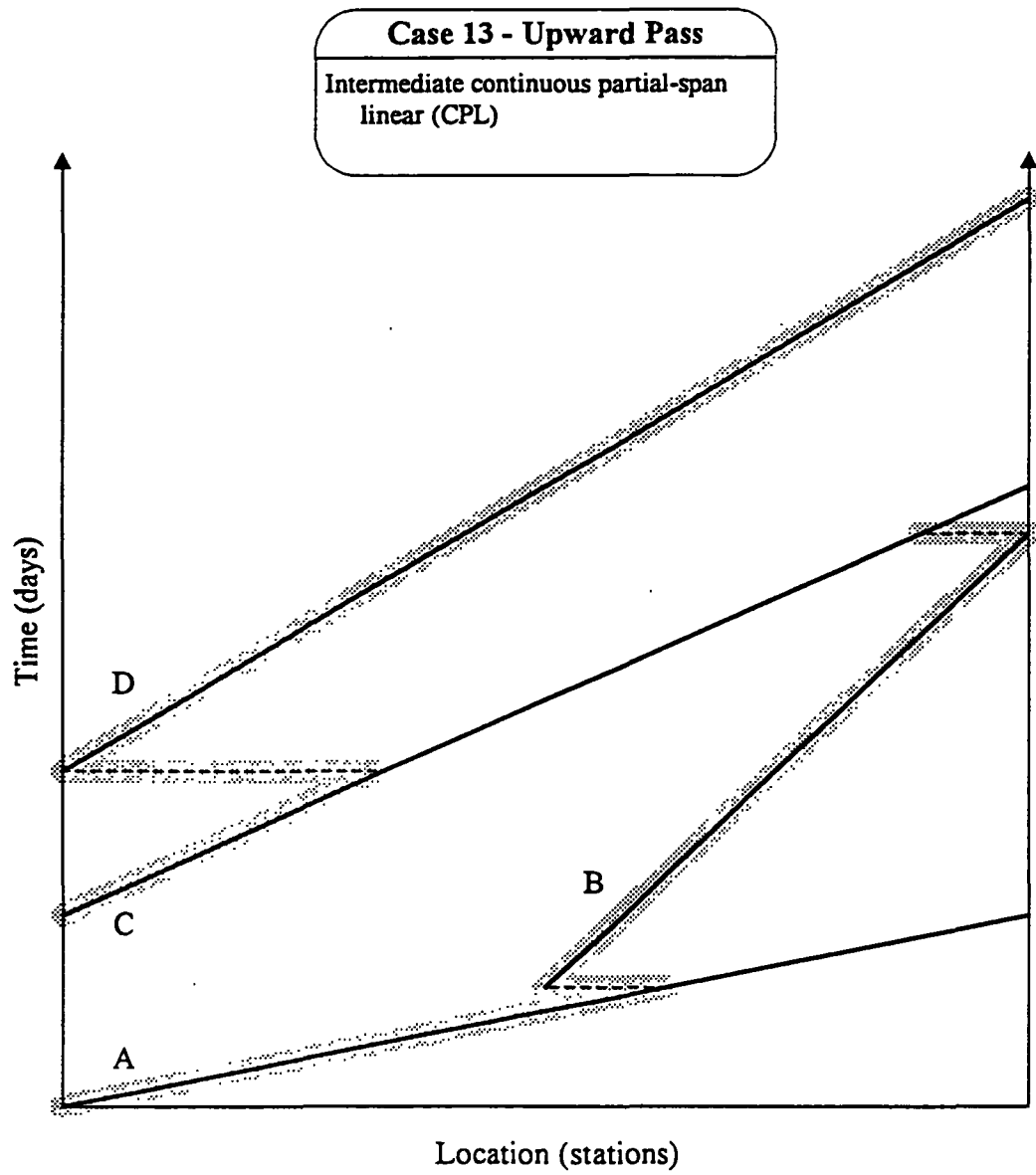
Figure 59 Case 13 - Upward Pass Step 2
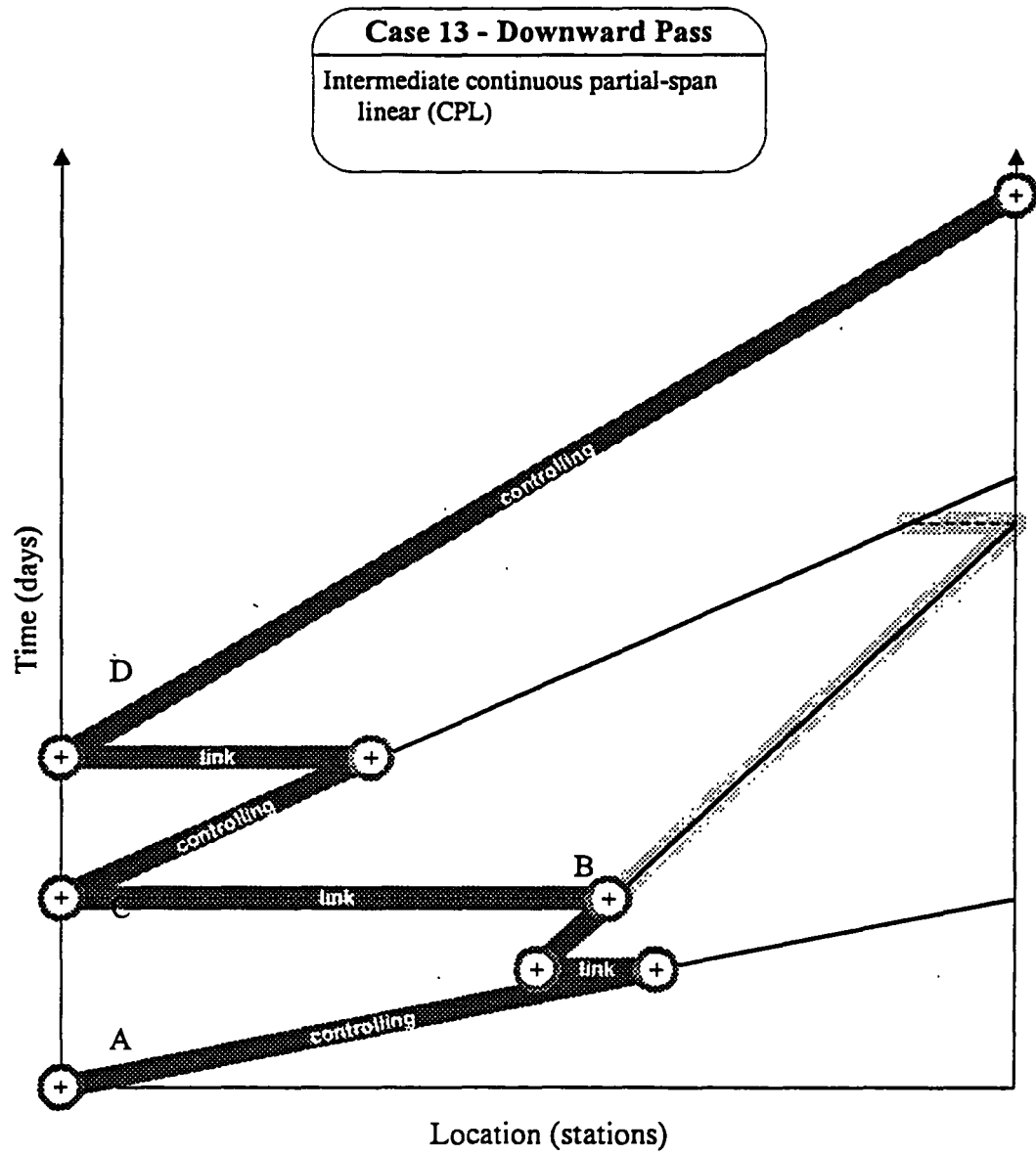
Figure 60 Case 13 - Downward Pass

time that it becomes important to use *Start* or *End* activities in the analysis of the controlling activity path is if either the first or last activity on the project is an intermediate activity. The *Start* or *End* activity can then be used as one of the CFL activities which are needed to "sandwich" the *intermediate activity* to perform the controlling activity analysis. Cases 14 and 15 present examples of how the *Start* and *End* activities are used.

<u>Case 14</u>

Figure 61 shows two examples of where a *block* activity occurs before the first CFL activity on the project. One example shows the *block* activity on the potential controlling activity path and the other example indicates that the block is not on the potential controlling activity path. The analysis is performed as usual, and Figure 62 shows the possible results of the downward pass for each example.

<u>Case 15</u>

This case shows two examples of a *partial-span block* activity as the last activity on a linear schedule. In the upper example on Figure 63, the least time interval (LT) occurs between the last CFL activity and the *End* activity. Since a *coincident duration* does not exist between the last CFL activity and the *End* activity, the entire CFL activity is potentially critical. Any portion of the *block* activity that, at any time, is the only activity in progress, it is automatically a *controlling segment*. Figure 64 illustrates this fact, and shows the result of the *downward pass* for each situation.

Rate Float

This section will explain the analysis of *rate float* for *non-controlling segments* of

Case 14 - Linear Schedule

Intermediate activity earlier than first
  CFL activity
Pseudo start CFL activity

First CFL activity

LT

Location (stations)

If the least time interval
between the intermediate
activity and the first CFL
activity is equal to 0, the
intermediate activity is on
the critical path as long as
the CFL activity is on the
critical path.

First CFL activity

LT

Location (stations)

Time (days)

Figure 61 Case 14 - Linear Schedule

Case 14 - Downward Pass

Intermediate activity earlier than first
    CFL activity
Pseudo start CFL activity

First CFL activity

Location (stations)

Time (days)

link
controlling

If the least time interval
between the intermediate
activity and the first CFL
activity is equal to 0, the
intermediate activity is on
the critical path as long as
the CFL activity is on the
critical path.

First CFL activity

link
controlling

Controlling

Location (stations)

Figure 62 Case 14 - Downward Pass

Case 15 - Linear Schedule

Intermediate activity later than last CFL
  activity
Pseudo end CFL activity

Assume this is a CFL
activity and evaluate
as usual

LT

LD

Location (stations)

Time (days)

Assume this is a CFL
activity and evaluate
as usual

Last CFL activity

LT

Location (stations)
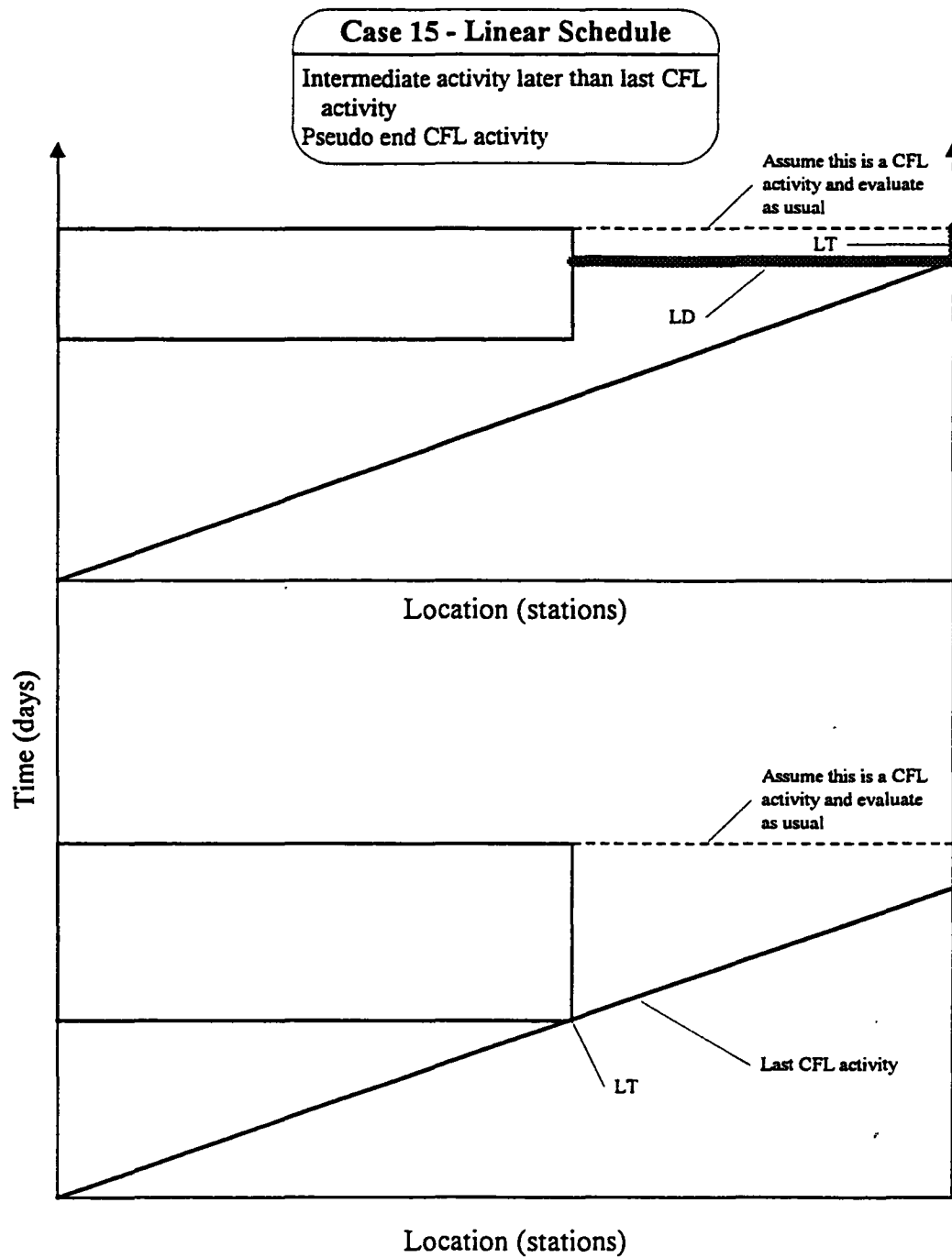
Figure 63 Case 15 - Linear Schedule
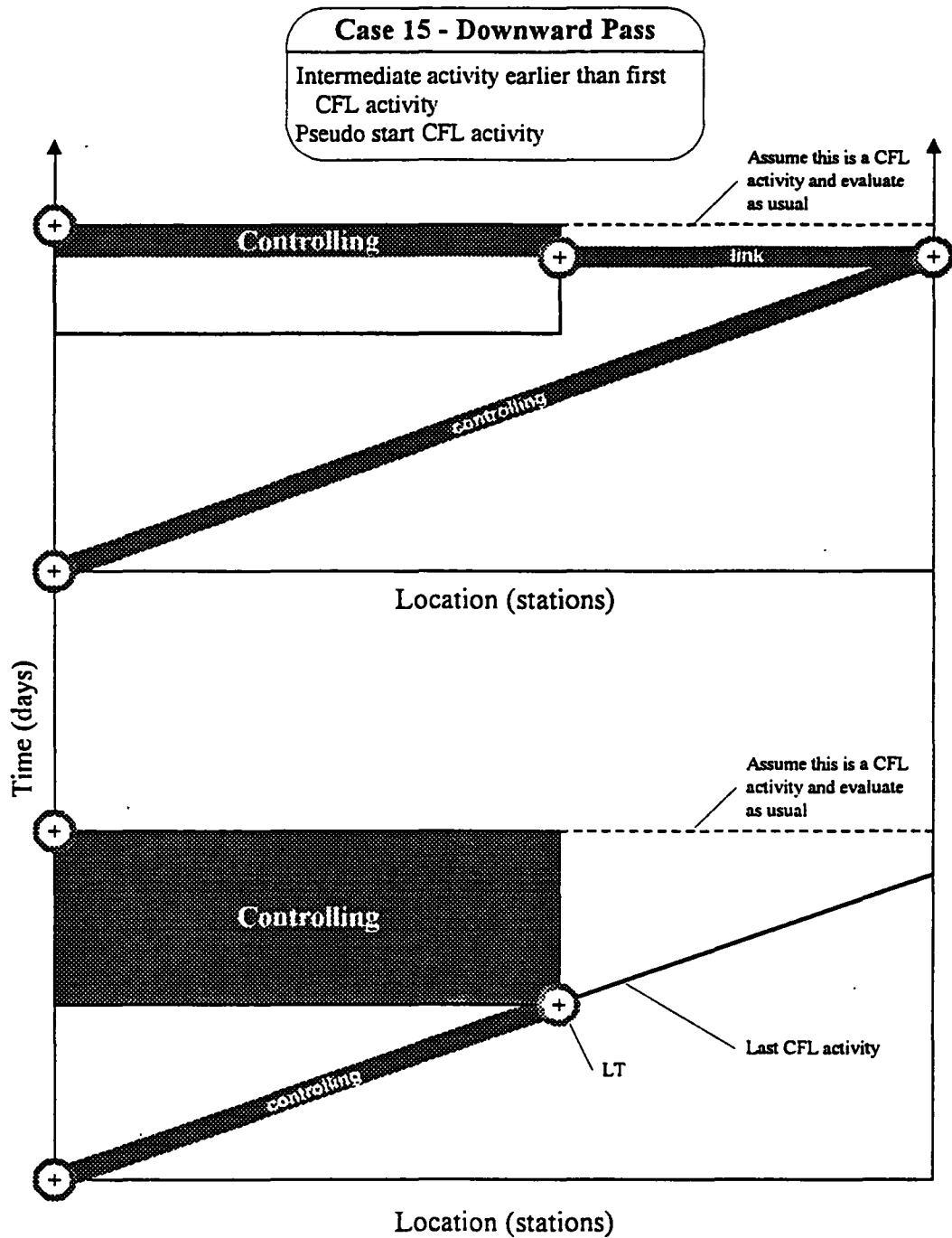
Figure 64 Case 15 - Downward Pass

activities on a linear schedule. Float is a term commonly used in CPM scheduling techniques to describe the amount of time that the duration of a non-critical activity can be extended, or the start of the activity delayed, before it enters the critical path. In the Linear Scheduling Method (LSM), the term *rate float* is used. *Rate float* is the amount the production rate of a *non-controlling* linear activity segment can be lowered before the activity would become a controlling segment. For block and bar type activities, the definition of float is similar to that in CPM.

A linear activity, on a linear schedule, represents the planned start of the activity and the anticipated rate at which it is planned to progress. Given this information and the location that the activity is to cover, the end of the activity is determined. Any point on the linear activity (x,y), can be described as a location and time. The x-coordinate represents a location and the y-coordinate represents a time. Any two points on a linear activity (x_1,y_1) (x_2,y_2) represent a segment of the activity. The slope of this line segment, $(x_2-x_1)/(y_2-y_1)$, represents the rate at which the activity progresses in terms of units of location per units of time.

The slope of any line segment represents the anticipated rate at which the planner expects this activity to progress. In the actual construction of this activity, the production rate may vary from the fixed rate depicted on the linear schedule. To account for this natural variance in production rate, the planner leaves some space between activities on the linear schedule. This space, or buffer, increases as the expected variance in production rate of any particular activity increases. The least time interval, described in the controlling activities section, identifies points where the buffer between activities is at a minimum. For purposes of

calculating activity floats, these buffers can be viewed as points of intersection between adjacent activities. Two activities cannot occupy the same space at the same time, nor can they violate the logic in the activity sequence list. Therefore, the least time interval is the minimum time that can occur between any two adjacent activities on the activity sequence list at any location on the project. This is an important part of determining activity float on a linear schedule.

The upward and downward passes through a linear schedule identify controlling segments of linear activities. The controlling segments can occur anywhere along the path of the linear activity. Portions of a linear activity that are not controlling segments can occur before the controlling segment, after the controlling segment, or both before and after the controlling segment. This means that, with respect to controlling and non-controlling segments, a linear activity, at most, can be divided into three segments. These three possible segments would be a non-controlling segment, followed by a controlling segment, followed by another non-controlling segment. Any combination of these segments is possible, including: a controlling segment only, a non-controlling segment only, a leading non-controlling and controlling segment only, or a leading controlling segment and a non-controlling segment only. To identify these possible activity segments, a convention must be used. A beginning non-controlling segment will be identified as the activity name combined with a minus sign (-) and an ending non-controlling segment will be identified as the activity name combined with a plus sign (+). For example, assume activity A has both a beginning and an ending non-controlling segment. The three segments of A would then become A-, A, and A+. The

original activity name will always remain with the controlling segment. If an activity does not have a controlling segment, the activity name will not change.

Beginning Non-Controlling Segments

The discussion of rate float will begin with activities that have a beginning non-controlling segment followed by a controlling segment. Figure 65 shows several activities of this type; **D-**, **E-**, and **F-**. The analysis of controlling activities on a linear schedule will produce groups of activities that have rate float, much as non-critical paths in CPM schedules produce sets of non-critical activities. On a linear schedule, these groups of activities will be bounded by controlling activities or the limits of the project. In Figure 65 for example, notice that activity **C** does not have a beginning non-controlling segment, while activity **D** does have one. Activity **C** becomes a boundary activity for the subsequent group of activities that have beginning non-controlling segments. These activities are **D-**, **E-**, and **F-**. Since activity **F** is the last activity in this schedule, it defines the end of this group. If, however, activity **F** was followed by an activity that did not have a beginning non-critical segment, this activity would become the upper boundary of this group.

The only way that the rate of progress can occur at a rate lower than the planned rate for a beginning non-controlling segment is if the activity is started earlier than planned. The analysis of beginning non-controlling activities on a linear schedule determines how much sooner an activity can reasonably be started and at what rate it can progress without affecting the overall project duration. The analysis of rate float in beginning non-controlling activities, begins with the earliest activity in the group. In this case, the earliest activity is activity **D-**.
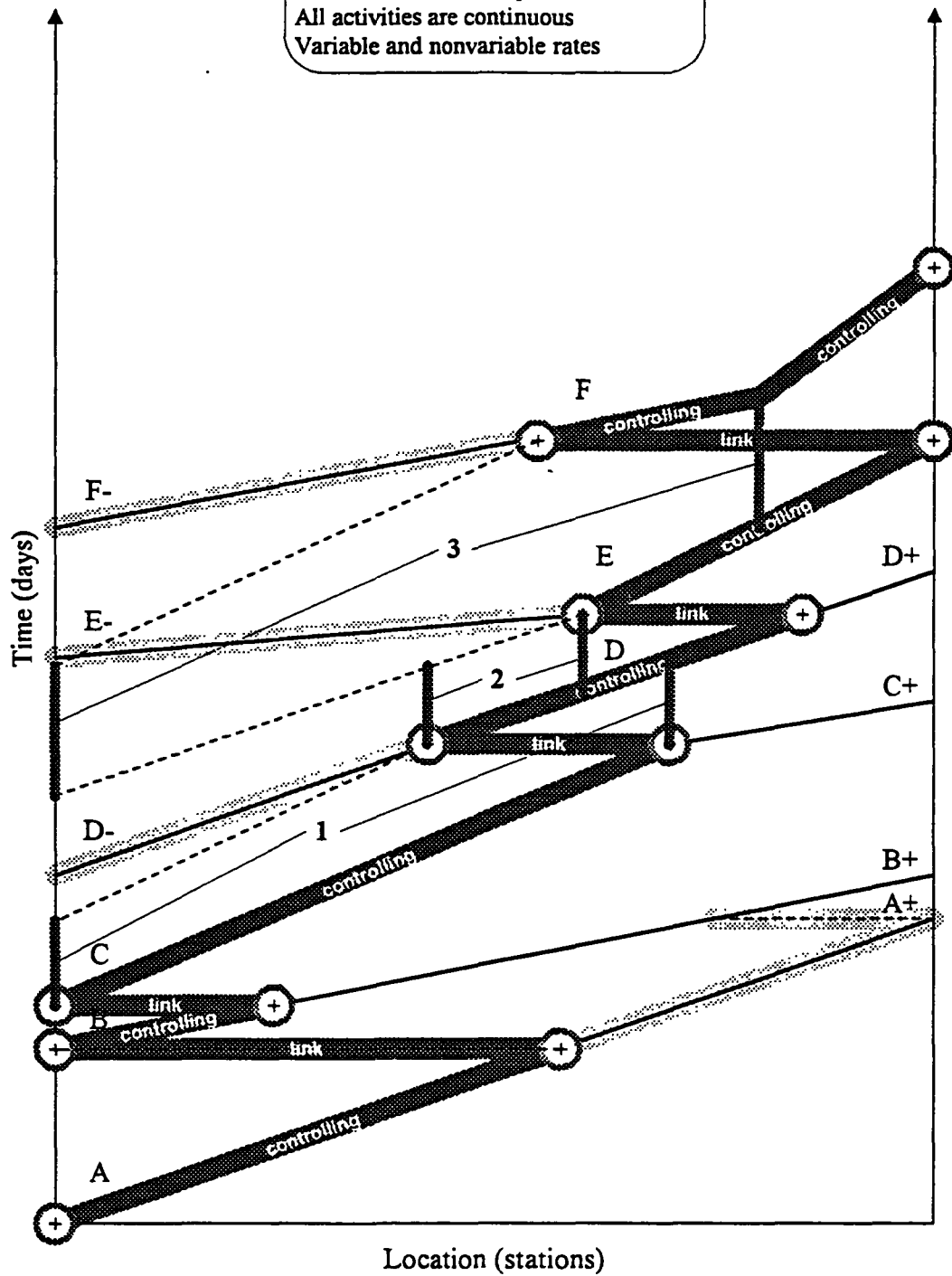
Figure 65 Rate Float for Beginning Activities

The least time interval, between any two activities, defines how close these two activities can be at any location on the project. Item number 1 on Figure 65, shows the original least time interval on the right, and assuming activity D- were to start earlier, it defines, on the left, how soon activity D- can start after activity C starts.

Assume that the original start point of activity D- is (x_{1D-}, y_{1D-}), the end point is (x_{2D-}, y_{2D-}), and the new earliest start time is (x'_{1D-}, y'_{1D-}). In this case, the lowest possible production rate is:

$$\frac{(x_{2D-} - x'_{1D-})}{(y_{2D-} - y'_{1D-})} = Lowest\ possible\ production\ rate \tag{1}$$

The rate float for activity D- is the difference between the planned production rate and the lowest possible production rate:

$$\frac{(x_{2D-} - x_{1D-})}{(y_{2D-} - y_{1D-})} - \frac{(x_{2D-} - x'_{1D-})}{(y_{2D-} - y'_{1D-})} = Rate\ float \tag{2}$$

In general, the rate float for any beginning non-controlling segment of a linear activity is:

$$\frac{(x_{2\alpha} - x_{1\alpha})}{(y_{2\alpha} - y_{1\alpha})} - \frac{(x_{2\alpha} - x'_{1\alpha})}{(y_{2\alpha} - y'_{1\alpha})} = Rate\ float \tag{3}$$

where α is any beginning non-controlling segment

Once the earliest start time for activity D- has been determined the earliest start time

for the next activity, activity E- can be determined. Item number 2 on Figure 65 shows the least time interval on the right and on the left, it shows the location of the least time interval that determines the closest point that can occur between activity E- and activity D. Notice that the interval between the start times of activity E- and activity D- is greater than the least time interval. To determine rate float however, each activity segment must remain one activity segment. Therefore, the earliest start time of E- is later than what may actually be possible.

The last beginning non-controlling activity in this group is activity F-. Item number 3 on Figure 65 shows the original least time interval and the location of the least time interval between F- and E. In this case, the earliest start of activity F- is determined by the earliest start date of activity E-. The earliest start time of E- is, again, the time found by the analysis of rate float and not the absolute earliest start time, as the figure suggests.

Beginning non-controlling linear activities only have rate float if they are started earlier than the time indicated on the linear schedule. The controlling activity analysis only identifies which activity segments could potentially be started earlier and progress at a rate lower than that planned. If the contractor chooses to construct the project as indicated on the linear schedule, then the beginning non-controlling segments could potentially be viewed as controlling segments because, in effect, they must progress at the planned rate to be at the location when the activity becomes controlling as indicated on the linear schedule. Also, the impact of not starting any beginning non-controlling activity as early as indicated by the earliest possible start time on subsequent activities must be carefully evaluated.

Ending Non-Controlling Segments

The other portion of a linear activity where non-controlling segments can occur is at the end of the activity. This is an *ending non-controlling segment*. Figure 66 shows an example of several activities that have *ending non-controlling segments*: **A+**, **B+**, **C+**, and **D+**. On an *ending non-controlling segment*, the start point of the segment is fixed by the end of the controlling segment. The end point of the segment, however, can actually occur at a later time than that indicated on the linear schedule. The fact that the end of the activity can be delayed, lowers the rate at which the activity segment needs to progress. The amount that the production rate can be lowered defines the rate float for this activity segment.

As with beginning non-controlling segments, ending non-controlling segments can occur in groups of one or more bounded by ending controlling activity segments, or the start or finish activities on a project. Notice on Figure 66, that the *ending non-controlling segments*, **A+**, **B+**, **C+**, and **D+**, are bounded by the start of the project and the ending controlling activity segment **E**.

The only way that the rate of progress can occur at a rate lower than the planned rate for an *ending non-controlling segment*, is if the activity is completed later than planned. The analysis of *ending non-controlling activities* on a linear schedule determines how much later the activity can reasonably be completed, and at what rate it can progress, without affecting the overall project duration. The analysis of rate float in ending *non-controlling activities* begins with the latest activity in the group. In the case of the four previously mentioned activities, that is activity **D+**. As before, the least time interval defines how close any two

124
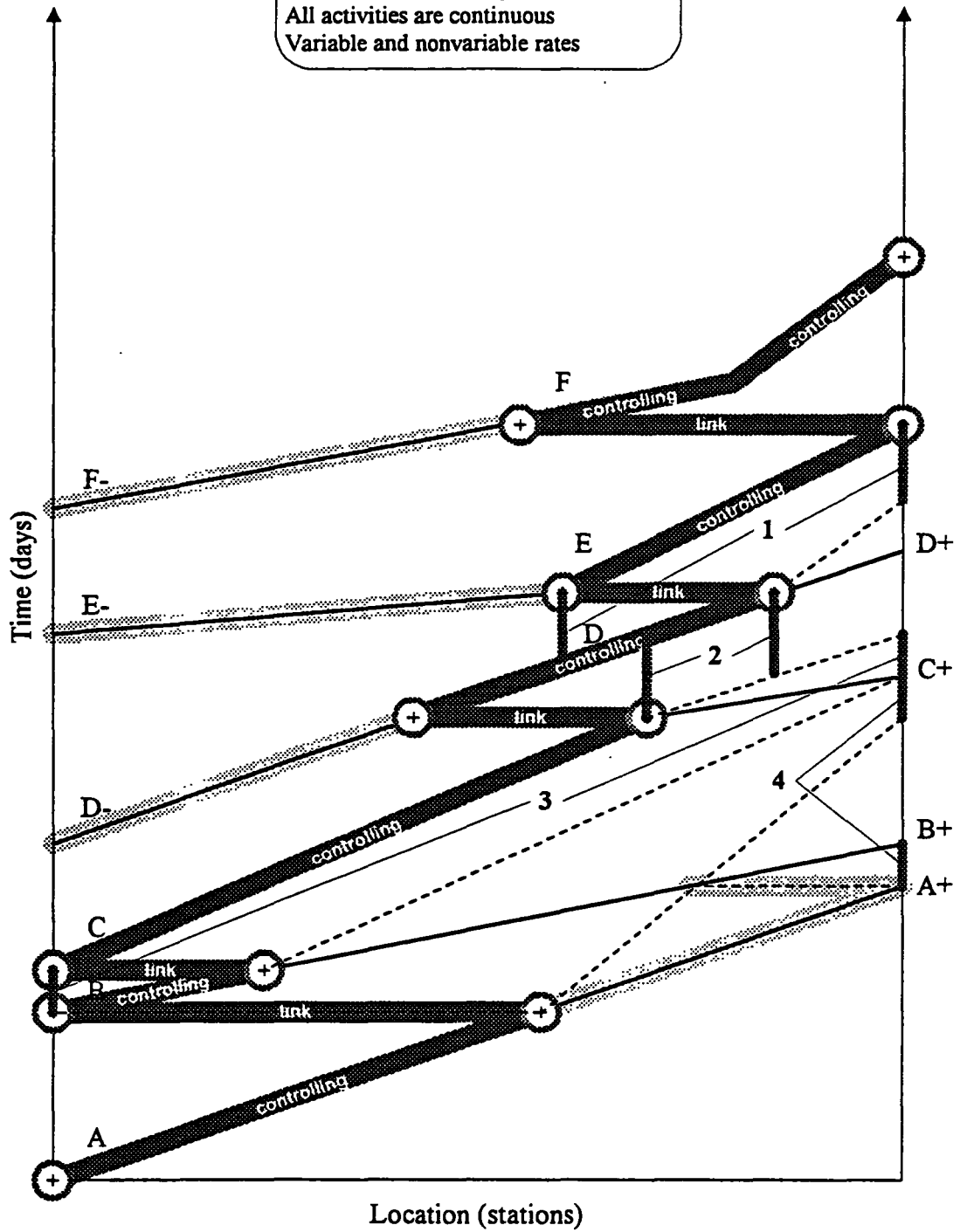
Figure 66 Rate Float for Ending Activities

activities can occur at any location on the project. Item number 1 on Figure 66 shows the original least time interval on the left and, on the right the least time interval that determines how late activity D+ can possibly finish.

Assume that the original start point of activity D- is (x_{1D+}, y_{1D+}), the end point is (x_{2D+}, y_{2D+}), and the new latest end time is (x'_{2D+}, y'_{2D+}). In this case, the lowest possible production rate is:

$$\frac{(x'_{2D+} - x_{1D+})}{(y'_{2D+} - y_{1D+})} = Lowest\ possible\ production\ rate \tag{4}$$

The rate float for activity D+ is the difference between the planned production rate and the lowest possible production rate:

$$\frac{(x_{2D+} - x_{1D+})}{(y_{2D+} - y_{1D+})} - \frac{(x'_{2D+} - x_{1D+})}{(y'_{2D+} - y_{1D+})} = Rate\ float \tag{5}$$

In general, the rate float for any ending non-controlling segment of a linear activity is:

$$\frac{(x_{2\beta} - x_{1\beta})}{(y_{2\beta} - y_{1\beta})} - \frac{(x'_{2\beta} - x_{1\beta})}{(y'_{2\beta} - y_{1\beta})} = Rate\ float \tag{6}$$

where β is any ending non-controlling segment

Once the latest completion time for activity D+ has been determined, the latest completion time for the preceding activity in the group can be determined. Item number 2 on

Figure 66 shows the position of the original least time interval and the least time interval that controls the rate float for activity C+. Since the interval occurs at a location other than the end of the activity, the amount of time between the completion of activity C+ and activity D+ is greater than the least time interval. The reason for this difference in completion time is that activity C+ must remain a single line segment and cannot be closer to D+, at any location, than the least time interval. Items number 3 and 4 on Figure 66 indicate the least time interval locations to determine the latest completion times for activities B+ and A+ respectively.

Ending non-controlling segments always have rate float which is defined by the latest possible completion time. It may also be possible to delay the start of an ending non-controlling segment and then progress at a rate higher than the lowest possible rate. This possibility, however, is not defined in the rate float calculation. The assumption is, that when the controlling segment ends, the non-controlling segment will begin and can progress at a lower rate than planned without affecting the overall project duration. Delaying the start of a non-controlling activity would need to be carefully evaluated for each specific instance to determine the possible effects on subsequent activities.

Intermediate Activity Float Analysis

Block and bar type activities can have float, but it is not rate float since the production rate of these activities cannot be ascertained from the linear schedule. Float in a block or bar type activity can occur either before, after, or before and after the activity. Depending on the situation, this float can be utilized in several ways:

1. The activity could be performed at a later point in time.

2. The duration of the activity could be extended.

3. The activity could start at an earlier time.

Figure 67 presents an example of float analysis for an intermediate activity. This figure was examined earlier as an example of determining controlling activity segments for intermediate activities. Recall that when performing the controlling activity segment analysis, the least time intervals indicated by item number 1 were compared to determine the potential controlling segments. When analyzing float for intermediate activities, however, the least time interval has less influence than it does when comparing continuous full-span linear activities. One approach, would be to stipulate that the least time interval between activity A and activity C must be less than the sum of the least time intervals between activities A, B, and C. The difficulty arises in determining whether the buffer occurs before, after, or both before and after, activity B. For intermediate activities, a better solution is to ignore the least time interval, and calculate float by eliminating all buffers between the affected activities.

Figure 68, shows the result of eliminating the buffers between activities A+, B, and C, indicating the maximum amount of float in these activities. The constructors of this project must realize, however, that the risk of affecting the controlling path with the non-controlling activities increases substantially as the indicated situation of float reduction is approached.

The next two examples show some other possible situations that could occur in analyzing float in intermediate activities. Figure 69 shows a block activity in which only a portion of the activity is on the controlling activity path. The dashed lines indicate that the duration of the upper portion of the block activity could be extended until it contacts the CFL
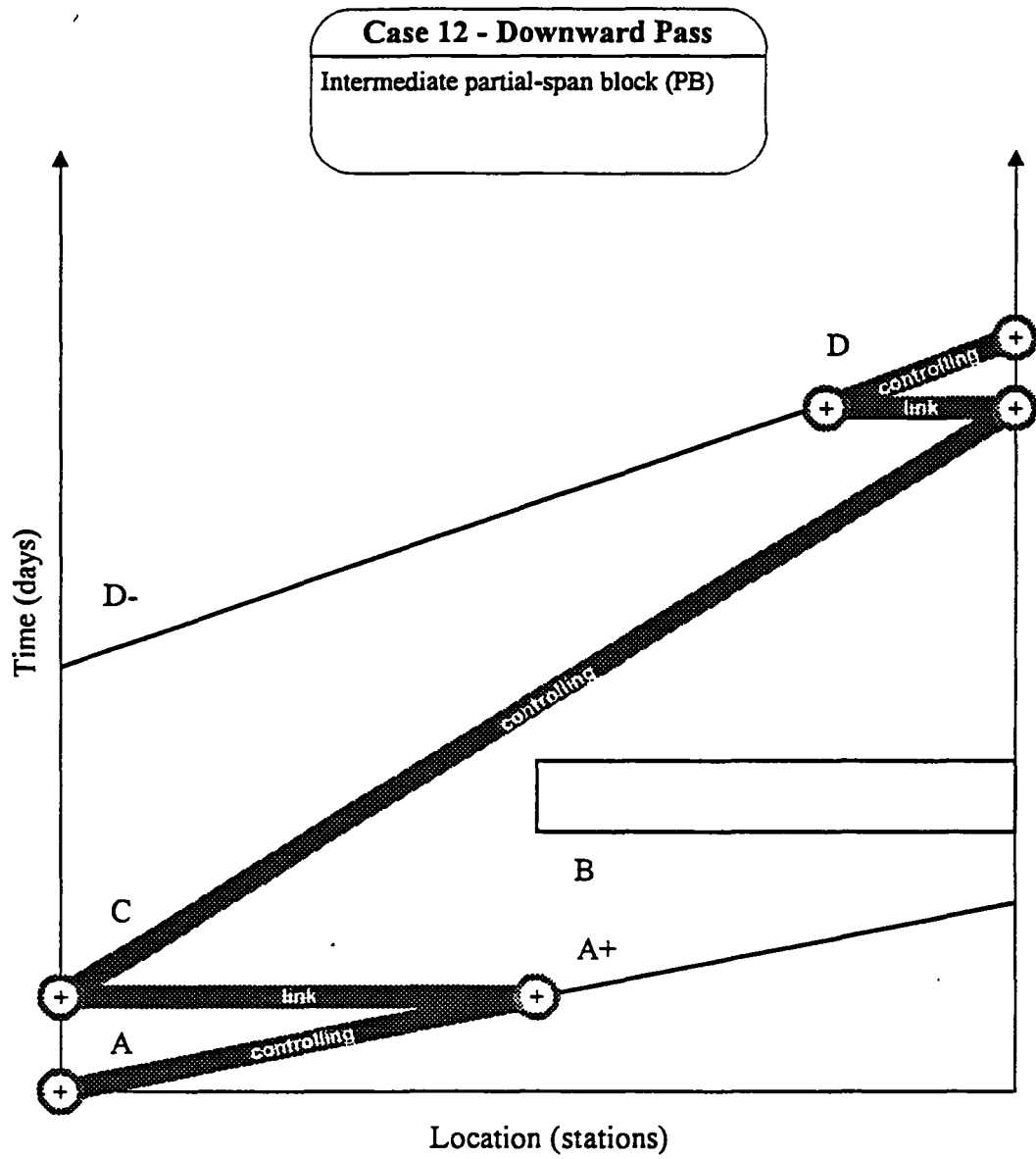
Figure 67 Float for a PB Activity
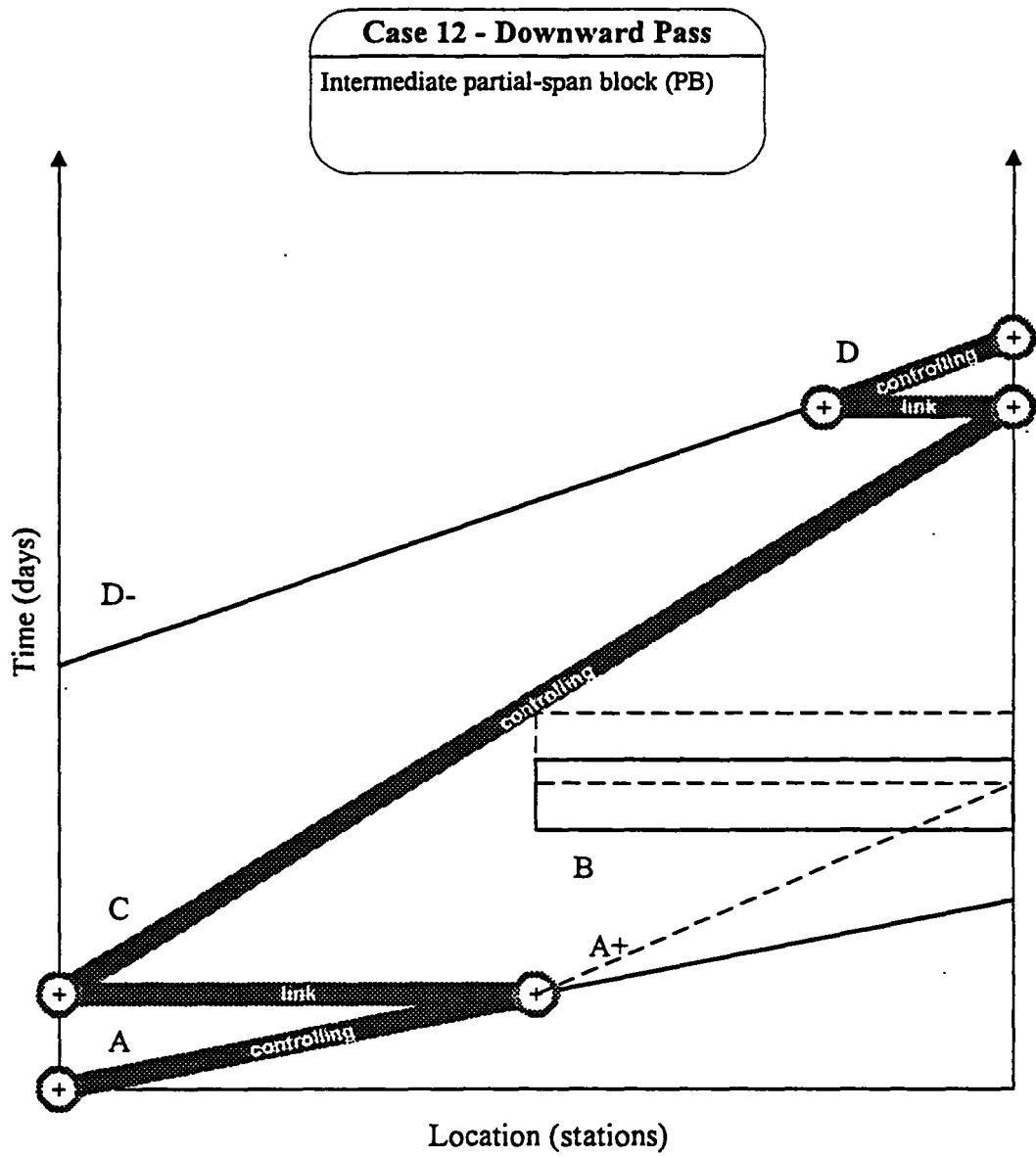
Figure 68 Float Analysis for a PB Activity
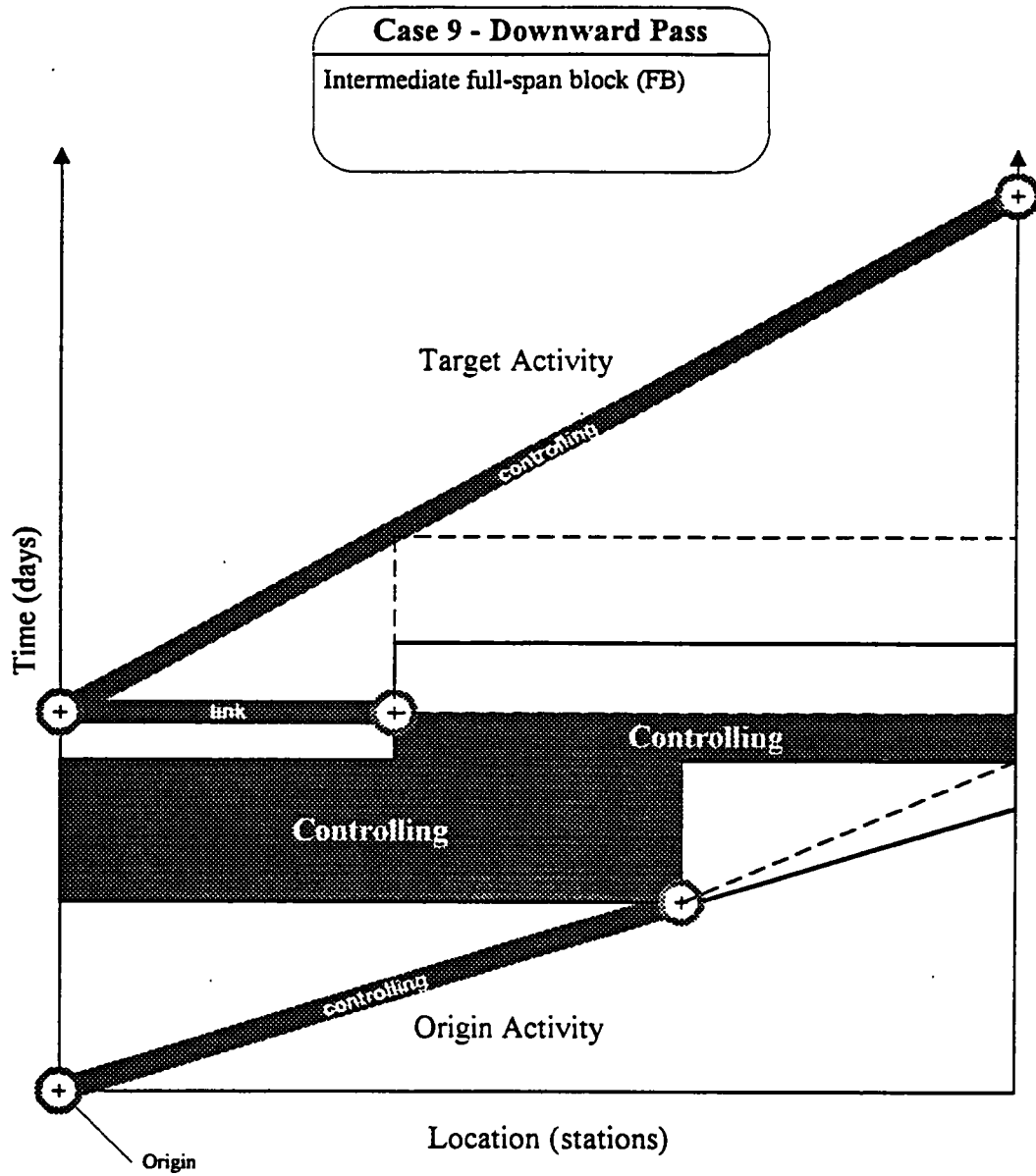
Figure 69 Float Analysis for a FB Activity

activity . The figure also indicates that the ending segment of the origin activity has some rate float. Figure 70 shows the float analysis in a group of ending non-controlling activity segments. As discussed earlier, the analysis of ending non-controlling activities begins with the latest activity in the group. The latest activity is C+ and, since activity D is a CFL activity,the float analysis is accomplished by the method provided for CFL activities utilizing the least time interval indicated by item number 1 on the figure. Preceding activities are moved up in time as far as possible, as indicated by the dashed lines, to determine the amount of float or rate float available to each.

As-Built Schedules

A linear as-built schedule is simply a record of the work progress on the project. Since the chart used in the development of the linear schedule provides a visual planning area of the project location and time, this same chart can be used to plot the progress of each activity on the project. A linear as-built schedule produced in this way, provides a method of capturing all of the significant events on a project and a method of communicating this information to others in a form that is easily comprehended. Linear as-built schedules provide, at a glance, an overview of the entire project and, upon closer examination, are found to contain daily progress data for each activity on the project. A linear as-built schedule is an excellent example of a method of visualizing the relationships in large amounts of data. Relationships that are difficult, if not impossible, to understand by any other means can easily be visualized in a linear schedule..

The creation of the as-built schedule is quite simple. For each day any activity is in
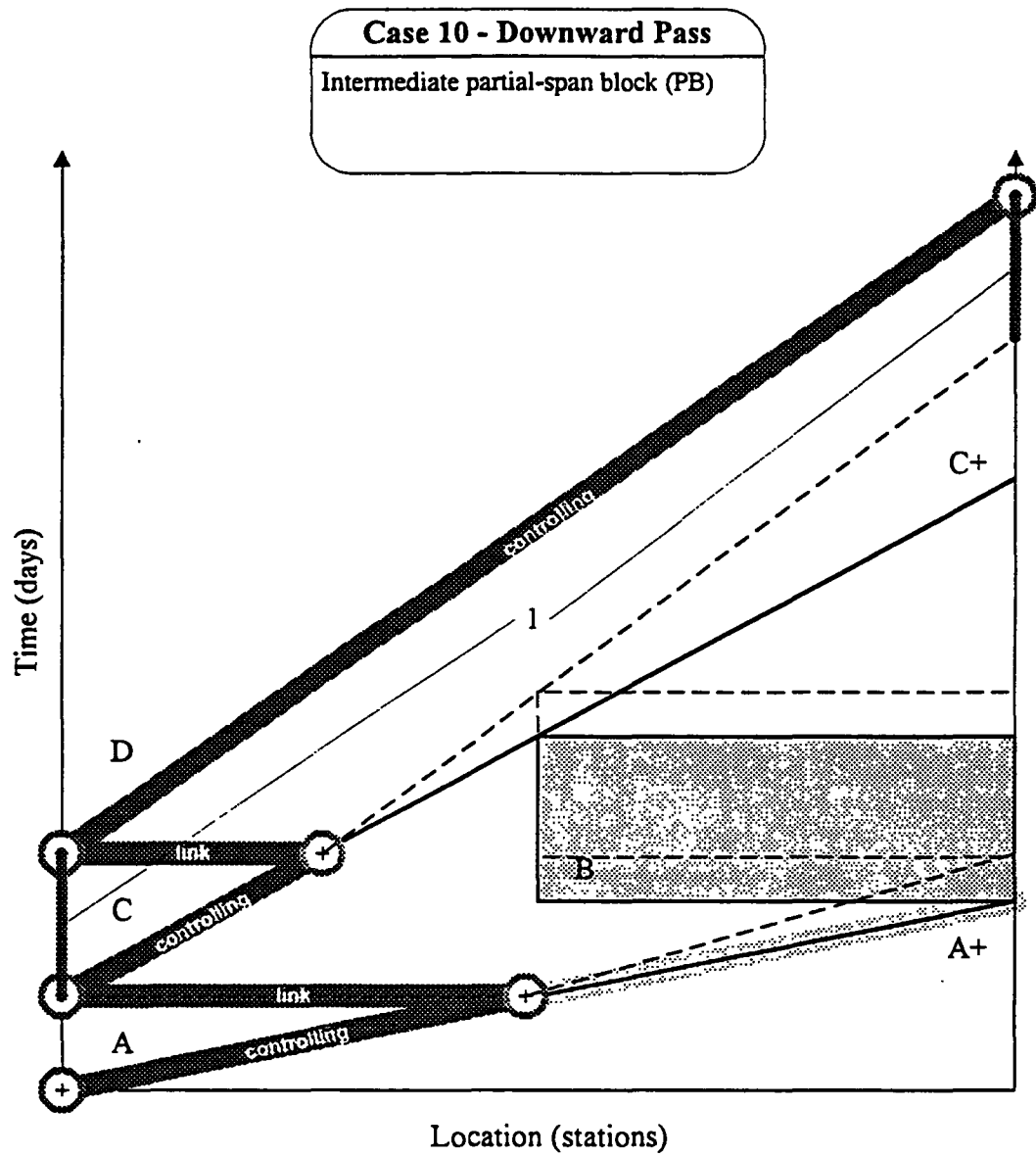
Figure 70 Float Analysis for Ending Activities

progress, the location of the progress is recorded on the linear schedule. For example, if the activity pavement removal progressed from station 1000 to station 2000 on day 10 of the project, a line would be drawn from point (1000, 10) to (2000, 11). The time span from 10 to 11 describes the duration of the 10th day of the project.

Besides the work progression, other information can be recorded on the schedule. Highway construction projects typically involve a substantial amount of utility work that is performed by others not under contract to the owner. The timely completion of utility work can have a significant impact on the project. The ability to quantify this impact, however, can be difficult. If the completion of work performed by others is captured on the as-built schedule, the visual record created, readily demonstrates the impact on any project activities. Any influence on the project can be depicted on the linear as-built schedule, even the influence of a flood. The result is an extremely detailed visual record of the entire project.

The section on the research projects for the Iowa Department of Transportation provided examples of as-built schedules for the three projects on which linear scheduling field tests were performed. Figures 3, 4 , and 5 show linear as-built schedules for projects I, II, and III, respectively. These schedules provide a detailed record of all the work that was performed on the project. They describe, on a daily basis, what activities were in progress and the number of units completed. Since a horizontal line across the schedule defines a point in time, all of the activities in progress and the location of each of these activities can be easily visualized. Without this record, the determination of the activities in progress during any particular time period, requires the careful examination of extensive amounts of field

inspection data.

The linear as-built schedule also provides information regarding the influence of the progress of any specific activity on any subsequent activities. Although, this information is impossible to ascertain using other scheduling techniques, the information can be determined visually from the linear as-built schedule. As the linear as-built schedules prepared for the Iowa Department of Transportation projects demonstrate, the specific project circumstances determine the information that needs to be captured on the linear as-built schedule. Since each project has a unique set of circumstances, the preparer of the schedule must find ways to explain the events of the project using the linear as-built technique.

Calendars

Calendars are an important aspect of scheduling. In the Linear Scheduling Model (LSM) there are several calendars that are used for different reasons. The research conducted for the Iowa Department of Transportation (IDOT) established the need for various calendars in highway construction work. The calendars that may be necessary for highway construction work are:

1. Date (normal) calendar

2. Planned work days

3. Charged work days

4. Closure days

Date (normal) Calendar

This is the normal everyday calendar that provides dates (month, day, year) for each

unit on the y-axis of the linear schedule. Each unit on the y-axis corresponds to one day on the date calendar. The beginning of the y-axis is the beginning of the project, and first day of scheduled work should coincide with the start date for the project. The y-axis extends upward to include enough units to reach the planned end of the project. If the project duration extends beyond the planned duration of the project, the number of units on the y-axis may need to be increased.

Planned Work Day Calendar

A planned work day calendar identifies those days that the contractor anticipates working. If the contractor plans on working a five days a week, for example, the work days would likely be Monday, Tuesday, Wednesday, Thursday, and Friday. Saturday and Sunday would not be considered as work days. Holidays and other non-work days would also need to excluded from the planned work day calendar. The main function of the planned work day calendar is to associate activity start times, durations, and end times to the date calendar. For example, consider an activity that starts on the first day of the project and has a duration of five days. Assume that the project start date is May 1, 1995, and that the contractor is using a five day work week. Activity durations are given in working days so, in this case, the activity will be in progress for 10 work days. May 1, 1995, is a Monday. Therefore, since a five day week is being used, Saturdays and Sundays are non-work days. The activity is planned to be complete, at the end of the day, on Friday, May 12, 1995. The work day calendar's function is to relate the number of planned work days to actual calendar dates by spanning the non-work days.

Charged Work Day Calendar

The charged work day calendar relates to specific contract requirements for certain highway construction projects. In Iowa certain projects with a significant impact on the motoring public will have an incentive/disincentive clause. On these projects, a specific number of working days are allowed in which to complete the project. If the contractor completes the work in less time than allotted by IDOT, a bonus or incentive is paid to the contractor for each day saved. On the other hand, if the contractor goes over the number of working days allowed, the contractor is accessed a penalty or disincentive for each day over.

The criteria to determine whether or not a day is a charged working day is the contractor's ability to perform work at a reasonable rate on the controlling activity(s). If the contractor is not able to work on the controlling activity because of rain, for example, the day is not counted as a charged working day. The purpose of the charged work day calendar is to . relate activity progress to the planned work day calendar to determine an activity's completion status. Completion status refers to whether or not the activity is at the proper point of completion at any time during the activity's duration. For example, let's say that the 10 day activity mentioned earlier was completed on May 17, 1995. To determine whether or not the activity was completed on time, the amount of charged days used needs to be compared to the number of planned day for the activity. Assume that from May 1, 1995 through May 17,1995, only nine days had been charged against the project. This means that even though the activity finished three planned working days later than expected, it in effect finished one day ahead of schedule.

Closure Days

On interstate and divided highway construction projects, it is sometimes necessary to place both lanes of traffic one side of the median. The term for this is head-to-head traffic, and is the result of closing one set of lanes. This has a direct impact on the public in terms of inconvenience and a greater risk of an accident occurring. The Iowa Department of Transportation tries to limit the time that traffic is head-to-head by allowing the contractor a limited number of closure days. Severe penalties are accessed for using more closure days than allowed.

Typically, during a closure period, all days are counted as closure days, even if the contractor is not able to work. Only under extreme conditions is the contractor awarded relief from the closure day restriction. The planner must be sure that all of the activities that need to be performed during the allowed closure days are properly scheduled. During this time, all days are counted as charged days whether or not the contractor is able to work or even planned to work. In this instance charged work days could be greater than planned work days. In the previous discussion of planned work days, the charged days could equal planned days, but should never exceed them.

Calendar Example

Figure 71 is an example of the various calendars use in the Linear Scheduling Model. The units of the y-axis are days and there is a scale that indicates the number of days from the beginning of the project. To the far left is a scale indicating the calendar day. This scale indicates that the first day of the project is Monday, May 1, 1995. On the left hand y-axis,
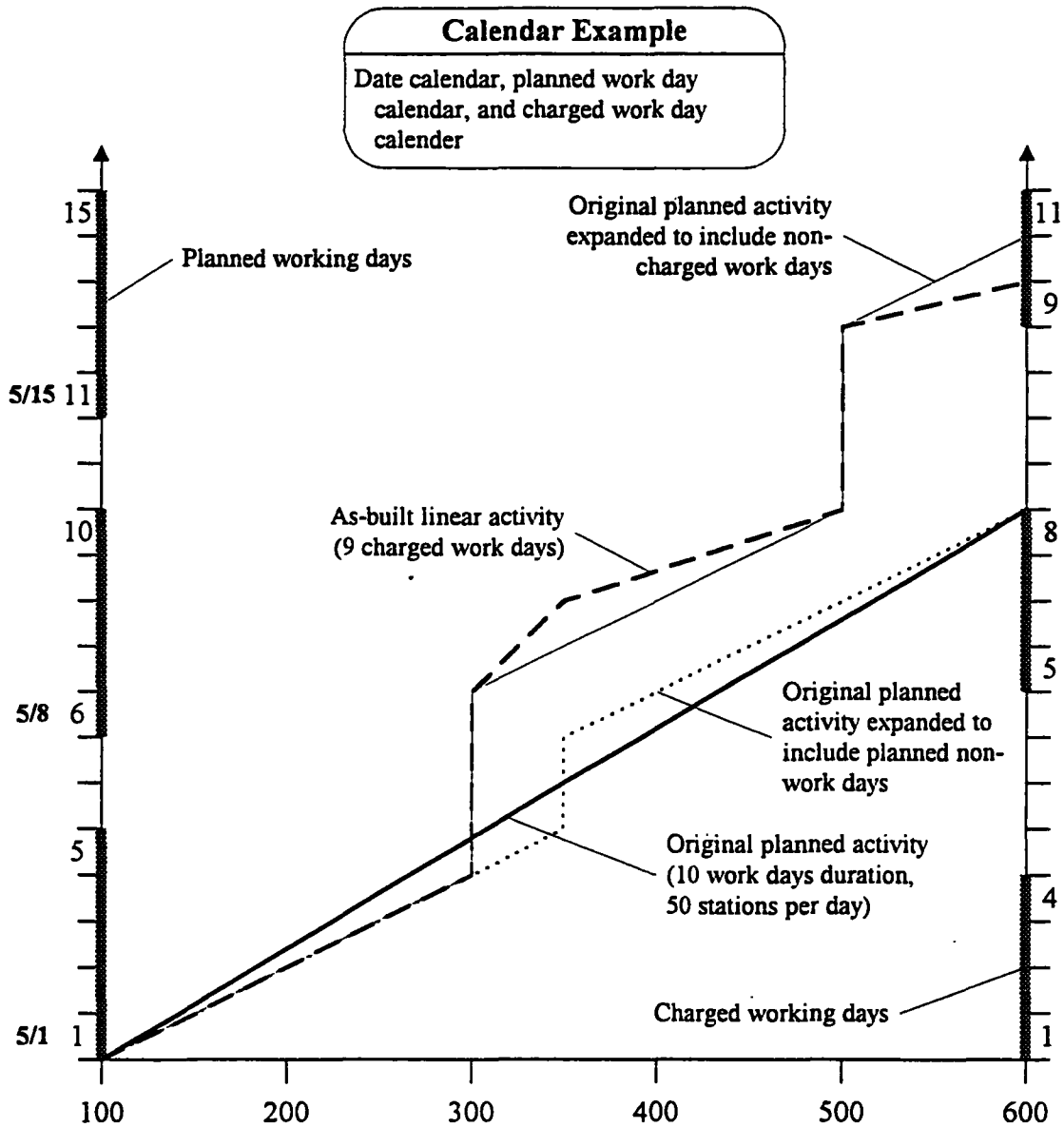
Figure 71 Calendar Example

notice that some of the days are marked by a shaded bar. The shaded bar represents planned work days. In this case, a five day work week was used. The bold solid activity line on the chart represents the original planned activity duration. This activity begins on day 1 at station 100 and progresses to station 600 at the end of planned work day 10. The dotted line shows this same activity with no progress on the non-work days. If the activity were to be completed exactly as planned, the as-built line for this activity would look like this dotted line. On the right hand y-axis there are shaded portions similar to those on the left hand y-axis. These shaded portions indicate the days that were actually charged as working days. In effect, these charged working days revise the planned working day calendar. The thin solid line indicates how the planned activity should look if it were to be completed as expected using the charged day calendar in place of the working day calendar. The last line on the chart is the bold dashed line which closely follows the thin solid line just mentioned. The bold dashed line is the actual progress of the activity, or the as-built line for this activity. Notice that the as-built line indicates that the activity was completed in 9 days, one day less than the 10 days that were planned for the activity. This activity was completed 1 day ahead of schedule, even though it was completed five calendar days later than planned. The next section explains, in more detail, how the progress of activities can be evaluated.

Progress Analysis and Reporting

The last example showed an example of several of the calendars and how they are used to record an activity's progress. The information contained in the schedule for any particular activity can be evaluated to provide an analysis of the progress of the activity. Table 3 shows

Table 3 Activity Progress Spreadsheet

Day	Planned Work Days	Charged Work Days	Revised Planned Work Days	Planned Progress	Cumulative Planned Progress	Actual Progress	Cumulative Actual Progress	Schedule Comparison
1	1	x	1	50	50	50	50	0
2	2	x	2	50	100	50	100	0
3	3	x	3	50	150	50	150	0
4	4	x	4	50	200	50	200	0
5	5				200		200	0
6					200		200	0
7					200		200	0
8	6				200		200	0
9	7	x	5	50	250	25	225	-25
10	8	x	6	50	300	25	250	-50
11	9	x	7	50	350	75	325	-25
12	10	x	8	50	400	75	400	0
13					400		400	0
14					400		400	0
15					400		400	0
16					400		400	0
17		x	9	50	450	100	500	50
18		x	10	50	500			
19		x						
20								

a spreadsheet of the data contained in the linear schedule in the previous example. The first column shows the schedule day and corresponds to the y-axis scale. The second column corresponds to the shaded bars on the left-hand y-axis, and indicates the days during which the activity is planned to be in progress. Notice that the activity has a 10-day duration, and a 5-day work week is being used. The next column is charged work days. This column represents the days that the contractor was able to work on a controlling activity and, therefore, was charged as a work day. Notice that the contractor was not able to work on every day that they planned to work. The revised planned work day column shows how the planned work days have been moved to meet the actual days that the contractor was able to work, and subsequently charged as a working day.

The revised planned work day calendar now becomes the scale by which progress can be measured for the activity in question. Given the revised planned work days, the next column, planned progress, indicates the number of units (stations) that are expected to be completed by this activity. Since the activity is represented by a single line segment, the actual production rate is 50 stations per day. The planned progress column relates to the thin solid line in Figure 71 which represents the original planned activity with the non-charged work days removed. The next column shows a cumulative sum of the planned progress of the activity. The activity covers a total of 500 stations. The column labeled actual progress indicated that actual units completed as the activity was built. This column relates to the as-built line in Figure 71 The next column, cumulative actual progress, shows the cumulative sum of the completed units for the actual construction of the activity. Notice that the activity

was complete (500 units completed) on revised planned work day 9. This indicates that the activity was completed in 9 days rather than the 10 days planned. The last column shows the comparison between the cumulative actual progress and the cumulative planned progress. The zeros indicate that the progress was exactly as expected. On revised planned work days 5 and 6, the actual progress is 25 units instead of the 50 units planned. The schedule comparison shows that the actual progress is 25 units behind on revised planned work day 5 and 50 units behind on revised planned work day 6. During the next two days, the actual progress is 75 units instead of the 50 units planned. By the end of these two days the actual progress has caught up to the planned progress as the schedule comparison column indicates. The number 50 on revised planned work day 9 indicates that the progress was 50 units ahead of schedule on this day.

The previous example shows how the progress of any one activity can be plotted, analyzed, and reported. The project managers, however, need information relating to the entire project, as well as to individual activities. The next section deals with project status analysis and reporting.

Project Updating and Status Reporting

The principle question that can be answered with project updating is, "When will the project be complete?" Contractors and owners, alike, are typically concerned with the issue of when will the project be complete. They need to know if the project is being constructed on-schedule, and if not, how many days early or late will it be finished. The linear schedule can be used as a tool to predict when the project will be complete. The schedule can also

predict when individual activities will be complete.

An example of a simple linear schedule will be used to demonstrate and explain how a linear schedule is updated and the type of activity status information that can be obtained. Figures 72 and 73 show a linear schedule and the controlling activity path, respectively. The y-axes scale on these schedules is working days, and the x-axis is scaled in stations. All of the activities are continuous full-span linear activities, with the exception of activity C, which is a full-span block. For purposes of project control and management, however, the activities need to be represented in calendar days. Assume that a 5 day work week was used to develop the planned schedule. Work days include Monday through Friday each week with no work planned for Saturday and Sunday or holidays.

Figure 74 shows the initial schedule after it has been expanded to include the non-work periods of Saturday and Sunday for each week. The y-axes have a scale representing calendar days, beginning with the first day of the project, Monday, May 1, 1995. The shaded bars on the left-hand y-axis show the planned work days, and the associated numbers indicate planned work days. Notice that activities which span non-work periods have vertical sections in them to represent points along the activity where time advances without any work progression. A tabular spreadsheet of the activity data contained on the linear schedule in Figure 74 can be seen in Table 4. The table shows the planned start and finish of each activity in work days and in calendar dates; the duration of each activity in work days, and the location in stations that each activity covers. The information presented so far represents the contractor's plan. The project has not yet started.
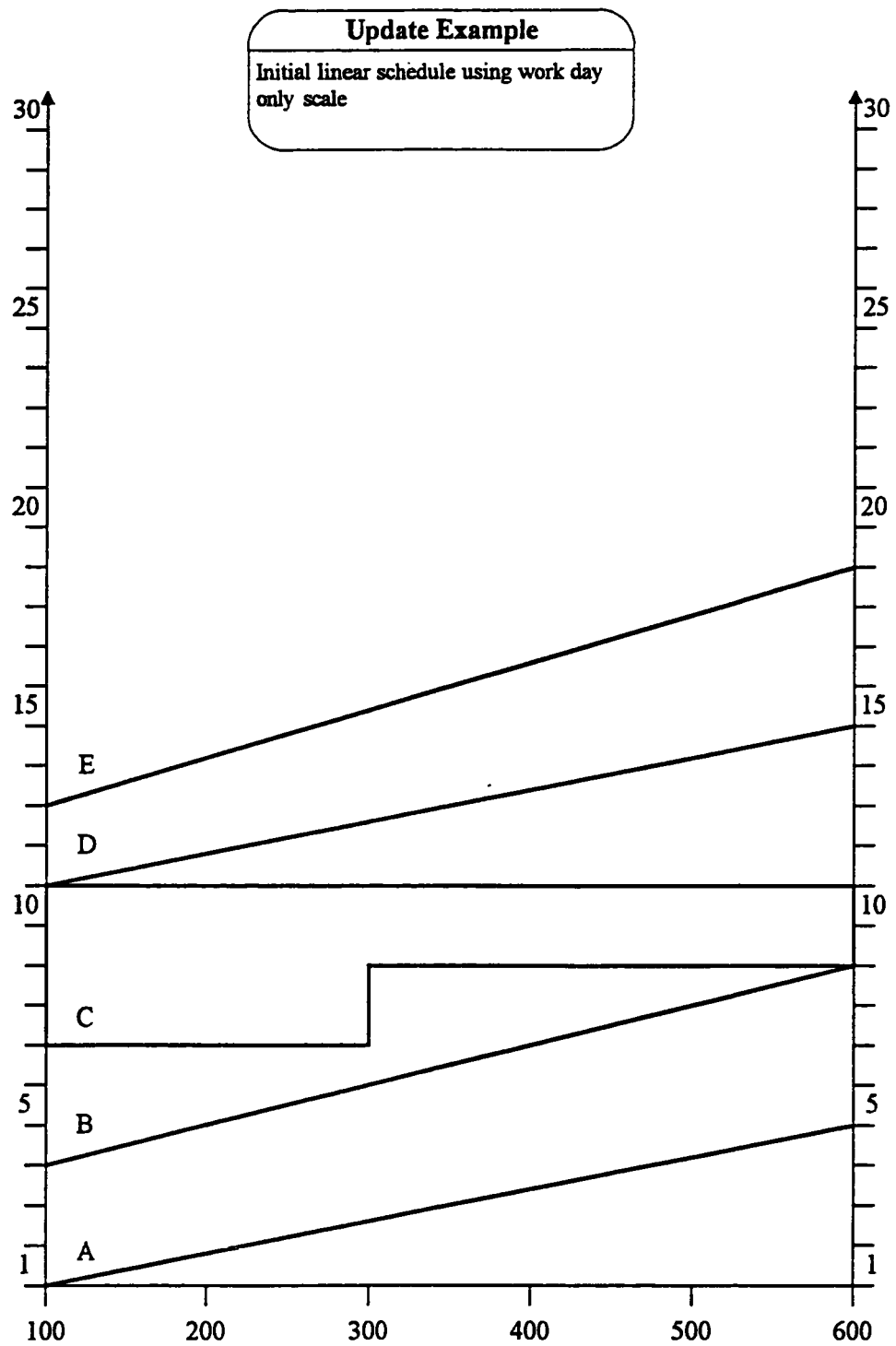
Figure 72 Update Example Linear Schedule
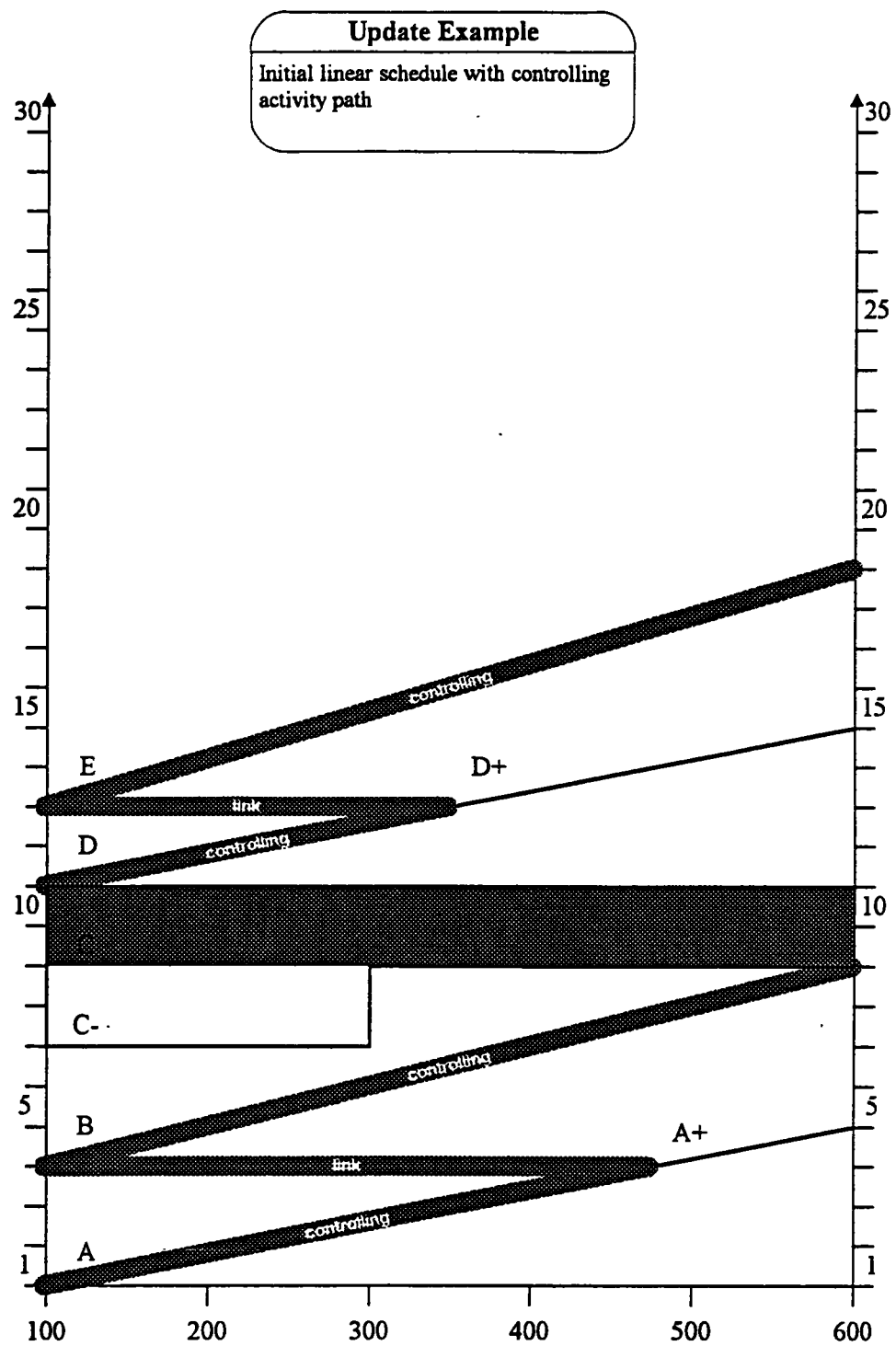
Figure 73 Update Example Downward Pass
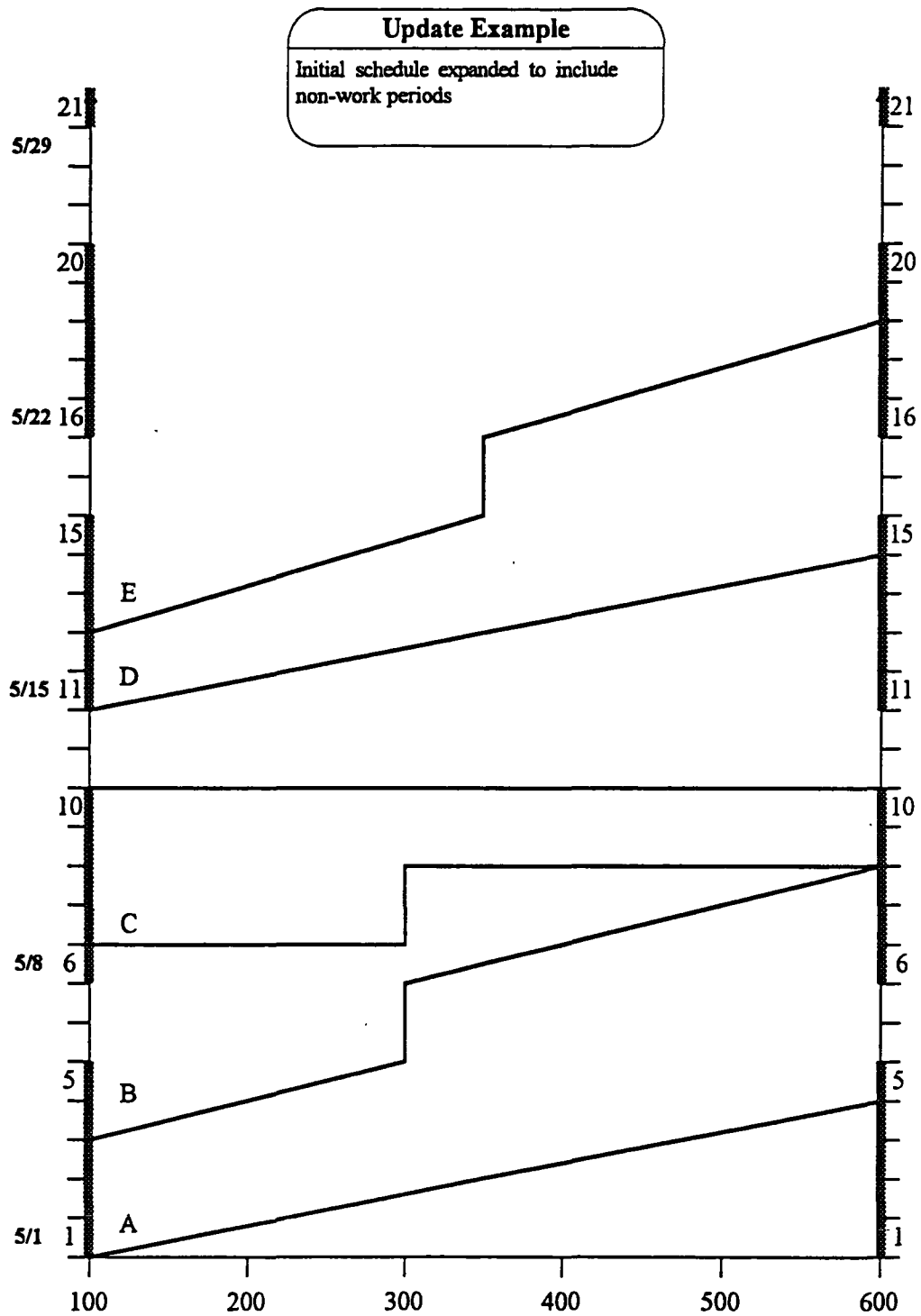
Figure 74 Expanded Initial Schedule

Table 4 Planned Activity Data

Activity	Planned Start Work Day	Planned Start Date	Planned Finish Work Day	Planned Finish Date	Duration Work Days	Start Stations	Finish Stations
A	1	05/01	4	05/04	4	100	600
B	4	05/04	8	05/10	5	100	600
C	7	05/09	10	05/12	4	100	600
D	11	05/15	14	05/18	4	100	600
E	13	05/17	18	05/24	6	100	600

Once the project has started, and work begins to progress on the activities described in the contractor's schedule, the need to update the schedule and provide activity status information becomes apparent. At regular intervals along the course of the project, updates to the schedule should be performed to provide the project's managers with relevant schedule information. Accurate and timely information is a crucial element in a managers ability to make sound project management decisions.

For the purposes of this example, assume that the first two weeks of the project have been completed and the schedule is to be updated after each two week period. The point in time when the update will be performed is called the data date. In this example this date is the end of the 10 planned work days, Friday, May 12, 1995, as Figure 75 indicates. The right-hand y-axis now represents the work days that have been charged to the project below the data date and the planned work days in the future, or above the data date. Remember from the discussion on calendars that, in effect, the charged days can be viewed as planned work days. This means that the plan presented in Figure 74, now needs to be further expanded to include the non-charged planned work periods as shown in Figure 75. Notice that there are three planned work days prior to the data date that have not been charged to the project. This means that the revised plan needs to include these non-charged planned work days. Figure 75 represents the new plan which indicates that the project will now finish on May 30, 1995, 6 days later than the initial schedule. If all the activities were to be completed as this schedule indicates, the project would still finish on-schedule. Table 5 shows a tabular spreadsheet indicating the revised calendar dates for the revised schedule. The work days, however,
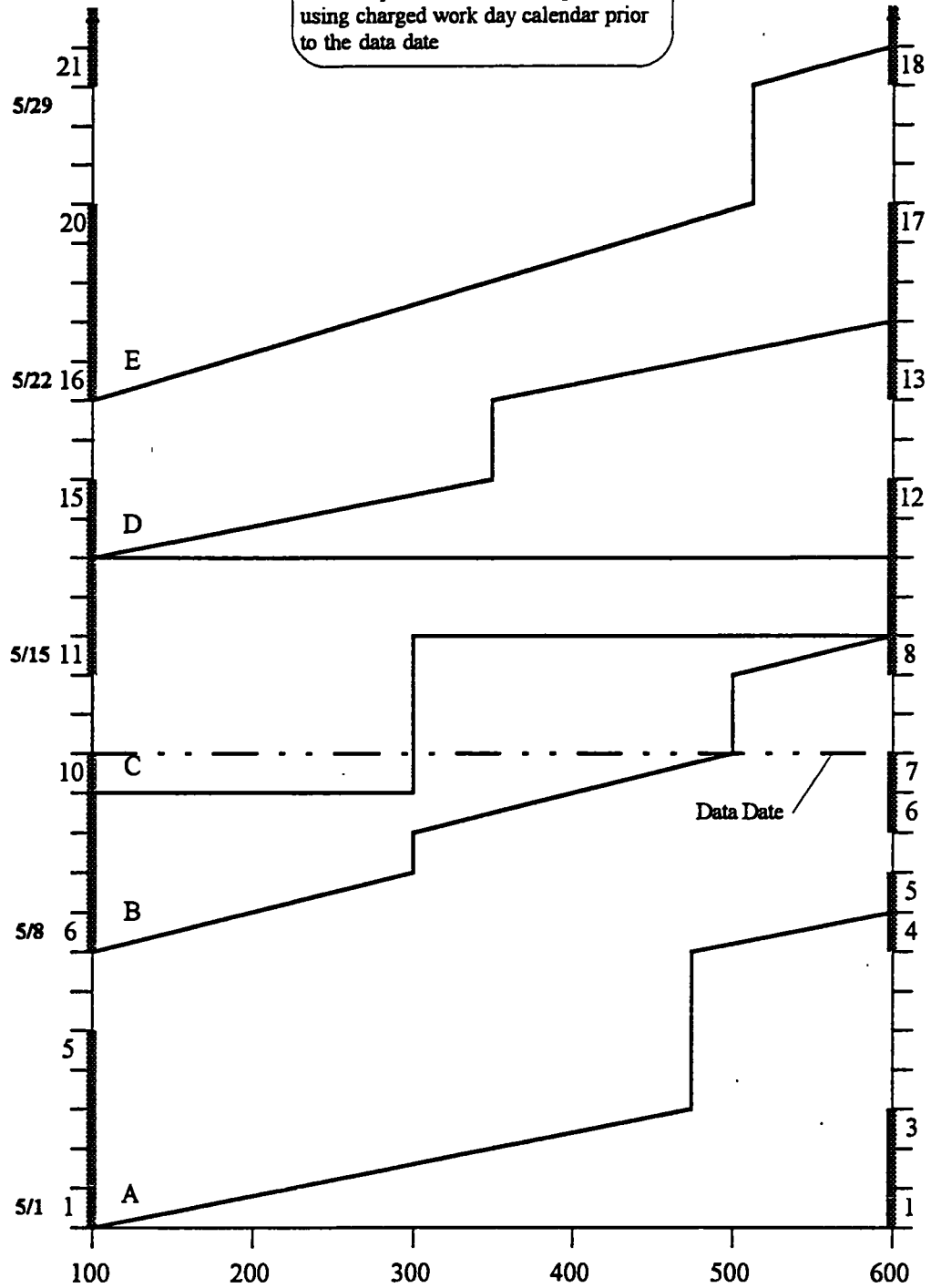
Figure 75 Expanded Revised Schedule

Table 5 Revised Activity Data (end of day 12)

Activity	Planned Start Work Day	Planned Start Date	Planned Finish Work Day	Planned Finish Date	Duration Work Days	*Revised Start Date	*Revised Finish Date
A	1	05/01	4	05/04	4	05/01	05/08
B	4	05/04	8	05/10	5	05/08	05/15
C	7	05/09	10	05/12	4	05/12	05/17
D	11	05/15	14	05/18	4	05/18	05/23
E	13	05/17	18	05/24	6	05/22	05/30

* Charged Day slipage = 3

remain unchanged for the planned start, finish, and duration for each activity.

After the data date has been established, and the schedule has been revised to accommodate the charged work day calendar, the as-built activity progress needs to analyzed to determine if activity progress has any effect on the future schedule. Figure 76 shows the activity progress in bold dashed lines. The only activities for which any progress has been recorded are activities **A** and **B**. Although the plan indicates that Activity **C** should have started no progress has been recorded for this activity. Table 6 shows a tabular spreadsheet of the activity progress information recorded on the revised schedule in Figure 76. The table shows planned and actual work days for the start and finish of each activity. Expected percent complete and actual percent complete is also shown for each activity in progress. Specifically for this example, the table shows that activity **A** finished one work day late and that activity **B** is 20 percent behind its expected completion. In order to predict the future activity completion dates, however, the impact of the activity progress must be evaluated on the schedule itself.

Figure 77 shows the unfinished portion of activity **B**, at the planned rate, move upward to the beginning of the next work day. Since activity **C** shown no progress the start has been moved up to the next planned work day also. This causes the planned work days of all subsequent activities to move upward on the schedule in time one day. This now represents the updated schedule, and indicates the plan for completing the project as of the data date. It is important to understand, that at this point no new work plan information has been added to the schedule. All uncompleted work as of the data date, is still expected to finish at the

Table 6 Activity Progress Data (end of day 7)

Activity	Planned Start Work Day	Planned Finish Work Day	Duration Work Days	Actual Start Work Day	Actual Finish Work Day	Expected Percent Complete	Actual Percent Complete
A	1	4	4	1	5	100	100
B	4	8	5	4		80	60
C	7	10	4				
D	11	14	4				
E	13	18	6				

* Charged Day slipage = 3
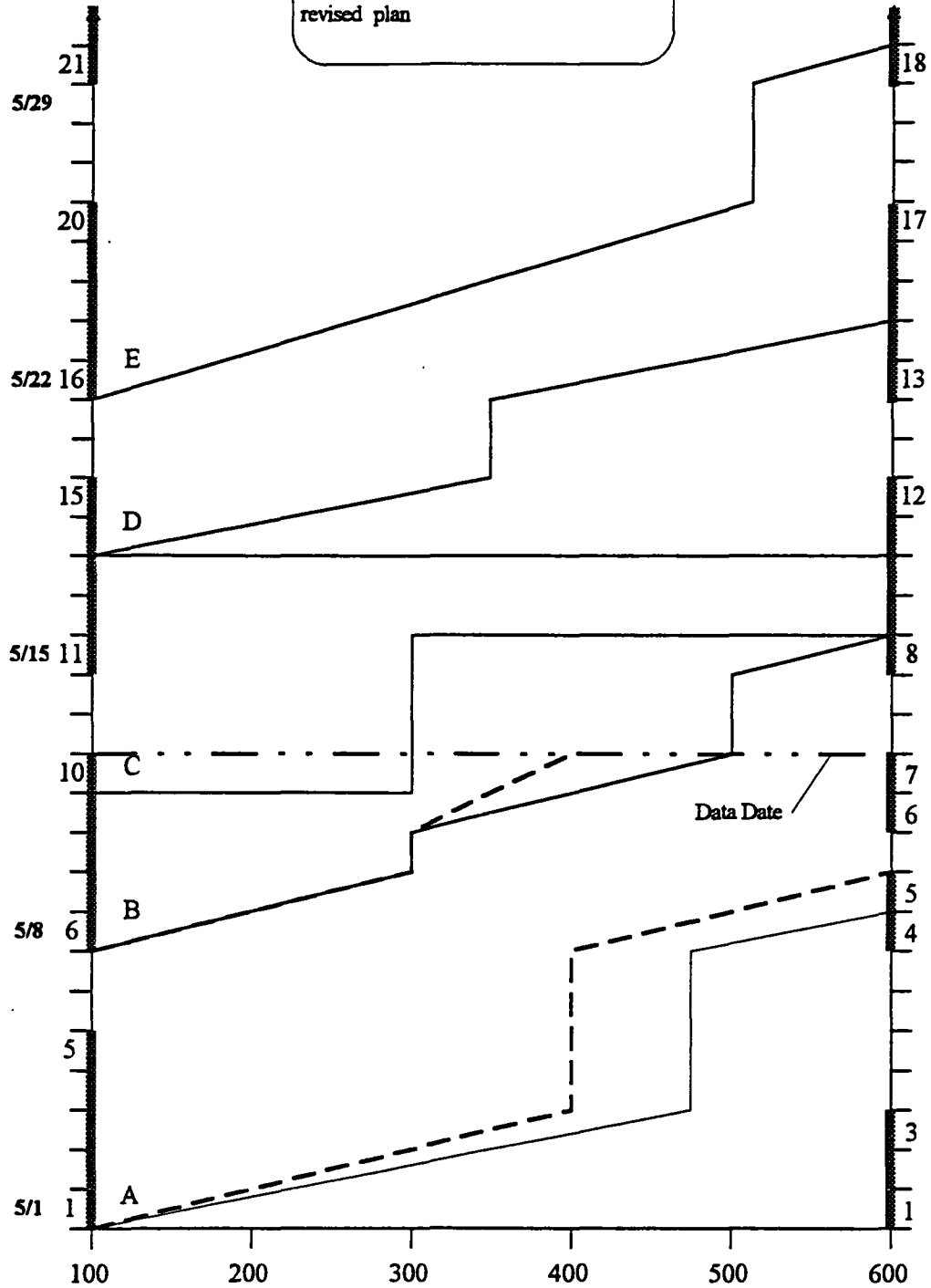
Figure 76 Activity Progress on Revised Schedule

154
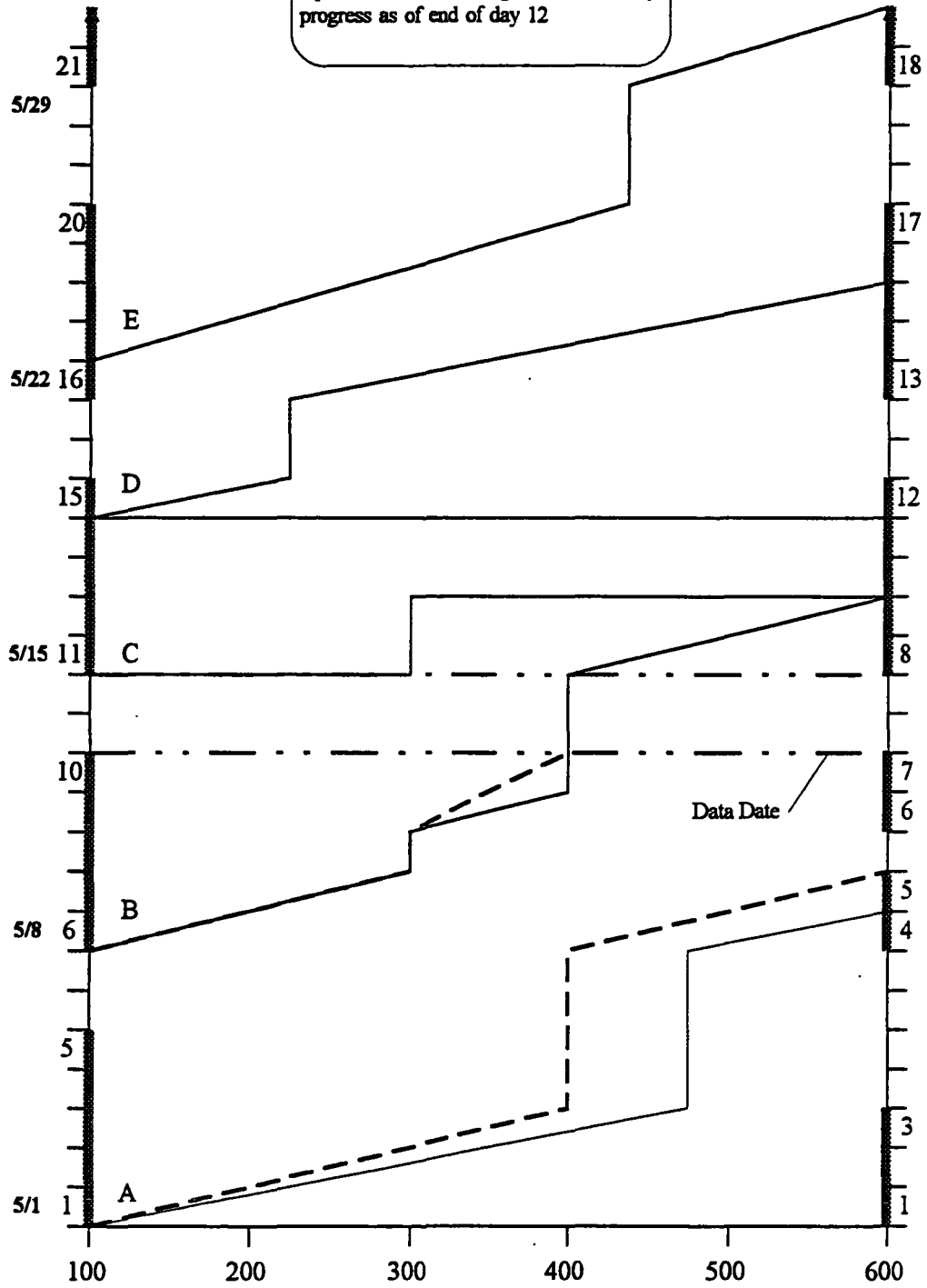
Figure 77 Updated Schedule

planned rates, regardless of previous work progress. It is conceivable that actual progress

information, or even some other rate, could be used to predict future activity rates. However

in this example, this possibility will not be explored. Table 7 shows in a tabular spreadsheet, a

summary of the information on the updated schedule. It shows the predicted start and finish

calendar dates of activities that have not started and of those that have started, but have not

yet finished. The activity status column indicates that all of the activities that have not been

completed, are now expected to complete as scheduled plus one day (+1). Since the last

activity to be completed is activity E, and activity E is expected to be completed one day after

it is scheduled to complete, the project is now considered to be one day behind schedule.

The final step in performing the schedule update is to determine if the controlling

activity path has been affected by the activity progress to date. To calculate the controlling

activity path, the updated schedule must be compressed to remove any non-work periods.

Figure 78 shows the updated schedule after it has been compressed. Any activity progress is

shown below the data date, and the updated planned activities are shown above the data date.

The controlling path is calculated only considering activities above the data date. Figure 79

shows the new controlling activity path as a result of the upward and downward pass on the

appropriate activities. The only change in the controlling activity path, is the inclusion of the

early portion of activity C between stations 100 and 300. At this point, the effect of any

completed activity, or any activity in progress, on the progress of any subsequent completed

activity can also be determined. Remember that activity A finished one day later than

expected. From the information in the schedule, however, this does not appear to be the

Table 7 Updated Activity Data (5/5, beginning of day 15)

Activity	*Revised Start Date	*Revised Finish Date	Updated Start Date	Updated Finish Date	Activity Status
A	05/01	05/08	****	****	
B	05/08	05/15	****	05/16	+1
C	05/12	05/17	05/15	05/18	+1
D	05/18	05/23	05/19	05/24	+1
E	05/22	05/30	05/23	05/31	+1

* Charged Day slipage = 3
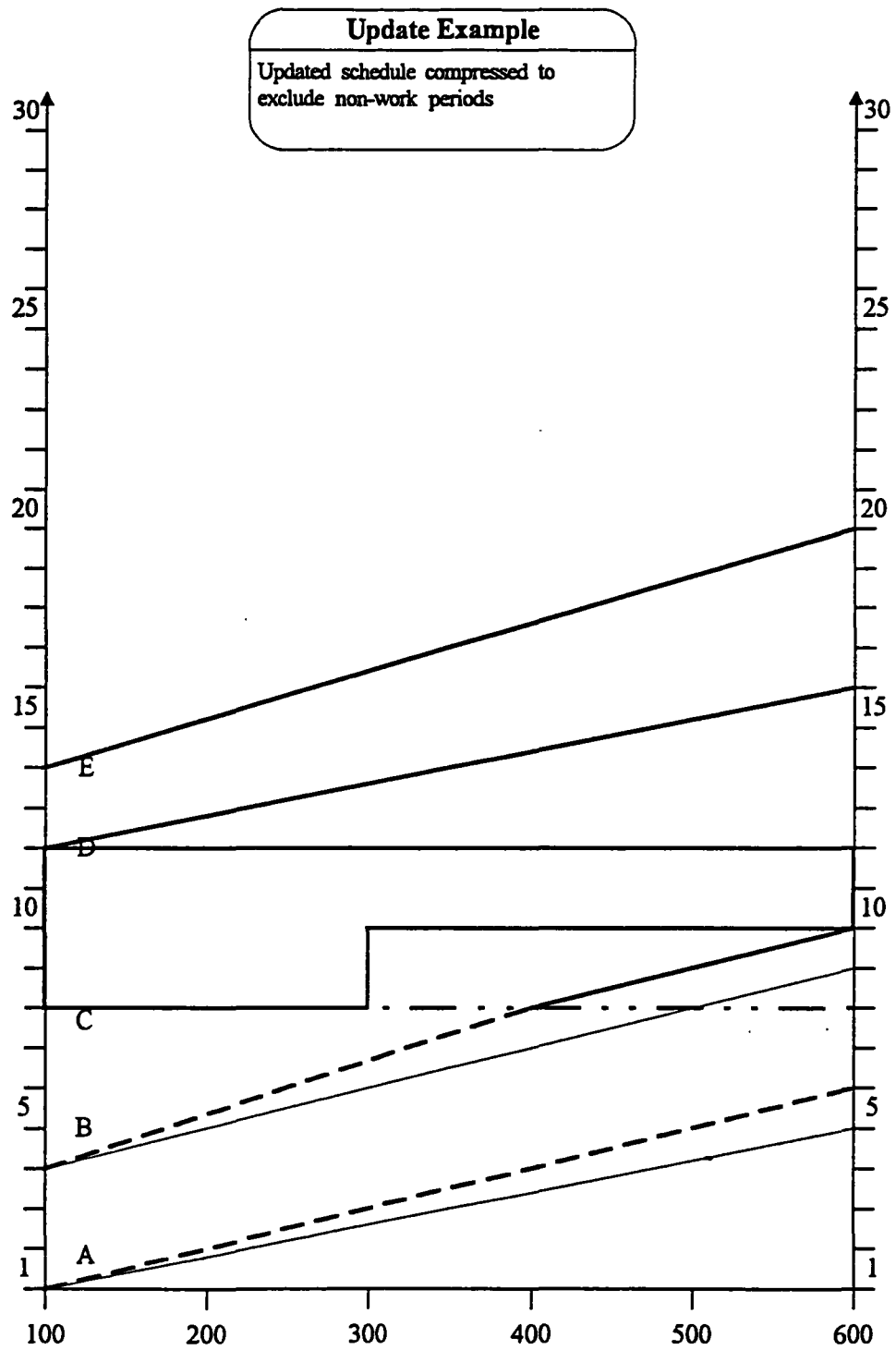
Figure 78 Compressed Updated Schedule
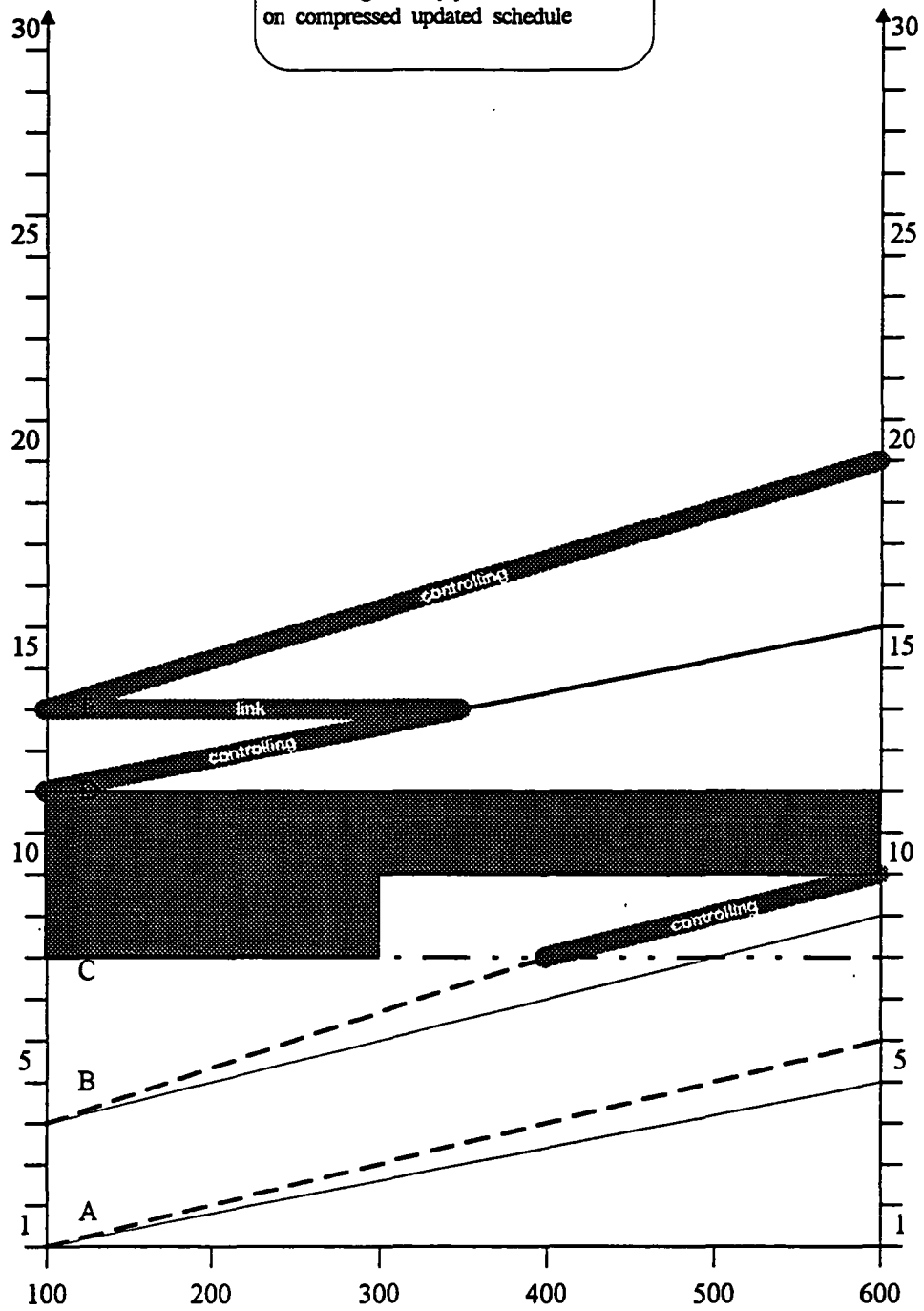
Figure 79 New Controlling Path

reason that activity **B** is behind schedule. It is possible that there is other relevant information that would show this cause and effect, but it is not apparent in the schedule. This completes the schedule update which produces the new current schedule.

In summary, the steps need to perform the schedule update on a linear schedule are:

1. Locate the data date and expand the current schedule to create the revised planned schedule.

2. Record as-built activity information on the revised schedule.

3. Create the updated schedule to reflect the influence of the activity progress information (all uncompleted work has to be above the data date).

4. Compress the updated schedule to determine the controlling activity path and any cause-and-effect conflicts between activities.

LSM COMPUTER APPLICATIONS

A fundamental application has been developed to perform basic linear scheduling functions on a computer. This application has been developed in AutoCAD because a means of displaying the graphical entities that comprise linear schedules was necessary. AutoCAD provides such an environment. While entities are displayed graphically on the screen, they are maintained in the CAD drawing database. Through the use of programming tools supplied with AutoCAD, the user has complete access to the drawing database and all of AutoCAD's functions.

The basic rules, at this stage in the applications development, for the creating a linear schedule using AutoCAD are as follows:

1. The axes are located using x-axis values relative to the actual project stationing, and y-axis values beginning with 1 for the first planned working day on the project.

2. All activities are created on individual layers. The name of the layer is the name of the activity.

3. All activities are created as polylines. Linear activities are drawn starting on the left (lowest x-value) to the right (highest x-value). Blocks are closed polylines drawn counter-clockwise from the lower left (lowest x-value and lowest y-value). Bars are polylines drawn from the bottom to the top.

Input Screens

An input screen has been developed that helps the user create the initial layout of the

Figure 80 Example Layout Input Screen

linear schedule. Figure 80 shows an example of that screen.

The *Project Location* box collects information about start and end points of the

project in stations. There is a button marked *Project Landmarks* which assists the user in

identifying landmarks such as intersections, bridges, or crossovers. These are indicated below

the x-axis and help to orient the schedule to actual project features. The next section, *Project*

Calendar, is concerned with the collecting the following information:

1. The planned start date of the project,

2. The number of working days allowed to complete the project,

3. The number days per week that the constructor plans to work, and

4. The first work day of the week.

The next section allows the user to identify additional days in which no work will be performed other than those identified by the work week definition. The user enters a date into the boxes in the *Edit Non-Work Periods* section and presses the *Add* button to add the date to the list of dates above. To delete a date the user highlights it in the *Non-Work Period* list box and then selects the *Delete* button in the section below. This will remove the highlighted date from the *Non-Work Periods* list. The last section on this screen will calculate the project completion date based on the information supplied in this screen.

All of the information that is entered in the layout screen is saved to the CAD drawing of the schedule. This approach allows for portability of the schedule by only exchanging the CAD drawing. Once the layout screen is finished, the program will draw the initial layout for the linear schedule. The start location and the "Julian" date of the project start date will form the x and y coordinates of the lower left hand corner of the schedule. The "Julian" date is calculated numerically by counting the number of days that have elapsed since January 1, 1900 until the day in question. This means that every date on the linear schedule corresponds to a y-value on the linear schedule. The D2J and J2D functions in Appendix B are used to change calendar dates into "Julian" dates and to convert "Julian" dates into calendar dates respectively. A y-axis is drawn at each end of the x-axis determined by the start and end stations for the project. The x-axis is labeled with the appropriate stations and the y-axis is labeled with ordinate dates starting with the first date of the project and calendar dates. Colored bars are also placed along the left y-axis to indicate days that are planned work days.

Figure 81 Activity Creation Screen

These bars are usually green.

Another input screen, shown in Figure 81, has been developed that helps the user

create activities on the linear schedule. On this screen the user can enter information

necessary in the creation of activities on the linear schedule. A description of the activity can

be entered and the type of activity identified. The start and end location of the activity are

entered next. The last piece of information necessary is the either the duration of the activity

or the production rate and the appropriate units. Once this screen is complete, the appropriate

layer will be created in the CAD drawing and the activity described in the input screen will be drawn on the linear schedule. These are the two primary input screen for the creation of the linear schedule and the activities that are contained on that schedule. The rest of this section explains how the linear scheduling application implements the creation of the controlling activity path defined in the Linear Scheduling Model.

AutoLISP Routines

A simple linear schedule produced in AutoCAD is shown in Figure 82. There are seven activities on this schedule, including continuous full-span linear (CFL), continuous partial-span linear (CPL), partial-span block (PB), and intermittent full-span linear (IFL) activities. The activities proceed in different directions, and one has a variable rate. In the CAD drawing, each of these activities is drawn as a polyline on a separate layer with the layer name that of the activity. Each polyline is comprised of a set of vertices. AutoCAD, through the AutoLISP programming language, provides functions to retrieve all of the polylines on a specific layer and then create a list of the vertices in that polyline. This is an important aspect of AutoCAD and is used repeatedly throughout the linear scheduling application.

The application identifies and displays the controlling activity path in a linear schedule using the following primary routines:

1. MAKE_ASL - Makes the activity sequence list

2. UPASS - Performs the upward pass in a linear schedule

3. DPASS - Performs the downward pass in a linear schedule

4. SHOW - Display the controlling activity path on the linear schedule
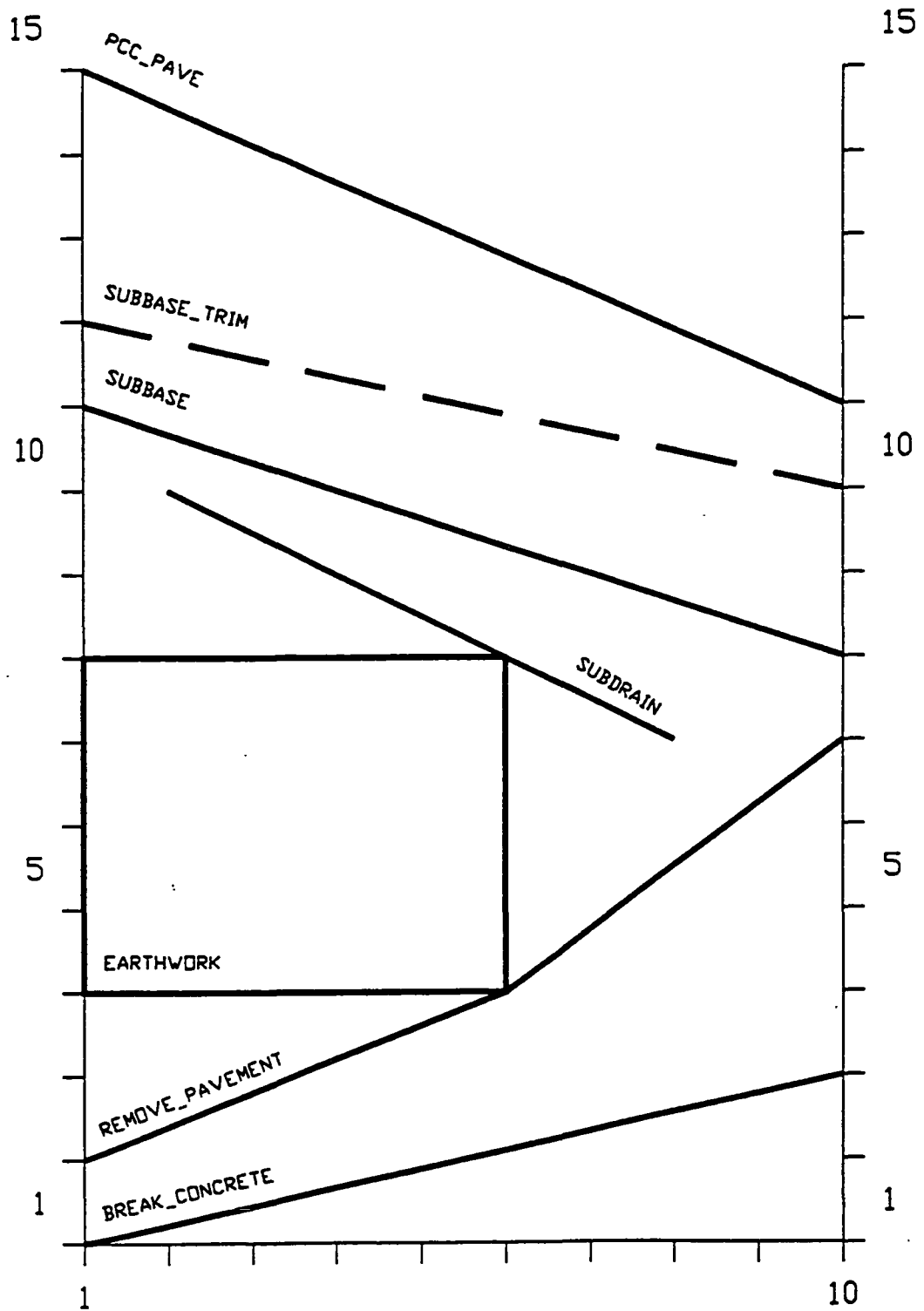
Figure 82 Sample Linear Schedule in AutoCAD

These four primary routines, their associated subroutines, and the output files that they produce are included in Appendix B.

Make_ASL

The *make_asl* routine is a user interactive routine that develops the activity sequence list. The user is prompted to select, in order, the activities in the activity sequence list, and then the routine writes the activity name to the file "drawing_name.asl". This routine will be replaced with a routine that develops the activity sequence list without user interaction. A sample output file, "test.asl", for the example linear schedule in Figure 82 is included in Appendix B.

Upass

The *upass* routine performs the upward pass in a linear schedule. A generalized flowchart of this routine appears in Figure 83 This routine gets activity names from the ".asl" file and uses these names to create a list of the vertices of the polyline(s) on the associated layer with the *vlist* subroutine. The first pair of activities is then analyzed to find the least time interval and the least distance interval using the subroutines; *findlt*, *findld*, and *verts*. The activity name and associated vertice list, along with the endpoints of the least time interval and the least distance interval, is written to the file "drawing_name.lnk". A sample of a .lnk file, "test.lnk", is included in Appendix B. The *upass* routine also uses a subroutine called *etos* which converts expressions to strings for writing to output files.

Dpass

This routine performs a downward pass in a linear schedule. *Dpass* uses two simple
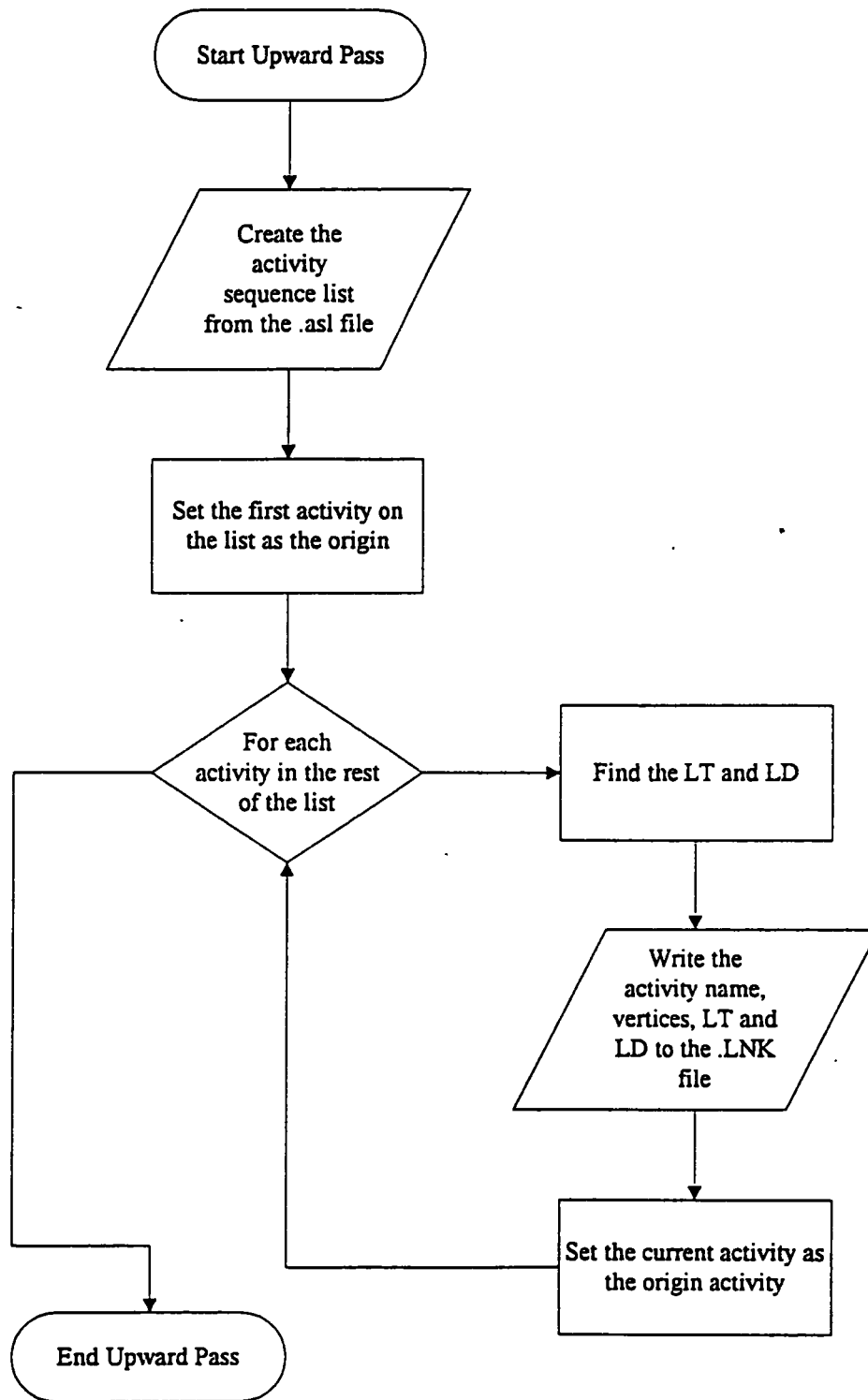
Figure 83 Upward Pass Flow Chart

subroutines; *ctline* and *getline*. *Ctline* simply counts the number of lines in the ".lnk" file, while *getline* uses the number of lines to retrieve them in reverse order since the downward pass starts at the end of the schedule. Figure 84 shows a generalized flowchart of the steps utilized by the *dpass* routine to perform the downward pass. For each activity and associated least distance interval in the ".lnk" file, the routine determines the controlling segment. The controlling segment name, associated vertices, and endpoints of the least distance interval, are written to the "drawing_name.ctl" file. This routine then steps through each pair of activities in the ".lnk" file writing the results to the ".ctl" file. The completed ".ctl" file contains the controlling activity path for the linear schedule.

Show

The show routine utilizes the ".ctl" file created in *dpass* to draw the controlling activity path onto the linear schedule. The controlling path is placed on a separate layer named "CONTROLLING", and is typically displayed with the color red. The bold line in Figure 85 is the result of the *show* routine. The controlling activity segments and the controlling links are displayed.

D2J and J2D

Appendix B also contains two other routines, *D2J* and *J2D*. As the linear scheduling model indicates, the y-coordinate of point on a linear schedule represents a time. The *D2J* and *J2D* routines have been developed to handle the conversion of y-coordinates to dates and dates to y-coordinates. The routine *D2J*, takes a calendar date in the form of (DD MM YYYY) and returns a Julian date. For example, if the routine was given "24 7 1995" it would
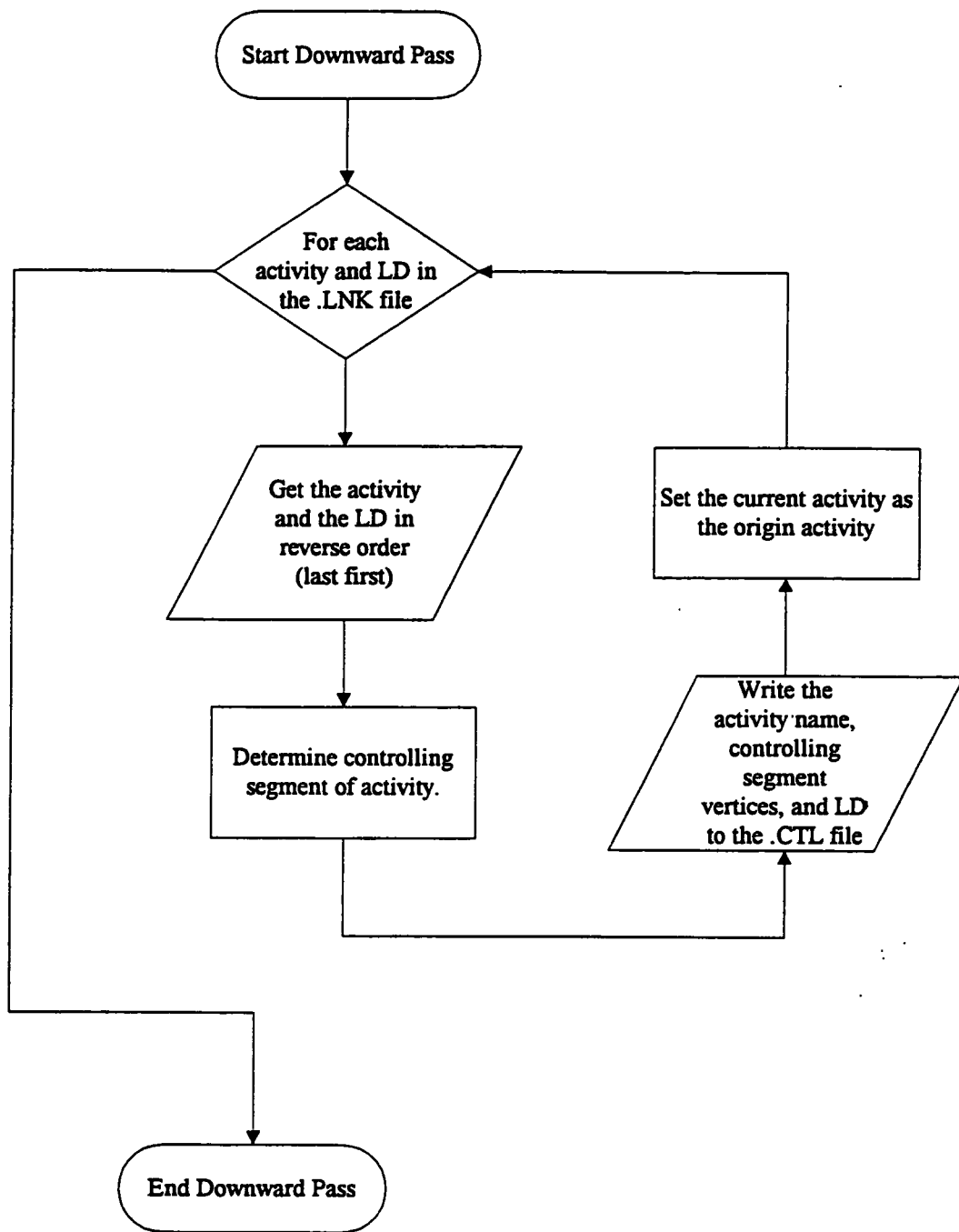
Figure 84 Downward Pass Flow Chart
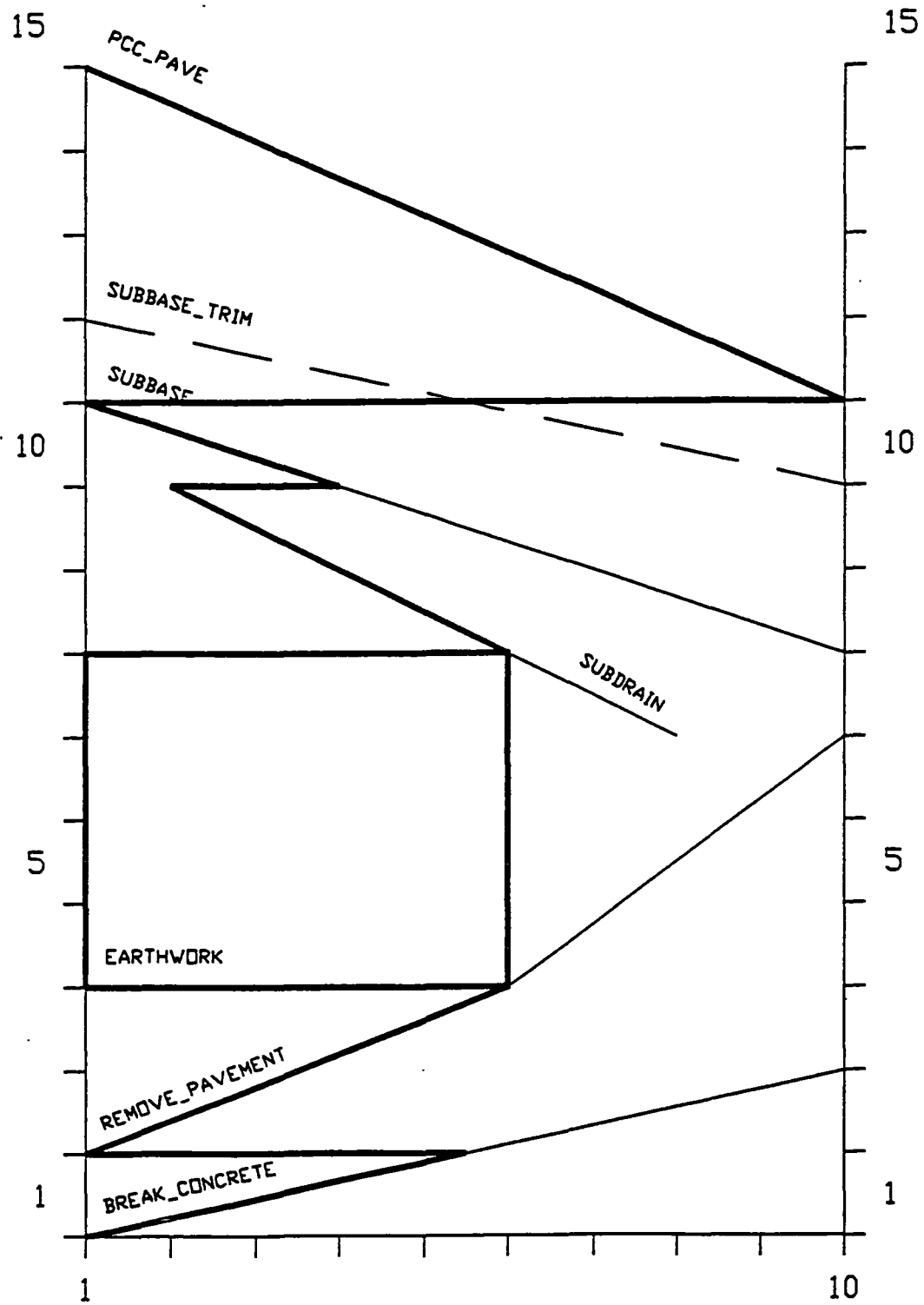
Figure 85 Controlling Path Identified

return the Julian date of "34843". A Julian date is a number representing the number of days from some selected date. In this case the starting date for the Julian calendar is January 1, 1900. Incidently, this is the same Julian calendar that most spreadsheet utilize in their date conversion routines. The *J2D* reverses the previous routine and returns a calendar date if given a Julian date. The routine also has the ability to provide the day of the week. For example, if J2D is provided the Julian date "34843" it returns "Mon 7/24/1995". These routines will be very useful as the linear scheduling application is fully developed.

Output Files

As the previous discussion indicates, three output files are produced by the AutoLISP routines developed to determine the controlling activity path. The files have the same name as that of the AutoCAD drawing containing the linear schedule, but each has a different extension. If the linear schedule was contained in a drawing named "test.dwg", the three associated output files would be "test.asl", "test.lnk", and "test.ctl". Examples of each of these files are contained in Appendix B.

Activity Sequence List File

"Test.asl" is created by the make_asl routine which produces the activity sequence list. The file simply contains the list of activity names, one name on each line, beginning with the first activity in the sequence. The *upass* routine uses the ".asl" to perform the upward pass.

Activity Link File

The file "test.lnk" is produced by the upward pass routine *upass*. This file contains the activity name and associated vertices on one line with the endpoints of the least time and least

distance intervals on the next line. The activity vertices and least distance intervals are used by the *dpass* routine in performing the downward pass.

Controlling Path File

The controlling activity path is contained in the file "test.ctl". This file is produced by the dpass routine, and contains the activity name and the vertices of the controlling segment on one line. The lines between the activities contain the controlling link between the pair of activities. The show routine uses the information in this file to display the controlling activity path on the screen.

LSM v CPM

The definition of a critical path is the sequence of activities along the longest continuous time path through the network. On projects with hundreds or even thousands of activities, identifying the longest continuous time path of activitis through the project can be very important, since progress on these activities is directly linked to the duration of the project. On highway construction projects, however, there seems to be only a small number of activities, usually no more than fifty to twenty. The concept of a critical path, a controlling sequence of activities through the project, becomes less meaningful than it is for projects with a large number of sequential activities. Most of the major activities on highway construction projects are not sequential, they progress in unison along the roadway and are graphically represented on the linear schedule as a group of near parallel lines. As seen on the I-80 Inlay project, there was one activity, pavement removal, in a group of six activities that established the overall rate of production for the entire group. Not until the completion of pavement removal did another activity in the group become critical. The next subsequent activity with a controlling production rate was the paving operation, and the questioon that need to be asked is, "where and when did this activity become critical?" The question was not "what was the sequence of critical activities?" The paving activity becomes critical as soon as the pavement removal activity is complete. Whatever paving was left to be completed, was on the critical path. To minimize the impact that paving had on the project, its completion should have followed that of pavement removal as closely as possible.

The primary differences between the Linear Scheduling Model (LSM) and the Critical

Path Method (CPM) relate to the linear activities present in LSM. Recall that a linear activity models the production rate of an activity as it moves along a path. Any point along this path defines the activity in location and time. Conversely, in CPM this same activity only has a start time and an end time. The location, or direction of travel, is not contained in a CPM activity. When describing several sequential linear activities with CPM, the planner typically attaches a duration to an activity and then creates a start-start relationship with a lag to the next activity. This is repeated for each activity in the group. If the course of the activity is long, say several miles, the only points modeled are the start time and the end time of each activity. Along the course of an activity, if the production rate varies enough to have an effect on another activity, the CPM technique is unable to identify the point of conflict.

One solution to the problem of long duration linear activities in a CPM schedule is to break them up into several pieces. For example, on a project ten miles long a planner could decide to break the project into one mile long segments to model the linear activities. This means that each linear activity is now represented by ten separate activities, and logical relationships must be created between each of these 10 activities and for each 10 activities of all of the other linear activities as well. Obviously, the number of activities and relationships can very rapidly become so numerous that the schedule's value as a project management tool is severely diminished. Not only do the schedules become unmanageable, but they still do not provide adequate information to identify and correct problems that occur somewhere along the mile-long activities. It is impossible to create a CPM schedule that can model linear activities as realistically as the linear scheduling can.

In the Linear Scheduling Model, all linear activities are modeled as line segments. Unlike CPM, in the Linear Scheduling Model activities can have multiple production rates as required. The planner can easily model changes in production rate. For example, if at some point along the course of a concrete removal activity, the contractor knows that another excavator will become available, the production rate of the activity can easily be modified to accommodate the expected increase in production rate without creating another activity. As the project is constructed, daily activity progress can easily be plotted against the planned progress rate to monitor the activities status. The effects of any reduction in production rate of an activity can be easily shown.

Another problem in using CPM to model linear activities is the ability to identify the controlling activities. In CPM an activity is either on the critical path or it is not. It cannot have a portion of the activity on the critical path and another portion not on the critical path. If linear activities are, lets say, ten miles long, then if any part of the activity can be considered critical, the entire activity is defined as critical. In fact, if multiple sequential activities are modeled in CPM using start-start relationships and equal durations, all of the activities in the group will be defined as critical for the entire length of the activity. Common sense contradicts this, making contractors suspicious of the ability of CPM's usefulness as a planning and management tool for highway construction. This report has shown, however, that the controlling activity path described by the Linear Scheduling Model, is specific to only those portions of activities that are actually controlling segments, and that the methodology can be rationally understood in terms of the particular activity production rates and

relationships to other activities.

Linear scheduling adds another important dimension to the planning process that CPM is unable to do. That added dimension is the location of activities on the project. A planner using CPM to model a linear project, or any project for that matter, must mentally visualise the the location of activities with respect to the project and to other activities. Because human's find it difficult to mentally visualise complex spacial relationships, this creates an inherant weakness in the effectiveness of CPM as a planning tool. Linear scheduling uses the spacial relationships of the project to directly provide the planner with a visual planning field in which to arrange activites not only logically but also spacially. The spacial aspect of linear scheduling eliminates any planning errors due to the location in which activities are preformed.

On any construction project it is important not only that a realistic plan for the project is created, but also that the plan can be easily communicated to other parties involved in the construction of the project. The bar charts, logic diagrams, and time-scaled logic diagrams that are typically produced using CPM scheduling procedures are at best very difficult to understand, and can become a frustrating obstacle to effectively communicating the plan for the project. The chart produced by the Linear Scheduling Model provides an excellent tool for communicating the project plan to others involved in the project. The research conducted with the Iowa Department of Transportation demonstrated that even with only a brief exposure to the scheduling technique, parties involved in the project had a clear understanding of the contractors plan to construct the projects. The Linear Scheduling Method provides a method of visualizing where every activity starts, where it ends, and the path with respect to

time and location that it will take. It shows the relationships between activites at any location

for an entire project in a single, easy to understand, diagram.

To summarize, the Linear Scheduling Model has several advantages over CPM for

planning a linear project:

1. The Linear Scheduling Model can realistically determine the controlling activity

   path.

2. The Linear Scheduling Model can accuartely model the production rate

   characteristics of linear activites.

3. As-built production rate information can be easily utilized to track the progress of

   linear activities on the project, providing managers with realistic information

   for making decisions.

4. The Linear Scheduling Model provides a visual method of planning linear projects

   and greatly facilitates the communication of the project plan to other parties

   involved in the project.

SUMMARY

In the past several decades, many attempts have been made to apply linear scheduling techniques to projects of various types. These attempts primarily showed that linear scheduling techniques are best suited to modeling projects in which the major activities are predominantly linear activities. There were two significant drawbacks to the implementation of the technique on actual projects in the construction industry however. The first drawback was that linear scheduling methods were unable to identify a controlling set of activities in a linear schedule. The second drawback was that the technique requires the creation of detailed diagrams to plan and manage the construction process. The rapid advances in computer technology have enabled CPM scheduling techniques to gain widespread use in a very short time, and for linear scheduling to gain widespread acceptance a computer application is necessary. This work is the first known attempt at overcoming the two drawbacks that have impeded the implementation of linear scheduling techniques in the past.

The Linear Scheduling Model presented in this report provides a framework within which linear scheduling can evolve into a technique as robust as CPM has become today. The model provides an analytical method by which controlling activities and a controlling activity path can be identified from within a linear schedule. It also defines an analytical approach for statusing and updating a linear schedule.

A computer application for implementing the Linear Scheduling Model described in this report has been developed for the Iowa Department of Transportation. This application was developed with application development tools provided in AutoCAD. The application

controls the development of linear schedules and performs the functions described in the Linear Scheduling Model.

The development of the Linear Scheduling Model and its implementation from within a computer based application, demonstrate that linear scheduling has the potential of becoming an acceptable method of planning and managing linear construction projects.

CONCLUSIONS

This report demonstrates that linear scheduling can have a significant impact on the planning and scheduling of certain types of construction projects. The Linear Scheduling Model (LSM) and the linear scheduling computer application verify that an analytical approach to linear scheduling exists and that the model can be implemented in a computer environment. The literature review and the research projects presented in this report indicated that the development of an analytical approach and computerization of linear scheduling pose significant barriers to the implementation of linear scheduling in the construction industry today.

The Linear Scheduling Model describes and defines the various activity types found on a linear schedule and provides a graphical means of visualizing the planning and management of a linear project. The Linear Scheduling Model provides an analytical framework for linear scheduling by defining several significant functions:

1. The model defines a method for determining controlling segments of the various activity types found in a linear construction project.

2. The model determines the controlling activity path by linking together relevant segments of the controlling activity segments.

3. The model provides a method to calculate the rate float for beginning and ending non-controlling activity segments.

4. The model defines several types of project calendars and provides a method of reconciling the calendars and the activities on the linear schedule.

5. The model defines a method of tracking progress of activities on a linear schedule and updating the project construction plan based on current activity progress.

6. The model describes how the information developed in updating the linear schedule can be used to create project status reports for managing the construction process on a linear project.

7. The model also provides a means of capturing the planned and as-built activity production information for an entire linear construction project in a single, easy to understand visual database.

The Linear Scheduling Model's definition of controlling items is substantiates what was learned during the three trial projects. For example, the Model would have correctly predicted that the pavement removal activity of the I-80 Inlay project would have been the controlling activity on the project as long as it was in operation, and that once it was complete the following operation would have been on the controlling path for a short while. The CPM schedule for the project indicated that the pavement removal activity and all of the following intermittent activities were on the critical path. Intuitively, personnel on the project knew that the following intermittent activities, such as dirt trim, will never be a controlling activity. The discrepancy between the construction team's practical knowledge and the schedule information produced using CPM techniques, contributes to the lack of confidence felt by the construction team with respect to CPM's usefulness as a management tool. The Linear Scheduling Model produces a much more realistic representation of the project's controlling activities and controlling activity path because it is able to account for the spacial and

production rate attributes of the activities as well as their duration and construction sequencing attributes.

Since the linear activities in a linear schedule are inherently based on production rates of the individual activities, and since production rates are the dominant factor in a linear activity, the Linear Scheduling Model focuses the planner's attention on production rates rather than on activity durations like CPM does. The linear schedule diagram provides the planner with a field in which the spacial and time needs of all of the project activities can be visually coordinated.

The Linear Scheduling Model also forms a basis for the determination of which activities have float and how much the actual production rate can vary from the planned production rate without effecting the controlling activity path. For beginning non-controlling linear activities, LSM identifies those activities, that if they were started earlier, could progress at a lower rate than planned. If the activity is started at the time it was planned to start, any rate float is lost and the beginning non-controlling segment is effectively a controlling activity. Ending non-controlling segments fit more closely into the definition of float traditionally provided by CPM scheduling applications. In this case, the end of the activity can be delayed without affecting the controlling activity path. As the section on rate float indicates, the availability of float for any individual activity segment in a group of either ending or beginning non-controlling activities, is dependant on the other activities in the group. This is very similar to the concepts of total float and free float commonly used in describing float in a CPM schedule.

The model's ability to manage all of the potential calendars on a linear project again provides for a much more realistic connection between the schedule and the actual construction project. Managing of the various calendars for a typical linear project within a CPM scheduling application is difficult. Since CPM techniques do not include a concept of charged days, the application must be artificially manipulated to produce the necessary results.

A significant contribution of the Linear Schedule Model is the production of the as-built schedule. This diagram contains all of the daily production data for every activity on the project in a single visual database. Not only can the production for any data be easily obtained but the effects of any activity's progress on other activities can be visually portrayed. Without the as-built schedule, the effects a linear activity's progress on subsequent activities is nearly impossible to clearly identify. The Linear Scheduling Model has tremendous potential to significantly impact the way in which linear construction projects are planned and managed in the future.

RECOMMENDATIONS

This work is only the beginning of the Linear Scheduling Model. It only develops the primary functions that allow it to become a common technique in the construction planner's assortment of project management tools. The Linear Scheduling Model lays the foundation for the development of additional features and applications. This technique has the potential of providing all of the analytical features that current CPM applications provide today.

There is a significant amount of work to be done in the future to develop the functions and applications. For example, resource and cost loading features present in CPM are needed for LSM. Since the variable production rates of any linear activity can be directly driven by the resources used to perform that activity, a method of leveling the production rate of linear activities based on resources needs to be developed. The cost of resources (materials, labor, and equipment) can also be directly applied to the production rates of the activities on a linear schedule to provide a myriad of project management information. The necessary information for project management has been fairly well defined in the past. The unique attributes of the Linear Scheduling Model need to be exploited to provide better information for the management of linear projects. Obviously, when CPM was first developed, only the primary functions were described, and others were developed and added as the technique matured. The Linear Scheduling Technique developed in this work is in its infancy; a significant amount of work needs to be done before the technique will reach maturity.

185

BIBLIOGRAPHY

References Cited

Arditi, D., and Albulak, M. Z. (1986). "Line of Balance scheduling in pavement construdtion." *Journal of Construction Engineering and Managenment*, ASCE, 112(3), 411-424.

Birrell, G. S. (1979). "Construction planning--beyond the critical path." *Journal of the Construction Division*, ASCE, 106(3), 389-407.

Carr, R. I., and Meyer, W. L. (1974). "Planning construction of repetative building units." *Journal of the Construction Division*, ASCE, 100(3), 403-412.

Chrzanowski, E. N., and Johnston, D. W. (1988). "Application of linear scheduling." *Journal of Construction Engineering and Managenment*, ASCE, 112(4), 476-491.

Johnson, D. W. (1981). "Linear Scheduling Method for Highway Construction." *Journal of the Construction Division*, ASCE, 107(2), 247-261.

Khisty, C. J. (1970). "The application of the line of balance technique to the construction industry." *Indian Concrete Journal*, 44(7), 297-300, 319-320.

Moselhi, O., and El-Rayes, K. (1992). "Scheduling of repetitive projects with cost optimization." *Journal of Construction Engineering and Managenment*, ASCE, 119(4), 681-697.

O'Brein, J. J. (1975). "VPM scheduling for high-rise buildings." *Journal of the Construction Division*, ASCE, 101(4), 895-905.

Peer, S. (1974). "Network analysis and construction planning." *Journal of the Construction Division*, ASCE, 100(3), 203-210.

Pietras, C. M., and Coury, B. G. (1994). "The development of cognitive models of planning for use in the design of project management systems." *International Journal of Human--Computer Studies* 40, 5-30.

Rowings, J. E., Harmelink, D. J., and Rahbar F. (1993). "A multi-project scheduling procedure for transportation projects -- Final Report." Iowa Department of Transportation, Ames, Iowa.

Rowings, J. E., Harmelink, D. J. (1993). "A multi-project scheduling procedure for transportation projects -- Final Report Part II." Iowa Department of Transportation, Ames, Iowa.

Russell, A. D., and Caselton, W. F. (1988). "Extensions to linear scheduling optimization." *Journal of Construction Engineering and Managenment*, ASCE, 114(1), 36-53.

Russell, A. D., and Wong, W. C. M. (1993). "New generation of planning structures." *Journal of Construction Engineering and Managenment*, ASCE, 119(2).

Sarraj, Z. M. (1990). "Formal Development of Line-of-Balance Technique." *Journal of Construction Engineering and Management*, ASCE, 116(4), 689-703.

Selinger, S. (1980). "Construction planning for linear projects." *Journal of the Construction Division*, ASCE, 106(2), 195-205.

Stradal, O., and Cacha, J. (1981). "Time space scheduling method." *Journal of the Construction Division*, ASCE, 108(3), 445-457.

Vorster, M. C., and Bafna, T. (1992) Discussion of "Formal Development of Line-of-Balance Technique." by Sarraj, Z. M. (1990). *Journal of Construction Engineering and Managenment*, ASCE, 118(1), 210-211.

Vorester, M. C., and Beliveau, Y. J., and Bafna, T. (1992). "Linear scheduling and visualization." *Transportation Research Record* 1351, 32-39.

Other References

Barrie, D. S. (1978). *Professional construction management.* McGraw-Hill, New York, New York.

Bennett, J. (1985). *Construction project management.* University Press, Cambridge, Great Britian.

Halpin, D. W. (1976). *Design of construcion process operations.* John Wiley & Sons, Inc., New York, New York.

Handa, V. K., and Barcia, R. M. (1986). "Linear scheduling using optimal control theory." *Journal of Construction Engineering and Managenment*, ASCE, 112(3), 387-393.

Harris, F. C., and Evans, J. B. (1977). "Road construction--simulation game for site manager." *Journal of the Construction Division*, ASCE, 103(3), 405-415.

Harris, F. (1986). *Worked examples in construction management*. Second Ed. William Collins Sons & Co. Ltd, London England.

Herbsman, Z. J. (1985). "Evaluation of scheduling techniques for highway construction projects." *Transportation Research Record* 1126, 110- 120.

Jaafari, A. (1984). "Criticism of CPM for project planning analysis." *Journal of Construction Engineering and Managenment*, ASCE, 110(2), 222-233.

Lochyer, K. G. (1984). *Critical path analysis and other project network techniques*. Fourth Ed., Pitman Publishing Limited, Marshfield, MA.

Loulakis, M. C., and Cregger, W. L. (1993). "Flawed CPM schedule may be costly to contractor." *Civil Engineering*, Nov. 1993, 38.

Melin, J. W., and Whiteaker, B. (1981). "Fencing a bar chart." *Journal of the Construction Division*, ASCE, 107(3), 497-507.

Moder, J. J. (1983). *Project management with CPM, PERT, and prededence diagramming*. Van Nostrand Reinhold Company Inc., New York, NY.

O'Brein, J. J., and Kreitzberg, F. C., and Mikes, W. F. (1985). "Network scheduling variations for repetitive work." *Journal of Construction Engineering and Managenment*, ASCE, 111(2), 105-116.

Perera, S. (1982). "Resource sharing in linear construction." *Journal of Construction Engineering and Managenment*, ASCE, 109(1), 102-111.

Prendergast, J. R., and Gobeli, D. H. (1991). "A survey of project scheduling tools." Engineering Management Journal, ASEM, 3(2), 35-42.

Pultar, M. (1990). "Progress-based construction scheduling." *Journal of Construction Engineering and Managenment*, ASCE, 116(4), 670-687.

Reda, R. M. (1990). "RPM repetitive project modeling." *Journal of Construction Engineering and Managenment*, ASCE, 116(2), 316-330.

Singh, A. (1991). "Knowledge bases for C/SCSC." *Cost Engineering*, 33(6), 39-48.

Tavakoli, A. (1991). "ODOT's construction and control process." *Cost Engineering*, 33(3), 13-19.

Thabet, W. Y., and Beliveau, Y. J. (1992). "Modeling work space to schedule repetitive floors in multistory buildings." *Journal of Construction Engineering and Managenment*, ASCE, 120(1), 96-131.

Thomas, R. T., and Mathews, C. T., and Ward, J. G. (1986). "Learning curve models of construction productivity." *Journal of Construction Engineering and Managenment*, ASCE, 112(2), 245-257.

APPENDIX A

LINEAR SCHEDULE TERMINOLOGY

Activity Sequence List - a list that describes the order in which activities will occur at any location on the project.

Activity Vertices - the beginning, end, or any point identifying a change in production rate on a linear activity; any corner of a block activity; either end point of a par activity.

Adjacent (consecutive) Activities - any two activities that are consecutive to each other on the activity sequence list.

Beginning Non-controlling Segment - a non-controlling segment at the start of an activity that immediately precedes a controlling segment.

Coincident Duration - this is the interval of time during which the origin and target activities are both in progress.

Continuous Activity - a linear activity that is expected to be in continuous operation from the time it begins until the time it completes.

Controlling Activity Path - this is the path of longest duration through the project that is identified after the downward pass has been completed.

Critical Vertex - the critical vertex occurs at the origin activity end of the critical link and marks the latest potential point for which an activity can be critical.

Downward Pass - beginning with the last activity on the sequence list, identify the critical segment of each activity until the first activity is reached.

Full-span Activity - a linear activity that covers the entire course or span of the project or

planning area.

Intermittent Activity - a linear activity that is not expected to be in continuous operation, but rather starts and stops to pace a preceding continous activity.

Least Distance Interval (LD) - the shortest distance between any two adjacent activities that lies within the coincident duration and intersects the least time interval.

Least Time Interval (LT) - the shortest time interval between any two adjacent (consecutive) activities. The interval will always occur at a vertices of at least one of the activities.

Origin - the earliest point in time on an activity.

Origin Activity - the activity on which the potential controlling segment is currently being determined.

Partial-span Activity - an activity that covers only a portion of the project or planning area. A bar activity is always a partial-span activity while linear and block activities may be - partial-span activities.

Potential Controlling Segment - this is the portion of the origin activity between the origin and the critical vertex. This portion of the origin activity is potentially critical until the downward pass is performed to determine if it is actually critical.

Rate Float - A measure of the amount the production rate for any non-controlling linear activity can be reduced without affecting the controlling activity path.

Target Activity - the first full-span continuous activity following the origin activity in the activity sequence.

Upward Pass - beginning with the first activity on the sequence list, the least time and least

distance intervals and potential controlling segments are identified for each activity

until the end of the list is reached.

APPENDIX B

AUTOLISP ROUTINES AND OUTPUT FILES

VLIST

```
;*******************************************************************
;****   This routine takes the ename of a poly line and      ****
;****   returns a list of the vetices.                        ****
;****                                                         ****
;****                                                         ****
;*******************************************************************


;   **** this returns a list of vertices in a polyline

(defun vlist (poly / ed en templist)
   (setq en (entnext (dxf -1 (entget (ssname poly 0)))))
   (setq ed (entget en))
   (setq templist (list (dxf 10 ed)))

   (while (and (setq en (entnext en)) (setq ed (entget en))
       (/= "SEQEND" (dxf 0 ed)))
       (setq templist (append templist (list (dxf 10 ed)))))

   ) ;while
);vlist
```

FINDLT

```
;******************************************************************
;****   This routine takes the vertice list of a the origin    ****
;****   and the target activities and returns the LT.          ****
;****                                                          ****
;****                                                          ****
;******************************************************************

(defun findlt (act1 act2 / pt1 pt2)
(setq lt nil)

;**** this nested foreach finds the LT from act1 to act2

(foreach pt1 act1
    (setq done nil)
    (setq under nil)

    (foreach pt2 act2

        (setq lttemp nil)
        (cond
           ((= done T))
           ((= (car pt2) (car pt1)) (setq lttemp (list pt1 pt2)) ;....
              (setq done T))
           ((< (car pt2) (car pt1)) (setq under pt2))
           ((and (> (car pt2) (car pt1)) (/= under nil))
              (setq lttemp (list pt1 (list (car pt1) (+ (* (- (car ;....
                 pt1) (car under))
                 (/ (- (cadr pt2) (cadr under)) (- (car pt2) (car under))))
                 (cadr under)) 0.0)))
              (setq done T))
        );cond

        (cond
           ((= lttemp nil))
           ((not (boundp 'lt)) (setq lt lttemp))
           ((< (distance (car lttemp) (cadr lttemp)) (distance (car lt);...
              (cadr lt)))
              (setq lt lttemp))
        );cond

    );foreach)

);foreach

;**** this nested foreach finds the LT from act2 to act1

(foreach pt1 act2
    (setq done nil)
    (setq under nil)

    (foreach pt2 act1

        (setq lttemp nil)
        (cond
           ((= done T))
           ((= (car pt2) (car pt1)) (setq lttemp (list pt1 pt2)) ;....
              (setq done T))
```

```
        ((< (car pt2) (car pt1)) (setq under pt2))
        ((and (> (car pt2) (car pt1)) (/= under nil))
           (setq lttemp (list pt1 (list (car pt1) (+ (* (- (car ;....
               pt1) (car under))
               (/ (- (cadr pt2) (cadr under)) (- (car pt2) (car under))))
               (cadr under)) 0.0)))
           (setq done T))
    );cond

    (cond
        ((= lttemp nil))
        ((not (boundp 'lt)) (setq lt lttemp))
        ((< (distance (car lttemp) (cadr lttemp)) (distance (car lt);...
            (cadr lt)))
           (setq lt lttemp))
    );cond

  );foreach)

);foreach

);defun findlt
```

VERTS

```
;********************************************************************
;****   This routine takes the vertice list of a polyline and   ****
;****   returns a list of vertices that are within the limits   ****
;****   of the LT and the coincident duration of the origin     ****
;****   and target                                              ****
;********************************************************************

(defun verts (act / pt1 pt2 pt3 pt4 dmin dmax templist templist2)

    (foreach pt1 act
       (if (and (or (<= (cadr pt1) (cadr (car lt)))
                    (<= (cadr pt1) (cadr (cadr lt))))
                (or (>= (cadr pt1) (cadr (car lt)))
                    (>= (cadr pt1) (cadr (cadr lt)))))
           (setq templist (append templist (list pt1)))

       );if

    );foreach

    (setq dmin (cadr (car tglist))
          dmax (cadr (car orlist)))

    (foreach pt2 tglist
       (if (< (cadr pt2) dmin) (setq dmin (cadr pt2))))); if, foreach
    (foreach pt3 orlist
       (if (> (cadr pt3) dmax) (setq dmax (cadr pt3))))); if, foreach

    (foreach pt4 templist
       (if (and (>= (cadr pt4) dmin)
                (<= (cadr pt4) dmax))
           (setq templist2 (append templist2 (list pt4)))

       );if

    );foreach

    (princ templist2)

);defun verts
```

FINDLD

```lisp
;*******************************************************************
;****   This routine takes each point in the list of possible  ****
;****   LD vertices and finds the point of intersection with   ****
;****   the appropriate activity.  The shortest LD is          ****
;****   returned.                                              ****
;*******************************************************************

(defun findld (overt tvert / ldtemp)
    (setq ld nil
          ldflag nil)
    (foreach pt1 overt
       (setq a (list 0.0 (cadr pt1))
             b (list 1000.0 (cadr pt1)))


       (setq c (car tglist))

       (foreach pt2 tglist
          (setq ldtemp nil)
          (if (inters a b c pt2) (setq ldtemp (list pt1 (inters ;....
             a b c pt2))))
          (cond
             ((= ldtemp nil))
             ((not (boundp 'ld)) (setq ld ldtemp))
             ((< (distance (car ldtemp) (cadr ldtemp)) (distance ;...
                (car ld) (cadr ld)))
                (setq ld ldtemp))
          );cond
          (setq c pt2)

       );foreach
    );foreach

    (foreach pt1 tvert
       (setq a (list 0.0 (cadr pt1))
             b (list 1000.0 (cadr pt1)))


       (setq c (car orlist))

       (foreach pt2 orlist
          (setq ldtemp nil)
          (if (inters a b c pt2) (setq ldtemp (list pt1 (inters;....
             a b c pt2))))
          (cond
             ((= ldtemp nil))
             ((not (boundp 'ld)) (setq ld ldtemp) (setq ldflag T))
             ((< (distance (car ldtemp) (cadr ldtemp)) (distance ;....
                (car ld) (cadr ld)))
                (setq ld ldtemp) (setq ldflag T))
          );cond
          (setq c pt2)

       );foreach
    );foreach

);defun finddt
```

MAKE_ASL

```
;****************************************************************
;****. This routine builds the activity sequence list in an    ****
;****   external file (drawing_name.ASL).  .ALS for activity    ****
;****   sequence list.  This routine prompts the user for the   ****
;****   sequence of activities                                  ****
;****************************************************************

(defun make_asl ()
   (setq ent nil)
   (while (not (and
      (setq ent (entsel "\nSelect first activity:  "))
      (= (dxf 0 (entget (car ent))) "POLYLINE")))
         (princ "You must select an activity.  Please try again")
   );while
   (setq aslfile (open (strcat (getvar "dwgname") ".ASL") "W"))
   (write-line (dxf 8 (entget (car ent))) aslfile)
   (close aslfile)
   (while
      (= (ukword 1 "Yes No" "Enter another activity (Yes or No)" ;....
         "No") "Yes")
         (princ (strcat "The last activity selected was " (dxf 8 ;....
            (entget (car ent)))))
         (while (not (and
            (setq ent (entsel "\nSelect the next activity:  "))
            (= (dxf 0 (entget (car ent))) "POLYLINE")))
               (princ "You must select an activity.  Please try again")

         );while
      (setq aslfile (open (strcat (getvar "dwgname") ".ASL") "A"))
      (write-line (dxf 8 (entget (car ent))) aslfile)
      (close aslfile)
   );while
   (princ ent)

);defun make_asl
```

UPASS

```
;******************************************************************
;****   This routine performs the upward pass using the .asl   ****
;****   file created with make_asl. It creates the .lnk file.  ****
;****   The .lnk file contains activity name and points and    ****
;****   LD and LT for each pair of activities.                 ****
;******************************************************************

(defun upass (/ temp)
   (setq asl nil)
   (setq aslfile (open (strcat (getvar "dwgname") ".ASL") "R"))
   (while
      (setq temp (read-line aslfile))
      (setq asl (append asl (list temp)))
   );while
   (close aslfile)
   (setq aslnum (length asl))

   (setq org (car asl))
   (setq asl (cdr asl))
   (setq activ (car asl));test line
   (setq orlist (vlist (ssget "X" (list (cons 8 org) '(0 . ;....
      "polyline")))))
   (setq tglist (vlist (ssget "X" (list (cons 8 activ) '(0 . ;....
      "polyline")))))
   (findlt orlist tglist)
   (findld (verts orlist) (verts tglist))
   (setq lnkfile (open (strcat (getvar "dwgname") ".LNK") "W"))
   (write-line (strcat "(" "\"" org "\"" (etos orlist)")") lnkfile)
   (write-line (strcat "(" (etos (if (= ldflag T) ld (reverse ld))) ;....
      (etos lt) ")" ) lnkfile)
   (write-line (strcat "(" "\"" activ "\"" (etos tglist)")") lnkfile)
   (close lnkfile)
   (setq org activ)
   (setq asl (cdr asl))
   (foreach activ asl
      (setq orlist (vlist (ssget "X" (list (cons 8 org) '(0 . ;....
         "polyline")))))
      (setq tglist (vlist (ssget "X" (list (cons 8 activ) '(0 . ;....
         "polyline")))))
      (findlt orlist tglist)
      (findld (verts orlist) (verts tglist))
      (setq lnkfile (open (strcat (getvar "dwgname") ".LNK") "A"))
      (write-line (strcat "(" (etos (if (= ldflag T) ld (reverse ;....
         ld))) (etos lt) ")" ) lnkfile)
      (write-line (strcat "(" "\"" activ "\"" (etos tglist)")") lnkfile)
      (close lnkfile)
      (setq org activ)
   );foreach

);defun upass
```

DPASS

```
;****************************************************************
;****   This routine performs the downward pass using the .lnk ****
;****   file created with upass.                            ****
;****                                                        ****
;****                                                        ****
;****************************************************************

(defun dpass (/ templist)
   (ctline)
   (setq temp1 (read (getline aslnum))
         aslnum (1- aslnum)
         name (car temp1)
         pts (cadr temp1)
         temp2 (read (getline aslnum))
         aslnum (1- aslnum)
         ld (car temp2))
   (if (and
      (/= (car (car pts)) (car (last pts)))  ;its a pline
      (< (cadr (car pts)) (cadr (last pts)))) (setq pts (reverse pts))
   );if
   (setq pcount 0)
   (while (and
      (setq pt (nth pcount pts))
      (> (cadr pt) (cadr (car ld))));and
         (setq templist (append templist (list pt)))
         (setq pcount (1+ pcount))
      );while
   (if (= (cadr pt1) (cadr (car ld)))
      (setq templist (append templist (list pt1)))
      (setq templist (append templist (list (car ld)))))
   );if
   (setq ctlfile (open (strcat (getvar "dwgname") ".CTL") "W"))
   (write-line (strcat "(" "\"" name "\"" (etos templist)")") ctlfile)
   (write-line (strcat (etos ld)) ctlfile)
   (close ctlfile)

   (while
      (> aslnum 1)
      (setq temp1 (read (getline aslnum))
         aslnum (1- aslnum)
         name (car temp1)
         pts (cadr temp1)
         temp2 (read (getline aslnum))
         aslnum (1- aslnum)
         ldnext (car temp2)
         templist nil)
      (cond
         ((/= (car (car pts)) (car (last pts)));its a line
            (if (< (cadr (car pts)) (cadr (last pts)))
               (setq pts (reverse pts)));if
            (setq pcount 0)
            (while (and
               (setq pt (nth pcount pts))
               (> (cadr pt) (cadr (car ldnext))));and
                  (if (<= (cadr pt) (cadr (car ld)))
                     (setq templist (append templist (list pt))));if
               (setq pcount (1+ pcount))
            );while
```

```
            (if (= (cadr pt) (cadr (car ldnext)))
                (setq templist (append templist (list pt)))
                (setq templist (append templist (list (car ldnext))))
            );if
            (setq ctlfile (open (strcat (getvar "dwgname") ".CTL") "A"))
            (write-line (strcat "(" "\"" name "\"" (etos templist)")") ;..
                ctlfile)
            (write-line (strcat (etos ldnext)) ctlfile)
            (close ctlfile))
        ((= (cadr (car pts)) (cadr (cadr pts))) ;its a block
            (foreach pnt pts
                (if (< (cadr pnt) (cadr (car ldnext)))
                    (setq pnt (list (car pnt) (cadr (car ldnext))))
                    (if (> (cadr pnt) (cadr (car ld)))
                        (setq pnt (list (car pnt) (cadr (car ld)))))) ;if, if
                (setq templist (append templist (list pnt)))
            );foreach
            (setq templist (append templist (list (car templist))))
            (setq ctlfile (open (strcat (getvar "dwgname") ".CTL") "A"))
            (write-line (strcat "(" "\"" name "\"" (etos templist)")") ;..
                ctlfile)
            (write-line (strcat (etos ldnext)) ctlfile)
            (close ctlfile))

    );cond
    (setq ld ldnext)
);while

(setq templ (read (getline aslnum))
      aslnum (1- aslnum)
      name (car templ)
      pts (cadr templ)
      ldnext '((1.0 1.0)(1.0 1.0))
      templist nil)
(if (< (cadr (car pts)) (cadr (last pts)))
    (setq pts (reverse pts)));if
    (setq pcount 0)
    (while (and
        (setq pt (nth pcount pts))
        (> (cadr pt) (cadr (car ldnext)))) ;and
            (if (<= (cadr pt) (cadr (car ld)))
                (setq templist (append templist (list pt)))) ;if
        (setq pcount (1+ pcount))
    );while
    (if (= templist nil)
        (setq templist (append templist (list (cadr ld))))) ;if
    (if (= (cadr pt) (cadr (car ldnext)))
        (setq templist (append templist (list pt)))
        (setq templist (append templist (list (car ldnext))))
    );if
    (setq ctlfile (open (strcat (getvar "dwgname") ".CTL") "A"))
    (write-line (strcat "(" "\"" name "\"" (etos templist)")") ctlfile)
    (close ctlfile)

);defun dpass
```

SHOW

```
;*******************************************************************
;****   This routine draws the controlling activity path on    ****
;****   the linear schedule in a layer named "CONTROLLING".    ****
;****                                                          ****
;****                                                          ****
;*******************************************************************

(defun show ()
;    (setq curlayer (getvar "CLAYER"))
;    (setvar "CLAYER" "CONTROLLING")

    (setq ctlfile (open (strcat (getvar "dwgname") ".CTL") "R"))
    (while
        (setq templ (read-line ctlfile))
        (setq temp (read templ))
        (if (= (type (car temp)) 'STR)
            (setq temp (cadr temp)));if
        (entmake '((0 . "POLYLINE") (40 . 0.05) (41 . 0.05) (8 .;....
            "CONTROLLING")))
        (foreach pt temp
            (entmake (list '(0 . "VERTEX") (setq v (list 10 ;....
                (car pt) (cadr pt)))))
        );foreach
        (entmake '((0 . "seqend")))
    );while
    (close ctlfile)

;    (setvar "CLAYER" curlayer)
);defun show
```

CTLINE

```
;*******************************************************************
;****   This routine counts the number of lines in the file   ****
;****   "dwgname".LNK. This is needed for the downward pass    ****
;****   which needs the last line first. Sets ASLNUM to the    ****
;****   number of lines in the file.                           ****
;*******************************************************************

(defun ctline ()
   (setq ctfile (open (strcat (getvar "dwgname") ".LNK") "R"))
   (setq aslnum 0)
   (while
      (read-line ctfile)
      (setq aslnum (1+ aslnum))
   );while
   (close ctfile)
);defun getline
```

GETLINE

```
;*****************************************************************
;****    This routine takes a number and returns the line      ****
;****    corresponding to that number from the "dwgname".LNK    ****
;****    file.                                                  ****
;****                                                           ****
;*****************************************************************

(defun getline (num / lineinfo count)
   (setq getfile (open (strcat (getvar "dwgname") ".LNK") "R"))
   (setq count 0)
   (while
      (< count num)
      (setq lineinfo (read-line getfile))
      (setq count (1+ count))
   );while
   (close getfile)
   (princ lineinfo)
);defun getline
```

ETOS

```
;*******************************************************************
;**** ETOS (Expression TO String) takes any expression and    ****
;**** converts to a string.  "STRings" are returned as double ****
;**** ""STRings"". The READ function can be used to return    ****
'**** the original value of any string returned by ETOS.      ****
;*******************************************************************

(defun etos (arg / file)
  (if (= 'STR (type arg)) (setq arg (strcat "\"" arg "\"")))
  (setq file (open "$" "w"))
  (princ arg file)
  (close file)
  (setq file (open "$" "r"))
  (setq arg (read-line file))
  (close file)
  (close (open "$" "w"))
  arg
);defun ETOS
```

TEST.ASL

```
BREAK_CONCRETE
REMOVE_PAVEMENT
EARTHWORK
SUBDRAIN
SUBBASE
PCC_PAVE
```

TEST.LNK

```
("BREAK_CONCRETE"((1.0 1.0 0.0) (10.0 3.0 0.0)))
(((1.0 2.0 0.0) (5.5 2.0))((1.0 1.0 0.0) (1.0 2.0 0.0)))
("REMOVE_PAVEMENT"((1.0 2.0 0.0) (6.0 4.0 0.0) (10.0 7.0 0.0)))
(((6.0 4.0) (6.0 4.0 0.0))((6.0 4.0 0.0) (6.0 4.0 0.0)))
("EARTHWORK"((1.0 4.0 0.0) (6.0 4.0 0.0) (6.0 8.0 0.0) (1.0 8.0 0.0)))
(((6.0 8.0) (6.0 8.0 0.0))((6.0 8.0 0.0) (6.0 8.0 0.0)))
("SUBDRAIN"((2.0 10.0 0.0) (8.0 7.0 0.0)))
(((4.0 10.0) (2.0 10.0 0.0))((2.0 10.0 0.0) (2.0 10.6667 0.0)))
("SUBBASE"((1.0 11.0 0.0) (10.0 8.0 0.0)))
(((10.0 11.0) (1.0 11.0 0.0))((10.0 8.0 0.0) (10.0 11.0 0.0)))
("PCC_PAVE"((1.0 15.0 0.0) (10.0 11.0 0.0)))
```

TEST.CTL

```
("PCC_PAVE"((1.0 15.0 0.0) (10.0 11.0)))
((10.0 11.0) (1.0 11.0 0.0))
("SUBBASE"((1.0 11.0 0.0) (4.0 10.0)))
((4.0 10.0) (2.0 10.0 0.0))
("SUBDRAIN"((2.0 10.0 0.0) (6.0 8.0)))
((6.0 8.0) (6.0 8.0 0.0))
("EARTHWORK"((1.0 4.0 0.0) (6.0 4.0 0.0) (6.0 8.0 0.0) (1.0 8.0 0.0) ...
        (1.0 4.0 0.0)))
((6.0 4.0) (6.0 4.0 0.0))
("REMOVE_PAVEMENT"((6.0 4.0 0.0) (1.0 2.0 0.0)))
((1.0 2.0 0.0) (5.5 2.0))
("BREAK_CONCRETE"((5.5 2.0) (1.0 1.0 0.0)))
```