World Transactions on Engineering and Technology Education
Vol.7, No.2, 2009

Identification of implicit technical relationships in overlapping networks

Chunlu Liu

Deakin University
Geelong, Australia

ABSTRACT: Overlapping network techniques to expand the single sequential finish-to-start relationship between preceding and succeeding activities in scheduling construction projects have been developed for decades. The hidden logic relationships between activities and two virtual activities, *Start* and *Finish*, however limit the applications of these techniques in practice as they may lead to incorrect time parameters of some activities. In this research, a novel approach has been developed to identify those concealed relationships in a structured procedure by a two-dimensional nested diagram. An empirical study was carried out to demonstrate the network modification approach.

INTRODUCTION

The primary objectives of project management are to create a quality project, completed on time, within budget and in a safe work environment. The best schedule is not the schedule showing the project completed in the shortest time, but the schedule that meets all primary objectives of the total project. Construction scheduling is an important function in project management and its effective executive is a condition precedent for successful project outcomes of achieving these objectives [1][2]. The success of a project depends heavily on how effective scheduling is and how tightly the project can be controlled [3]. Poor scheduling can easily result in completion delays and cost overruns. These, in turn, result in claims, counter claims, disagreements and disputes [4].

Although construction scheduling is a conventional teaching theme in the construction and other industries, techniques are still under development [5][6]. For instance, the scheduling issue was ranked as number one among the top research topical areas based on an analysis of the American Society of Civil Engineer's Journal of Construction Engineering and Management between 1985 and 2002 [7].

Network techniques have widely been considered the dominant methods in calculating and presenting construction schedules in both the academic setting and practice. In both activity-on-arrow networks and activity-on-node networks, it is assumed that work can only be scheduled sequentially. This means that an activity can only be scheduled according to a start time based on the completion of its preceding activities. In fact, the finish times of some activities are related to other activities. In addition, the start or finish time of an activity may depend not only on the completion of its preceding activities, but also on their start times. The overlapping network was developed to schedule a project in which two overlapped activities can be modelled easily [8]. Generally speaking, there is no overlapping network based on the activity-on-arrow model. Overlapping networks are just an extension of activity-on-node networks. The terms *activity-on-node networks* or *precedence networks* may also be used to cover overlapping networks [1][9]. The differences between the overlapping network and the activity-on-node network are within relationships. There are no differences between nodes, which represent construction activities.

Overlapping network techniques to expand the single sequential finish-to-start relationship between preceding and succeeding activities in a project schedule have been developed for decades [8]. The hidden logic relationships between activities, however, limit the application of these techniques in practice. In this research, a novel approach was developed to identify those concealed logic relationships in a structured procedure based on a summary of the common logic and rules of constructing overlapping networks. This article originally develops the visual steps by a novel two-dimensional nested table to demonstrate the structured procedure. An empirical example in the construction industry is used to demonstrate the developed procedure and compare the diagram with the one generated under the conventional technique. The final section concludes the major findings of this research.

OVERLAPPING NETWORK

Network Logic

In an overlapping network, activities are represented by nodes including label names (e.g. A, B, C) and durations. Activities can be numbered, but this is not necessary. A node may be represented by a square, rectangle, hexagon or circle. In 1981, the Australian Standard Association released an Australian Standard document on the glossary of items for network planning in the building and construction industry. This standard was withdrawn in 1998. To the best knowledge of the author, it was not replaced by a new standard and this denotes that there is no national standard on how to present a network diagram for education or practice in the building industry. An arrow is used to link two activities and represents their precedence relationships. An arrow with additional information may represent one of four relationships including finish-to-start (FTS), start-to-start (STS), finish-to-finish (FTF) and start-to-finish (STF) relationships. These relationships always relate two activities from the preceding activity to the succeeding activity. Two or more relationships may exist between two activities and, therefore, two or more types of arrow are required for such pairs of activities and, in principle, any two or more of the above four relationships can be combined. As the start and finish times of an activity are interdependent, only a combination of start-to-start and finish-to-finish relationships may be meaningful. The start and finish of a preceding activity may also represent a certain percentage of its duration. However, this kind of relationship between broken activities is not contained in the empirical example in this article. An arrow does not consume time or resources.

The term *lead time* is used to indicate the time required in each relationship between the preceding activity and the succeeding activity. A *preceding* activity and its *succeeding* activity are paired according to their relationship. It does not mean the succeeding activity is later than the preceding activity in time. The lead time may be positive, zero or negative. However, the negative relationship is not suggested and it can always be changed to a positive relationship by changing the order of two activities. If the lead time is equal to zero in a finish-to-start relationship between two activities, the number of zero is normally neglected on the network diagram. If all activities' relationships in an overlapping network are finish-to-start and all lead times are zero, this overlapping network is the conventional precedence network. In other words, the normal precedence network is a special kind of overlapping network, in which only finish-to-start relationships exist between activities and their lead times are all zero.

Network Construction Rules

There should be only one *Start* node and one *Finish* node to identify the start and finish of the project. Each activity should be represented by one and only one node. All nodes except the *Finish* activity must have at least one succeeding node. All nodes except for the *Start* activity must have at least one preceding node. Every arrow must have a head node and a tail node. An arrow specifies only the precedence relationship. Its length and thickness have no significance with respect to the time duration of either of the activities it connects or the lead time of a relationship. Two or more arrows may exist from one activity to another. Although a straight line with one-direction arrow is the normal way to represent the relationship of two activities, an arrow can also be represented in a curved or broken line.

Dummy activities are not required to maintain the logic relationships among activities. A dummy activity is only used to show the start or finish point if there are two or more activities to start or finish the project. The potential two dummy activities are called *Start* and *Finish* and have zero time of durations. It is normally suggested that a dummy *Start* or *Finish* is added even though there is only one activity to start or finish the project. An arrow connected to the dummy activity, *Start* or *Finish*, is normally drawn in a dotted line. It reflects the finish-to-start relationship between a real activity and the dummy *Finish* or between the dummy *Start* and a real activity. The lead time from the dummy *Start* activity to its succeeding activities or from an activity to its dummy succeeding finish activity is zero.

Cycles or close-loop paths through the network are not permitted. A line without an arrow should not appear in the overlapping network. The intersections of arrows should be avoided if possible so that the network can have a concise layout. A path represents a route from the first node (the dummy *Start* activity) to the last (the dummy *Finish* activity) of a network as well as all passed arrows and nodes.

If the early and late start times of an activity are the same, it is evident that the activity has no flexibility or float. Such activities are said to be critical. The critical path is the path from the beginning of the network to the end of the network on which all activities are critical [1]. As two or more arrows may exist between two activities in an overlapping network, the relationships should be indicated for some pairs of activities when presenting a path.

AN OVERLAPPING NETWORK EXAMPLE

An example, described in Table 1, contains the information of the names of eleven activities exclusive of *Start* and *Finish*, their time durations, relationships and lead times. These relationships defined between two real activities are called *logical relationships* in this article.

Table 1: An empirical example.

Activity	Duration	Relationship with preceding activity	Lead time
A	5	begins the project	NA
B	4	start to start with activity A	2
C	5	finish to start with activity A	0
D	3	finish to start with activity A	1
E	3	start to start with activity C	2
		finish to finish with activity C	2
F	6	finish to start with activity C	0
		finish to start with activity D	0
G	6	start to finish with activity B	3
H	5	finish to finish with activity E	2
		finish to start with activity F	0
J	4	start to start with activity F	3
K	2	finish to start with activity G	0
		start to start with activity H	0
L	1	finish to start with activity J	-1

Following the above general logic of activities and rules of an overlapping network, an overlapping network diagram has been constructed and is shown in Figure 1. Normally, *Start* and *Finish* activities are added to the diagram for presentation even though they may be unnecessary. No matter how complex relations among these 11 activities are, the defined relationships are easily represented by simply relating the activities together. In other words, the construction of a logically and technically correct overlapping network requires little knowledge of the network planner as long as the above-mentioned logic and rules are followed. It should, however, be noted that the final shape of a schedule may vary depending on the planner's personal preference and Figure 1 is just one possible configuration of a schedule. The dummy activity, *Start*, defines the start point of activity *A* as well as the whole project. Similarly, the dummy activity, *Finish*, defines the end point of the project, which has two end activities *K* and *L*.

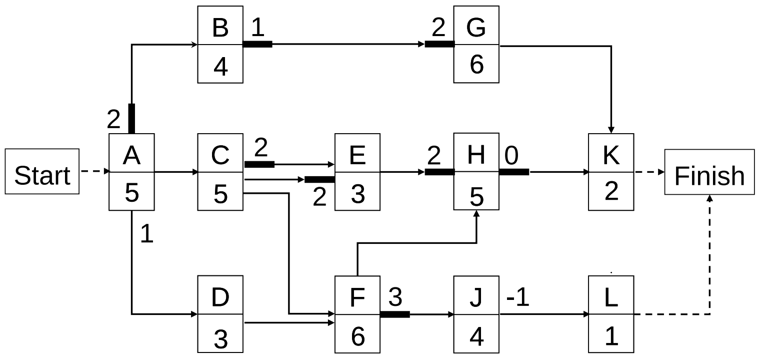

Figure 1: An overlapping network example.

The common computational process applied to an overlapping network is identical to that of an arrow or node network, consisting of forward and backward passes through the network, which was well explained in textbooks, for example [1][2][5][6][9-11]. First, a forward pass through the schedule from *Start* to *Finish* calculates the early start (ES) and early finish (EF) times of activities. A backward pass through from *Finish* to *Start* then calculates the late start (LS) and late finish (LF) times of activities. The calculation of total float (TF) and free float (FF) can start from any activity. Total float is the maximum amount of time that an activity can be delayed without delaying the project completion and free float is the maximum amount of time that an activity can be delayed without delaying the early time of all its succeeding activities. These six time parameters of all 11 activities are summarised in Table 2. It should be noted that the values of these time parameters may be different if a computational process, based on the concept of link lag, is applied. The critical path is from *Start*, through activities *A*, *C*, *F*, *H*, and *K*, to *Finish*.

Apparently, the time parameters of some activities presented in Table 2 are incorrect. For instance, the start time of activity *G* should be earlier than the start point of the project and the end time of activity *H* should not be later than the finish point of the project. As there are four types of relationships in an overlapping network, an activity at the middle of a network may be the activity to start or finish the project. In other words, an activity hiding inside a network may be the earliest or latest activity from the viewpoint of time.

Table 2: Time parameters of construction activities in the original network.

Activity	Duration	ES	EF	LS	LF	TF	FF
Start	0	0	0	0	0	0	0
A	5	0	5	0	5	0	0
B	4	2	6	13	17	11	0
C	5	5	10	5	10	0	0
D	3	6	9	7	10	1	1
E	3	9	12	16	19	7	7
F	6	10	16	10	16	0	0
G	6	-1	5	10	16	11	11
H	5	16	21	16	21	0	0
J	4	13	17	14	18	1	0
K	2	16	18	16	18	0	0
L	1	16	17	17	18	1	1
Finish	0	18	18	18	18	0	0

OVERLAPPING NETWORK MODIFICATION

An overlapping network constructed according to the activities' logical relationship must be modified in most cases by adding some technical relationships. The technical relationships are determined based on the rules of network construction, namely, each activity except for *Start* must have a preceding activity to define its start time and each activity except for *Finish* must have a succeeding activity to define its finish time. To achieve these rules, a three-step approach has been developed and is explained by the above example as follows:

Step 1: draw a two-dimensional diagram, and block the *Start* column, *Finish* row and all cells on the diagonal line as shown in Figure 2.

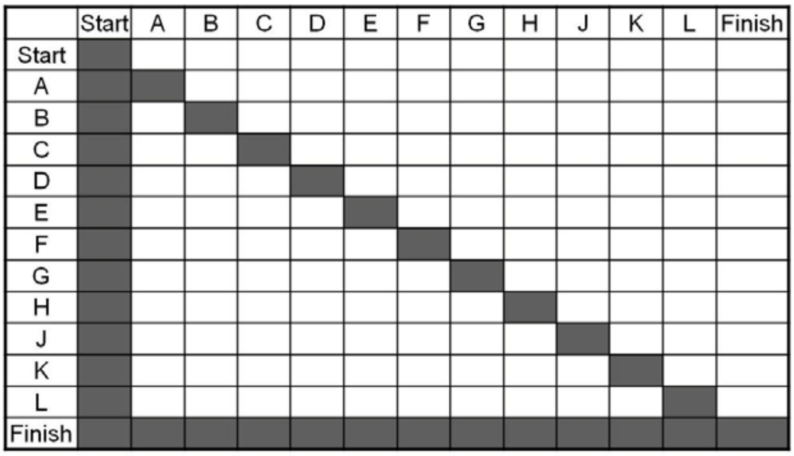

Figure 2: The initial two-dimensional diagram for relationship identification.

Step 2: Figure 3 shows the legends proposed to represent the logical relationships between two activities at their intersectional cell in the above two-dimensional diagram. The preceding activity can be any one listed on the first left-hand column and a succeeding activity is one activity listed at the top row in Figure 2. For example, there exists a FTS relationship from activity *A* to activity *C*, a small solid square is plotted at the left bottom corner of the intersectional cell of the A row and the C column. After all relationships given in Table 1 are added to Figure 2 according to the legends shown in Figure 3, a new two-dimensional diagram (Figure 4) is created as follows:

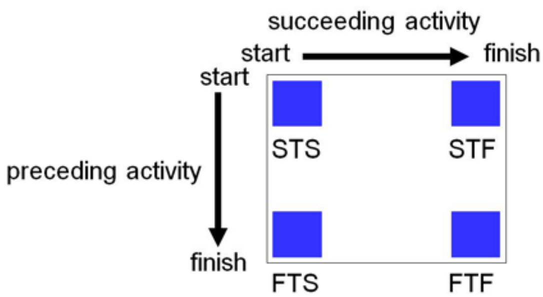

Figure 3: Legends for identifying activities' relationships.

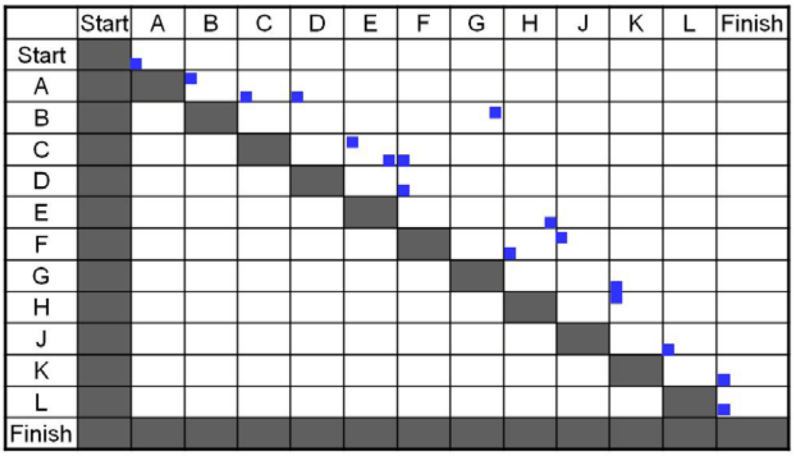

Figure 4: Identifications of logical relationships.

Step 3: add a technical FTS relationship (a blank square) from *Start* in the first left-hand column to the activity (succeeding activity at the first top row into Figure 4), which has neither STS nor FTS, at the second row; and add a technical FTS relationship from an activity (preceding activity at the first left-hand column), which has neither FTF nor FTS, to *Finish* in the top row. In total, three blank squares should be added for the finish-to-start relationships from *Start* to *G*, from *B* to *Finish* and from *H* to *Finish*. Figure 5 presents all logical and technical relationships.

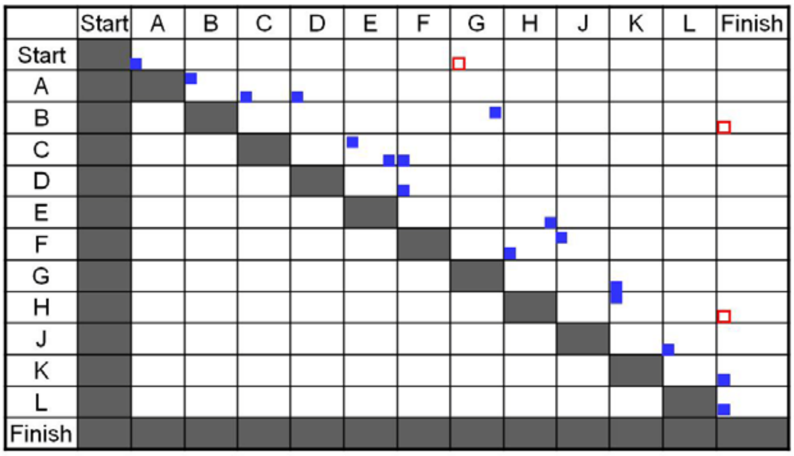

Figure 5: Identifications of logical and technical relationships.

The network modification procedure reflects the facts that the start time of an activity is not earlier than the time point of *Start* and the finish time of an activity is not later than the time point of *Finish*. Based on the identification of technical relationships, the activity *G* in the given example can start as early as the activity *A* can, and activities *B* and *H* can finish as late as activities *K* and *L* can. Following these discoveries, three dotted lines are added to the initial overlapping network to link *Start* to activity *G*, activity *B* to *Finish* and activity *H* to *Finish* as shown in Figure 6.

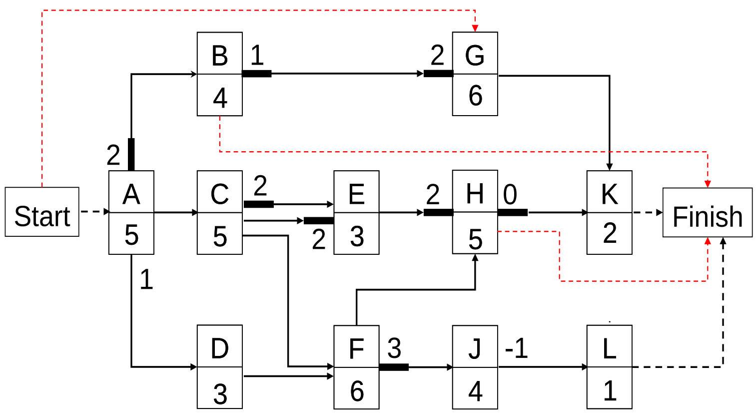

Figure 6: Overlapping network after modification.

All three relationships are finish-to-start relationships. In addition, repeating the above computational process, the correct time parameters of all activities of the modified network are calculated and shown in Table 3. Apparently, the proposed network modification process can avoid all errors appeared previously and guarantee the correction and rationality of calculated time parameter of all construction activities.

Table 3: Modified time parameters of construction activities.

Activity	Duration	ES	EF	LS	LF	TF	FF
Start	0	0	0	0	0	0	0
A	5	0	5	0	5	0	0
B	4	2	6	16	20	14	1
C	5	5	10	5	10	0	0
D	3	6	9	7	10	1	1
E	3	9	12	16	19	7	7
F	6	10	16	10	16	0	0
G	6	0	6	13	19	13	10
H	5	16	21	16	21	0	0
J	4	13	17	17	21	4	0
K	2	16	18	19	21	3	3
L	1	16	17	20	21	4	4
Finish	0	21	21	21	21	0	0

CONCLUSIONS

The overlapping network constructed only according to the logical relationships may be incorrect as demonstrated in the example used in this research. The modification process must be considered as a part of the construction of an overlapping network. This article introduces a novel process to identify the concealed technical relationships. The modification of an overlapping network is an imperative step in advance of computing time parameters to guarantee their corrections and rationality.

REFERENCES

1. Hinze, J.W., *Construction Planning and Scheduling*. New Jersey: Pearson Education (2004).
2. Newitt, J.S., *Construction Scheduling Principles and Practices*. New Jersey: Pearson Prentice Hall (2005).
3. Liu, C., Formulated scheduling procedure for repetitive construction projects. *Proc. 3rd Annual Project Management Australia Conference*, Melbourne, Australia (2006).
4. Arditi, D., Sikangwan, P. and Tokdemir, O., Scheduling system for high rise building construction. *Construction Management and Economics*, 20, **4**, 353-364 (2002).
5. Galloway, P.D., Comparative study of university courses on critical-path method scheduling. *J. of Construction Engng. and Management*, 132, **7**, 712-722 (2006).
6. Huang, R.-Y. and Sun, K.-S., Non-unit-based planning and scheduling of repetitive construction projects. *J. of Construction Engng. and Management*, 132, **6**, 585-597 (2006).
7. Abudayyeh, O., Dibert-Deyoung, A. and Jaselskis, E., Analysis of trends in construction research: 1985-2002. *J. of Construction Engng. and Management*, 130, **3**, 433-449 (2004).
8. Perera, S., Compression overlapping precedence networks. *J. of the Construction Division, American Society of Civil Engineers*, 108, **1**, 1-12 (1982).
9. Hegazy, T. and Wassef, N., Cost optimization in projects with repetitive nonserial activities. *J. of Construction Engineering and Management*, 127, **3**, 183-191 (2001).
10. Patrick, C., *Construction Project Planning and Scheduling*. Upper Saddle River: Pearson Prentice Hall (2004).
11. Uher, T.E., *Programming and Scheduling Techniques*. Sydney: UNSW Press (2004).