

# Algorithm for CPM Computations with Single Mode Activities

Kuldeep Singh Kushwah<sup>1</sup>, Sanjay Tiwari<sup>2</sup>

<sup>1</sup>ME CTM Student, MITS Gwalior

<sup>2</sup>Associate Professor, MITS Gwalior

**Abstract:** CPM is still the most popular network technique for construction projects. However CPM is very limited in scope when project deadline is restricted. This is because in conventional CPM each activity is assumed to have one fixed duration and corresponding normal cost. Thus the Project duration is fixed and project Manager do not have any control if the project is required to be completed within an earlier deadline as compared to project duration. The project duration is computed by summing the durations of only those activities that lie along the critical path. The project cost is the sum of cost of all activities of project. Hence Time cost trade-off technique is to be applied to CPM to overcome above restriction. For this CPM computations needs to be formulated. The paper presents an algorithm for CPM computations with single mode activities i.e. when each activity is assumed to have a fixed duration and cost.

**Keywords:** CPM

## 1. Introduction

In the literature project networks are represented by two different approaches: Activity on Node (AoN) representation and Activity on Arc (AoA) representation. Activity precedence relations are defined in terms of predecessor and successor activities. If start of activity *j* requires, activity *i* to be completed we say activity *i* is the predecessor of activity *j*, and activity *j* is the successor of activity *i*. If activity *j* can start immediately after activity *i*, then activity *i* is the immediate predecessor of activity *j*, and activity *j* is the immediate successor of activity *i*.

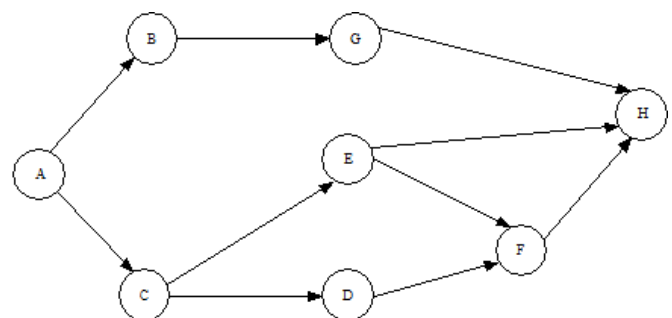
### Activity on Node (AoN) Representation

In AoN representation, the activities are shown by nodes. Arcs represent the immediate precedence relations. The direction of an arc shows the precedence direction. Assume we have an 8 activity project having precedence relations as shown in Table 1. AoN representation of the project network is given in Figure 1.

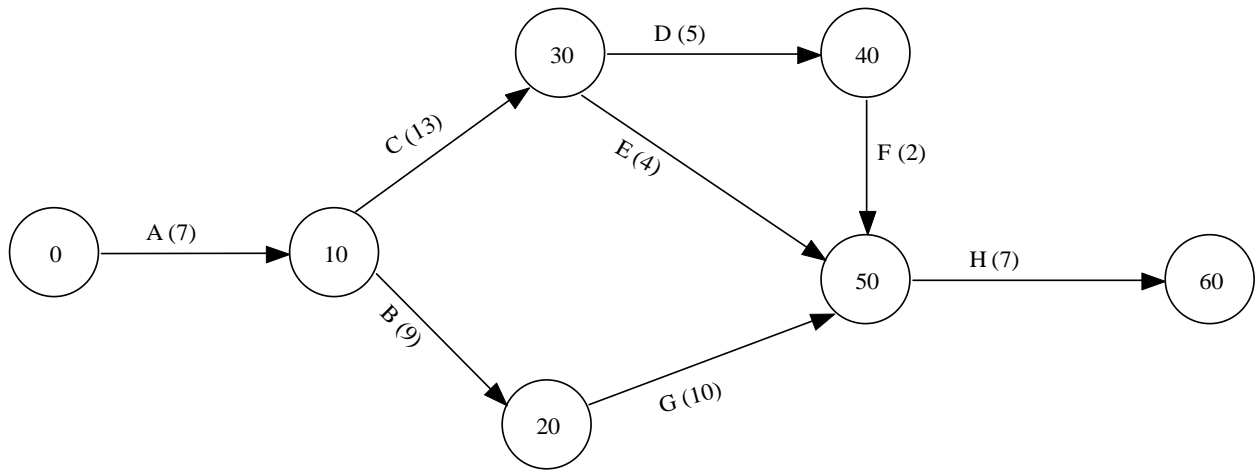
**Table 1:** Example Project Network

Activity	Immediate Predecessor	Immediate Successor
A	-	B,C
B	A	G
C	A	E,D
D	C	F
E	C	H
F	D	H
G	B	H
H	G,E,F	-

Note that activities A and B are predecessors of activity G, and activities F and H are successors of activity D. According to AoA representation scheme the activities are shown by arcs. All activities have a source node and a sink node. The source node of an activity represents an event of starting that activity whereas its sink node represents an event of completing that activity. The arcs representing the immediate successor activities of an activity should start from the sink node of the activity. Therefore, the corresponding node becomes the source node of the immediate successor nodes. In project networks, parallel arcs are not allowed, hence in some cases it is required to use dummy activities. AoA representation is more widely used to demonstrate project networks. AoA representation for the sample network, whose data are given in Table 1, is shown in Figure 2. (Numbers within parenthesis are duration of activities).



**Figure 1:** Activity on Node Representation of Example Network



**Figure 2** Activity on Arc (AoA) Representation

In Figure 2, node 0 is an event for the start of activity A, therefore, the start of the project. Node 30 represents the completion of activity C and start of activities D and E. Node 60 represents the completion of activity H, therefore, the completion of the entire project. The completion time of the last activity H defines the project completion time. The project completion time is also referred to as project duration.

In conventional CPM an activity has got one fixed duration and corresponding normal cost. Thus when there is single mode for each activity the problem is called *Single-mode* problem.

## 2. Single-Mode Problems

The problem is to determine the project duration. To find minimum project duration Critical Path Method (CPM) is used. Since there is only one mode for each activity, total project cost is simply the summation of costs of all activities. For single-mode problems the parameters that define the problem are precedence relations and duration of activities. The duration of activity,  $i$  is denoted by  $D_i$ . The set of immediate predecessors {successors} of activity  $i$  is  $P(i) \{ S(i) \}$ . The aim is to find the minimum project duration  $T$ . For a given project, the following information can be derived:

$S_i$ : Start time of activity  $i$

$LS_i$ : Latest completion time of activity  $i$

**Slack time of activity  $i$ :** Maximum possible duration of activity  $i$  without increasing the project duration. It is simply the difference between latest possible completion and earliest possible start time.

**Critical Activity:** The activity whose slack time is equal to its duration. In other words, any activity is called critical if an increase in its duration directly affects the project duration. All activities which are not critical are called noncritical activities. The set of critical activities are denoted as **Crit**.

To summarize, we have the following relations:

$$\text{Slack}_i = LC_i - S_i \quad i = 1, 2, \dots, N$$

$$\text{crit} = \{ i = 1, 2, \dots, N \mid \text{Slack}_i = D_i \}$$

**Critical Path:** A path from source to sink consisting of all critical activities.

### Algorithm:

```

Initialization:
 $S_i = 0; P_i = \phi$ 

Main Body:
Repeat
 $S_j = \text{MAX}_{i \in P_i} \{ S_i + D_i \}; \forall i \in P_i; S_j$  is calculated
Until  $S_i$  for  $i = 1, 2, \dots, n$  are calculated
 $T = \text{MAX}_i \{ S_i + D_i \}$ 
 $LC_i = T; S_i = \phi$ 

Repeat
 $LC_i = \text{Min}_{j \in S_i} \{ LC_j - D_j \}; \forall j \in S_i; LC_i$  is calculated
Until  $LC_i$  for  $i = 1, 2, \dots, n$  are calculated
Finalization:
 $\text{Slack}_i = LC_i - S_i \quad i = 1, 2, \dots, n$ 

 $\text{crit} = \{ i = 1, 2, \dots, n \mid \text{Slack}_i = D_i \}$ 
    
```

The CPM algorithm works as follows:

In initialization step, the earliest start times of the activities without predecessors are set to 0. Then the earliest start times of other activities are calculated. For each activity, the earliest start times of all immediate predecessors are known. The earliest start time of activity is the maximum of the earliest completion times of its predecessors. After all earliest start times are computed, the earliest project duration is found, which is the maximum of all earliest completion times. The latest completion times are then computed, starting from the activities having no successors. The latest completion time of an activity is the minimum of the latest start times of all its immediate successors. After all earliest start and latest completion times are found, the slack times are computed and the critical activities are detected. Below is the stepwise description of the CPM.

We illustrate the CPM method on the project instance whose data were given in Table 1. We now give the stepwise implementation of the method.

### Initialization:

Activity A has no immediate predecessor. Therefore  $S_A = 0$

**Main Body:**

We start by checking the activities that immediately succeed activity A. These activities are B, C. Activities B and C have single predecessor.  $S_B$  and  $S_C$  are calculated as follows:

$$S_B = S_C = S_A + D_A = 0 + 7 = 7$$

Using  $S_B$  and  $S_C$ , we calculate  $S_D$ ,  $S_E$  and  $S_G$

$$S_D = S_E = S_C + D_C = 7 + 13 = 20$$

$$S_G = S_B + D_B = 7 + 9 = 16$$

$S_F$  is calculated using  $S_D$

$$S_F = S_D + D_D = 20 + 5 = 25$$

Activity H has 3 predecessors. Therefore,

$$S_H = \text{Max}\{S_F + D_F; S_E + D_E; S_G + D_G\} = \text{Max}\{25 + 2; 20 + 4; 16 + 10\} = 27$$

Since all ES values are calculated, the project duration is calculated

$$T = \text{Max}\{S_i + D_i\} = 27 + 7 = 34; LC_G = 34$$

As a result, the project duration is calculated.

Now it is time to calculate the latest completion time values.  $LC_E$ ,  $LC_F$ ,  $LC_G$  are calculated using  $LC_H$

$$LC_E = LC_F = LC_G = LC_H - D_H = 34 - 7 = 27$$

Similarly,  $LC_B$  and  $LC_D$ , are calculated

$$LC_B = LC_G - D_G = 27 - 10 = 17$$

$$LC_D = LC_F - D_F = 27 - 2 = 25$$

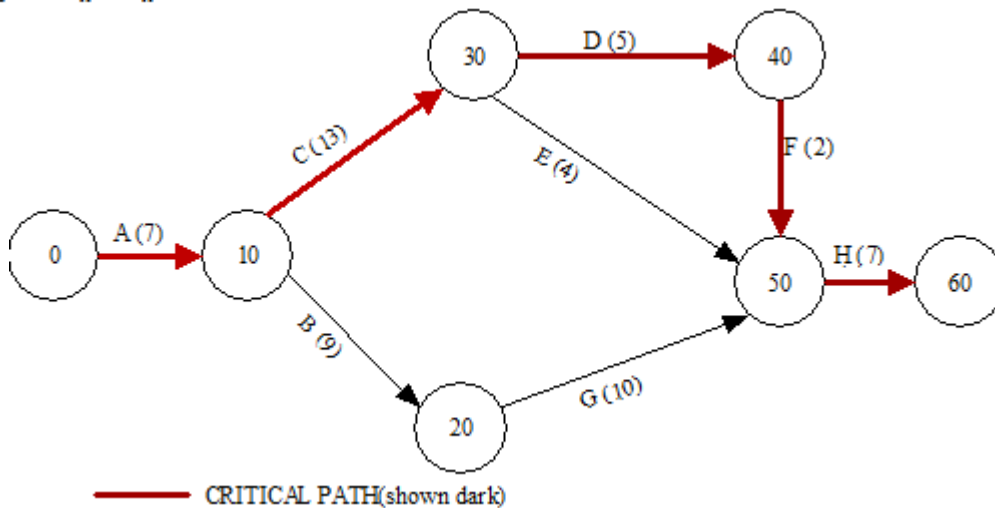
Activity C has 2 immediate successors. Therefore,

$$LC_C = \text{Min}\{LC_E - D_E; LC_D - D_D\} = \text{Min}\{27 - 4; 25 - 5\} = 20$$

Activity A has 2 immediate successors.

$$LC_A = \text{Min}\{LC_B - D_B; LC_C - D_C\} = \text{Min}\{20 - 13; 17 - 9\} = 7$$

All latest completion times and slack times are calculated and tabulated in Table 3. From the table, it can be observed that the slack times and durations of the activities 1, 2, 4, 6 and 8 are equal. These activities are critical, and the path defined by these activities is the critical path. The critical activities in Figure 2.4 are shown by solid lines.



**Table 3: CPM Calculations of an Example**

Activity	Pred.	Succ.	Duration	ES	LC	slack
A	-	B,C	7	0	7	7
B	A	G	9	7	17	10
C	A	E,D	13	7	20	13
D	B	F	5	20	25	5
E	B	H	4	20	27	7
F	D	H	2	25	27	2
G	C	H	10	16	27	11
H	E,F,G	-	7	27	34	7

Observe that  $T = D_A + D_C + D_D + D_F + D_H = 34$ . Any change in the duration of any critical activity directly affects the project duration. On the other hand, the duration of activity E is 4, and the slack time of that activity is 7. This means that the duration of activity could be increased by 3 units without increasing the project duration. In other words, one can delay the start time of activity E by 3 units without increasing the project duration.

**3. Conclusion**

The paper presents a simple algorithm for cccCPM computation and illustrate it with an example. The algorithm is being converted in to a computer code using C++.

**References**

- [1] Akkan,C., Drexl,A., and Kimms, A.,(2005b), "Networkdecomposition-based benchmarkresultsforthe discretetime-costtradeoffproblem", European Journalof OperationalResearch,165,pp.339-358.
- [2] Crowston,W.B.,andThompson,G.L.,(1967), "Decision CPM:A method for simultaneous planning,scheduling, andcontrolofprojects",OperationsResearch,15,pp.407-426.
- [3] Elmaghraby, S.E., (1993), "Resource allocation via dynamic programming inactivity networks",European Journal of Operational Research,64,pp.199-215.
- [4] Hinze, J. W., 2008, "Construction planning and scheduling", third edition, Pearson Education, Inc., Upper Saddle River, New Jersey
- [5] Murty, K. G. (1992). Network Programming. Englewood Cliffs, NJ: Prentice- Hall.
- [6] Rothfarb, B., Frank, H., Kleitman, D.M., Steiglitz, K., and Rosenbaum, D.M., (1970), "Optimal design of offshore

- natural-gas pipeline systems”, Operations Research, 18, pp. 992–1020.
- [7] Stevens, J. D., 1990, Techniques for Construction Network Scheduling, Stanford University.
- [8] Wysocki, R., 2007, “Effective project management”, Fourth edition, Wiley publishing, Inc., Indiana.