Monte Carlo Simulation for Understanding
Risk in Project Management

*Dr. Deepak Subedi, **Dr. Uday Tate

* Professor of Quantitative Methods and Operations Management College of Business Marshall University Huntington, USA
**Professor of Marketing College of Business-Corbly Marshall University Huntington, USA

ABSTRACT

This article develops an algorithm for assessing the risk in project management. The main risk considered here is the risk of a project not finishing on time. This algorithm can be considered as an extension of the classical methods like Critical Path Method (CPM) and Project Evaluation and Review Technique (PERT). This extension can provide a richer understanding of risk inherent in any project planning.

Specifically, the present article discusses the classical project management tools such as CPM and PERT by pointing out to their applications as well as shortcomings in the context of risk management. It then shows the steps to conduct a Monte Carlo simulation of CPM using EXCEL spreadsheet. To conduct the simulation, first CPM is recast as a couple of optimization (linear programming) problems. And then the scripts written in Visual Basic for Applications (VBA) in Excel is used to run the CPM for one hundred times. The simulation gives a range of project completion times as well as the probability for each activity becoming a part of critical path.

Keywords: Project Management, Critical Path Method, Linear Programming, Risk, Simulation

INTRODUCTION

A project can be defined as a set of connected activities, which has definite beginning and ending. However, finishing any project on time and within the budget limits has always been a challenge. Hence, many organizations (e.g., DuPont) and government agencies (e.g., U. S. Department of Defense)? early on developed and refined algorithms to schedule activities in a CPM project (Lawrence et al., 1998, pp 272-283 and Pich et al., 2002).

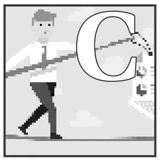
Critical Path Method (CPM), developed by Dupont, computes the time required to complete a project, including the starting and ending time for each of the activities. It also focuses on the order in which the activities can be started as inputs for the calculation. For example, an excavation must be completed before laying a foundation, or wiring must be completed before any inspection. Then, based on this information, a network diagram is developed. This diagram is used to find the critical path. The critical path is the longest path in the network. The project completion time is equal to the sum of durations of all the activities in the critical path (Critical Path Method, Lawrence et al., 1998, pp 272-283).

In any project, there is always a risk that a project may not complete on time (Jamshadia et al., 2017; Galli, 2017, Pich et al., 2002, Pritsker, 1966). This contrasts with the assumption of CPM, which requires that the duration for each of the activities is deterministic. Therefore, CPM as such cannot evaluate risk inherent in a project. To overcome or mitigate this deficiency, PERT was developed. Here the completion time of each of the activities is taken as stochastic. Hence the project completion time is stochastic as well (Lawrence et al., 1998, pp 293-299).

However, PERT has its own limitation. When the time required to complete each of the activities is stochastic, the critical path itself can change. However, PERT takes a critical path determined by CPM as given. A Monte Carlo simulation of CPM can be a better solution in such situations. The simulation can be used not only to find a range of project completion times, but also to see the probabilities of each of the activities falling on the critical path (Pich et al., 2002).

The purpose of this article is to develop an algorithm to understand and assess the risks embedded in a project management endeavor. Importantly, the article provides a deeper understanding of risk assessment in any project planning and demonstrates the ease of programming and user-friendliness as the notable features of the algorithm developed in this article.

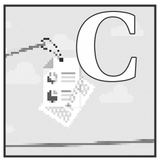
Following, Lawrence et al., this article utilizes Linear Programming to find the critical path as well as project completion time (1998, pp 299-300). I used, SOLVER which is an add-in tool in EXCEL (2001, pp 171-182; https://www.solver.com/excel-solver-linear-programming) to run the Linear programming. Then, following Albright (2001) I developed VBA scripts for the Monte Carlo Simulation.



ONCEPTUAL FRAMEWORK

This section explains CPM and PERT with an example and elaborates on the logic behind the simulation. The following example illustrates the CPM/PERT concepts as well the simulation run. The example is shown in Table 1. This example-the part of Table 1 and the corresponding CPM-is mostly based on Wikipedia page on Critical Path Method (https://en.wikipedia.org/wiki/Critical_path_method). However, it should be noted that the author has extended the original one to present an argument on PERT and simulation. The conceptual parts of CPM and PERT can be found in the works of Lawrence et al. (1998, pp 272-283 and pp 293-299).

According to Table 1, there are eight activities in the project. The amount of time taken by each of the activities is given. In CPM only the average time is considered. For example, time to complete activity 'A' is 10. It is considered deterministic. The units of time can be anything (e.g., minutes, hours, etc.). This article defines the unit as day. Furthermore, “predecessors” are the tasks that must be completed before starting a specific activity. For example, Activity A should be completed before staring B. Similarly, both predecessors D and H must be completed before starting E, and so on.



RITICAL PATH METHOD

As stated previously, the purpose of CPM is to find the start and end time for each of the activities and the minimum time required to complete the project. The project completion time is the sum of the time required by the activities in the critical path (Lawrence et al., 1998, pp 272-283).

The network diagram in Figure 1 shows the critical path method. Each of the nodes (circles) represent an activity. For

TABLE 1: ACTIVITIES AND DURATION

		Duration					
Activity	Predecessor	Mean	Min	Max	Median	Standard	Variance
A	-	10	4	20	9	2.67	7.11
B	A	20	10	30	20	3.33	11.11
C	B	5	1	9	5	1.33	1.78
D	C	10	3	21	9	3.0	9
E	D, H	20	10	30	20	3.33	11.11
F	A	15	5	25	15	3.33	11.11
G	F, C	5	2	10	4.5	1.33	1.78
H	A	15	4	30	14	4.33	18.78

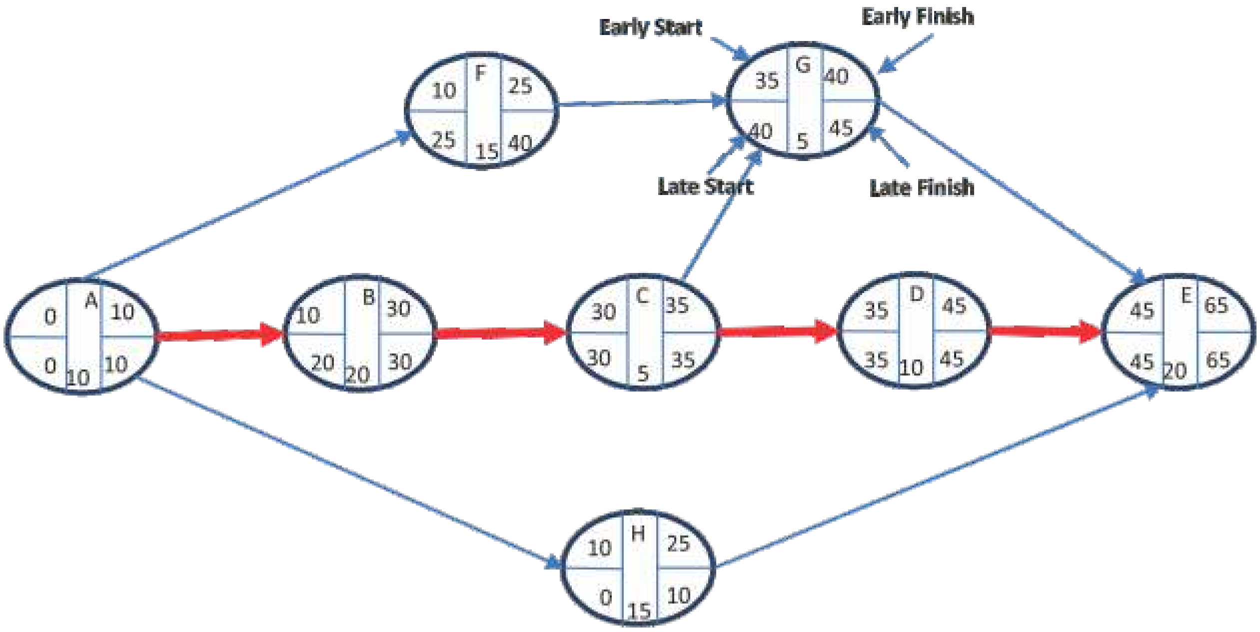


FIGURE 1: FINDING THE PROJECT COMPLETION TIME WITH CPM

example, Activity G is represented by the node that is designated as G. It takes 5 days to complete. Therefore, if this activity starts on day 35, it can finish on day 40. The reason it can only start on day 35 is dictated by the completion time of its two predecessor activities. Activity C ends on day 35 and F at 25. Since both preceding activities must be completed before G starts, it can only start on day 35.

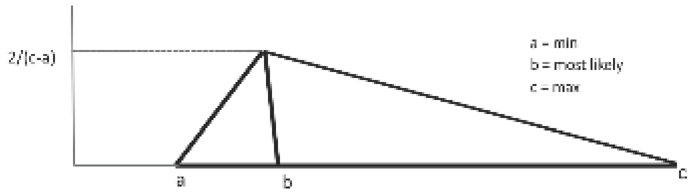
Moreover, the late finish time is related to the late start time of the succeeding activity. E is the succeeding activity of G, and it can start as late day 45. Therefore, G can have a late finish time of 45, leading to a late start time of 45-5 = 40. Hence, there is a 5 day slack between late start and early start (or late finish and early finish).

Activities on the critical path do not have slack. That is, any delay in any of these activities can lead to a delay in project completion time. The critical path consists of activities A, B, C, D and E. The number of days required to finish the project is 65 (10+20+5+10+20) (Lawrence et al., 1998, pp 272-283).

PROJECT EVALUATION AND REVIEW TECHNIQUE

One must recognize that a completion time for each of the activities cannot be deterministic. At best, it is the average time drawn from a wide range of possibilities. Generally, completion time for each of the activities is considered as having a triangular distribution. This is illustrated in Figure 3. In this distribution, data for three sets of times (minimum, maximum and most likely) are collected. Mean and standard deviation of each of the time periods can be calculated as following: - The above figure needs some work in its display and role.

$$Mean = \frac{Min + Max + 4\,Most\ Likely}{6},\quad Standard\ deviation = \frac{Max - Min}{6}.$$



Recall that the time estimates of each of the activities in Table 1 are the means calculated with the formula given above. Table 1 also gives the standard deviations and variances of each of the activities. Variances are the square of their respective standard deviation. The estimate of the project completion time (65 days) calculated above can be considered as the mean, with 50% chance that it can be longer and another 50% chance to be shorter. Like averages, a variance of the completion time is the sum of variances of each of the activities in the critical path. To find the standard deviation of the path, we should find its variance first and take its square root [Lawrence et al., 1998, pp 293-299]. The value of standard deviation turns out to be 6.33. Based on the mean and standard deviation, the range of 99% confidence interval of the project completion time can be calculated in the range of 50 to 80.

LIMITATIONS OF PERT

Like CPM, PERT has its own limitations. For example, it assumes that the critical path as originally identified from CPM remains unchanged. However, the completion time of each of the activities, not just those in original critical path, are stochastic (and not deterministic). Two alternative solutions of CPM are shown in Figures 2 and 3. They are calculated using the same data from Table 1. Here, the completion time of each of the activities is randomly drawn from the triangular distribution of the activity times. These figures show that, not just the project completion time, but the critical path is also different from that shown in Figure 1. In Figure 2, the critical path is ABCGE and the project completion time is 57.3 days,

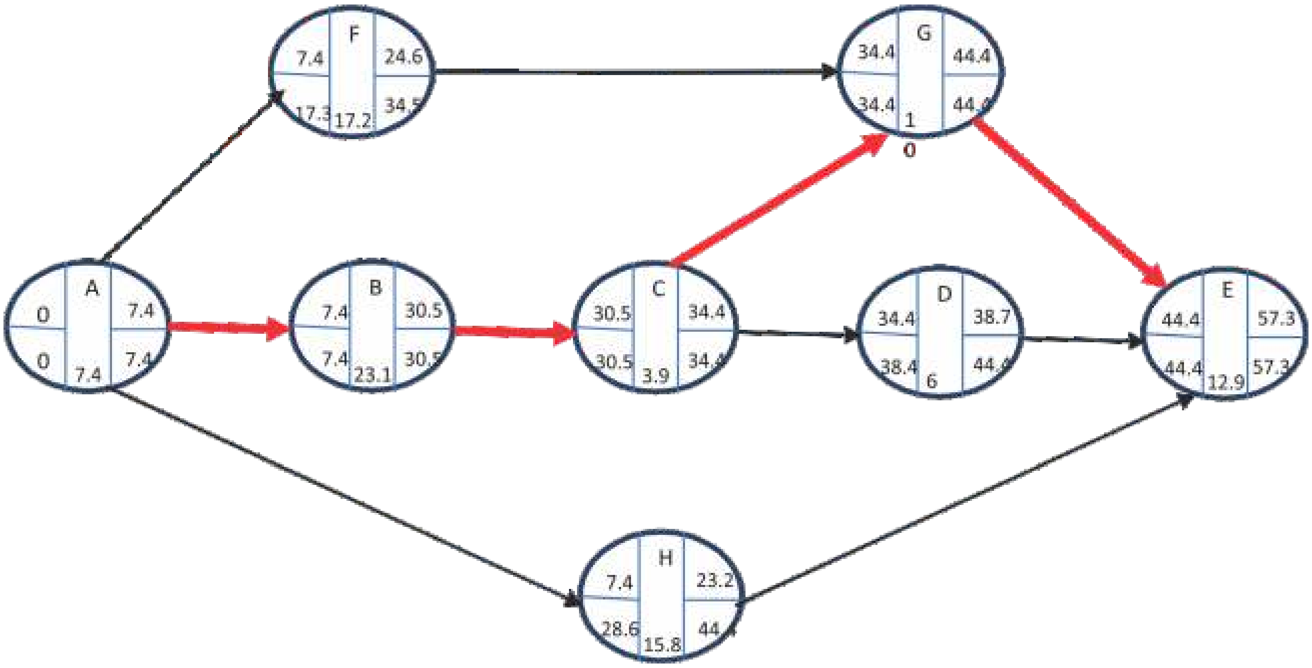


FIGURE 2: FINDING THE PROJECT COMPLETION TIME WITH CPM-ALTERNATIVE ACTIVITY TIMES

whereas in Figure 3 the critical path is AFGE and the project completion time is 61.9 days. Therefore, there is a need for finding, not just one critical path, but evaluating the probability of each of the activities falling in the critical path.

To overcome the above limitations, one can use a Monte Carlo simulation (Pich et al, 2002). To achieve this, MS Excel 2016 along with its SOLVER option, can be used to conduct the simulation. This process generates CPM one hundred times in succession. For each iteration, the duration of each of the activity is drawn randomly. It not only gives the range of project completion times, but also the probabilities of each of the activities falling in the critical path, thus, giving a more complete picture of project completion resulting in a better understanding and analysis of project risk management.

MONTE CARLO SIMULATION

To conduct the Monte Carlo simulation, an example with 22 activities is used (Gido and Clements, 2013, pp 177). However, the present author has added a set of minimum, maximum and median durations which are required for the Monte Carlo simulation. This approach has resulted in a more realistic example. Table 2 shows the results from the example.

This Simulation consists of the following steps.

Step 1. Find the project completion time and the critical path for using estimated durations. Conceptually, one can draw CPM diagram like the one shown in Figure 1, but for practical purpose, the details are given in a table-format only. Table 3

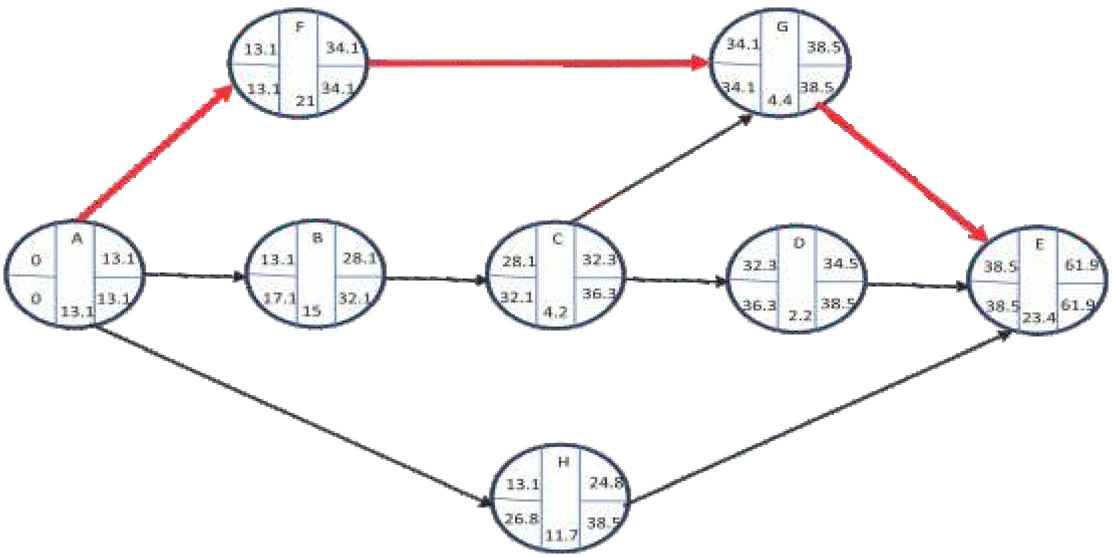


FIGURE 3: FINDING THE PROJECT COMPLETION TIME WITH CPM -ALTERNATIVE ACTIVITY TIMES

has the details.

Step 2. Randomly draw the duration for each of the activity. The numbers are drawn based on the triangular distribution, and within the range of maximum, minimum and median as given in Table 2.

Step 3. Repeat Steps 1 and 2 one hundred times.

Step 4. Calculate the range of project completion times, and the probability that each of the activities falls in the critical path. The details of the outcome of these steps are shown in Tables 4 and 5.

Following paragraphs elaborate on the above steps.

CPM USING EXCEL SOLVER

Finding the critical path and the project completion time is the most crucial step in this algorithm. Making a network diagram like Figure 1 is not viable here, because a diagram with 22 activities can be very burdensome.

The current article used Liner Programming (LP), aided by EXCEL SOLVER to get the critical path and the project completion time. Two Linear Programs are formulated following Lawrence et al. (1998, pp 299-300). The first one finds the minimum completion time of the project (this number is called "FINISH" here) and the early start time of each of the activities. In general form, it is shown below:

$Min \sum_{i=1}^{m} x_i^s + x_{fin}$ Where, x_i^s is the early starting time of i^{th} activity, and x_{fin} is the time when project ends.

S.T. $x_i^s - x_{i-1}^s - d_{i-1}$ $\forall x_i^s$ Where, d_i is the duration of the i^{th} activity

$x_i^s \geq 0 \quad \forall x_i^s$

The output of the above formula is the minimum project completion time x_{fin}. The value of x_{fin} is a constant "FINISH."

TABLE 2: ACTIVITIES AND DURATION OF LONGER EXAMPLE

#	Activity	Immediate Predecessors	Durations					
			Estimated	Minimum	Maximum	Median	Standard Deviation	Variance
1	A		3	1.000	6.000	2.750	2.041	4.167
2	B		4	2.000	7.000	3.750	2.041	4.167
3	C	A, B	1	0.500	1.500	1.000	0.408	0.167
4	D	C	5	3.000	8.000	4.750	2.041	4.167
5	E	C	8	6.000	10.000	8.000	1.633	2.667
6	F	D	5	3.000	7.000	5.000	1.633	2.667
7	G	E, F	1	0.500	2.000	0.875	0.612	0.375
8	H	G	8	4.000	12.000	8.000	3.266	10.667
9	I	G	10	6.000	12.000	10.500	2.449	6.000
10	J	H, I	2	1.000	4.000	1.750	1.225	1.500
11	K	J	2	1.000	4.000	1.750	1.225	1.500
12	L	J	15	8.000	19.000	15.750	4.491	20.167
13	M	J	10	4.000	12.000	11.000	3.266	10.667
14	N	J	6	4.000	8.000	6.000	1.633	2.667
15	O	L, M, N	2	0.500	6.000	1.375	2.245	5.042
16	P	O	6	3.000	10.000	5.750	2.858	8.167
17	Q	O	4	2.000	8.000	3.500	2.449	6.000
18	R	O	4	2.000	8.000	3.500	2.449	6.000
19	S	P, Q, R	1	0.500	5.000	0.125	1.837	3.375
20	T	S	4	1.000	9.000	3.500	3.266	10.667
21	U	S	2	1.000	4.000	1.750	1.225	1.500
22	V	T, U	1	0.500	3.000	0.625	1.021	1.042

For the problem given in table 2, the value of "FINISH" is 57. Early start of each of the activities are given by x_i^e. In the case of the problem given in table 2, x_i^e changes from x_A^e to x_V^e. The output x_A^e to x_V^e are the early start times of each of the activities. The EXCEL spreadsheet and the related SOLVER inputs are shown in Appendices 2a and 2b.

$Max \sum_{i=1}^{n} x_i^l + x_{fin}$ Where, x_i^l is the late starting time of i^{th} activity, and x_{fin} is the time when project ends.

S.T. $x_i^l - x_{i-1}^l \geq d_{i-1}$ $\forall x_i^l$ Where, d_i is the duration of the i^{th} activity

$x_{fin} <= FINISH$

$x_i^l >= 0$ $\forall x_i^l$

In the second LP, the main issue is to find the late start time for each of the activities. Therefore, the objective here is to maximize the time. But, when it is maximized without constraint, it may go to an infinite level and this problem cannot be solved. To control this problem, one extra constraint x_{fin}<=FINISH is added. The Excel spreadsheet and the related SOLVER inputs are shown in Appendix 3a.

To be able to automatically run the simulation, SOLVER is run using VBA codes (Albright, 2001, pp 173-183). These codes are given in Appendices 2c and 3b respectively.

Finally, the time slacks of each of the activities are calculated. The formula for the slack is $x_i^l - x_i^e$. Slacks are zero for activities on the critical path. The output of this exercise is shown in Table 3. According to it, the activities with 0 slack are in critical path. The Critical Path is B-C-D-F-G-I-J-L-O-P-S-T-V. The project completion time thus calculated is 57. For PERT the standard deviation thus calculated is 8.216, with a 99% confidence interval of the project completion time as 43.49 to 78.16.

RANDOMLY DRAWN DURATION OF ACTIVITIES

The first step of the simulation is to randomly draw a duration for each activity. Since there is no built-in function for a triangular distribution in EXCEL, a special code was developed for this function. First, a random number must be drawn. To do this, one can use built-in functions in Excel or VBA. Specifically, these functions draw a randomly generated number, ranging from 0 to 1 from the uniform distribution. This number is called "U."

The value of a randomly drawn duration depends on U. Besides; it also depends upon the ratio of $\frac{b-a}{c-a}$, which is called "d" here. If d is greater than U, then the duration of the activity is equal to $(a + (c - a) * \sqrt{dU})$; otherwise it is equal to $(a + (c - a) * [1 - \sqrt{(1 - d)(1 - U)})$ (Albright, 2001, pp 160-161 and pp 169; Triangular Distribution). The code of the function developed in the article to randomly draw the duration of each activity is given in Appendix 1.

SIMULATION AND THE RESULT

Referring to the Monte Carlo Simulation steps, Step 3 runs Steps 1 and 2 one hundred times (Albright, 2001, pp 173-183). The result is the hundred different instances where the project completion times are calculated; the result also shows the critical paths for each of the instances. Table 4 summarizes the statistics from this simulation based on Step 3. It demonstrates that the average completion time is different from what was calculated with CPM and the standard deviation is different from what was calculated with PERT. In this case, average is higher and that is because the critical path itself can change. However, the standard deviation is smaller because many variables can cancel each other's volatility.

TABLE 3: CRITICAL PATH AND PROJECT COMPLETION TIME

#	Activity	Immediate Predecessors	Time		
			Early Start	Late Start	Slack
1	A		0	1	1
2	B		0	0	0
3	C	A, B	4	4	0
4	D	C	5	5	0
5	E	C	5	7	2
6	F	D	10	10	0
7	G	E, F	15	15	0
8	H	G	16	18	2
9	I	G	16	16	0
10	J	H, I	26	26	0
11	K	J	28	55	27
12	L	J	28	28	0
13	M	J	28	33	5
14	N	J	28	37	9
15	O	L, M, N	43	43	0
16	P	O	45	45	0
17	Q	O	45	47	2
18	R	O	45	47	2
19	S	P, Q, R	51	51	0
20	T	S	52	52	0
21	U	S	52	54	2
22	V	T, U	56	56	0
23	Finish	V	57	57	0

TABLE 4: SUMMARY STATISTICS

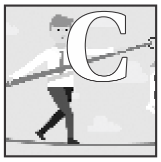
Mean	59.94789
Standard Deviation	3.96609
Maximum	69.86159
Minimum	51.75411
Median	59.97126

Further, Table 5 shows the percentage of time each of the activities can be in the critical path. It shows about 30% of the time Activity A can be in the critical path. Similarly, 25% of the time Activity H can be on the critical path as well, whereas, Activity P is on the critical path for 65% of the time. In fact, except for Activities K and N, each of the activities can be in the critical path. The original critical path, B-C-D-F-G-I-J-L-O-P-S-T-V as calculated by CPM, remained the critical path for only

TABLE 5: % OF TIME EACH ACTIVITY IS IN CRITICAL PATH

#	Activity	Immediate	% of time
1	A		30
2	B		70
3	C	A, B	100
4	D	C	94
5	E	C	6
6	F	D	94
7	G	E, F	100
8	H	G	25
9	I	G	75
10	J	H, I	100
11	K	J	0
12	L	J	94
13	M	J	6
14	N	J	0
15	O	L, M, N	100
16	P	O	65
17	Q	O	18
18	R	O	17
19	S	P, Q, R	100
20	T	S	91
21	U	S	9
22	V	T, U	100

on the average about 30% of the time in the simulations run.



ONCLUSION

The purpose of this exercise was to understand the risk, especially the probabilities of delays, in project management. There are two main benefits of this method. First, as part of project management, CPM and PERT are described as the problem-solving tools for managers. The Monte Carlo Simulation, by providing additional insights beyond CPM and PERT, can provide and enhance managers' understanding of the risk pertaining to project completion times. Furthermore, EXCEL spreadsheets and SOLVER are widely available. These are tools used extensively by academicians and practitioners of many disciplines with ease. Therefore, the algorithm that is demonstrated in this article can be a useful tool for both students as well as practitioners.

Indeed, there are other methods and procedures available to assess risks in project management (Jamshidia et al., 2017; Pritsker, 1966). But these other methods require sophisticated software and complicated calculations. On the other hand, the algorithm discussed here can be easily applied to understand risk in project management. However, there is a caveat: focussing on one kind of risk (time), while not considering other risks in project management (Galli, 2007), can be considered as the main limitation of the algorithm developed in this article. Future research effort is needed to extend the present algorithm to address other types of risk as well.

REFERENCES

i Albright, S. Christian (2001), VBA for Modelers Developing Decision Support Systems with Microsoft Excel, Duxbury, Thomson Learning

ii Critical Path Method, https://en.wikipedia.org/wiki/Critical_path_method as seen on 6/15/2017

iii Gido, Jack and James P.Clements (2013), Successful Project Management, 6e, Cengage Learning

iv Galli, Brian J (2017), Risk Management in Project Environments: Reflections on the Standard Process, Journal of Modern Project Management, September/ December, 40-49

v Jamshidia, Afshin, Daoud Ait-Kadib, Angel Ruizc (2017), And Advanced Dynamic Risk Modeling and Analysis, Journal of Modern Project Management, May/August, 6-11

vi Lawrence, John A Jr and Pasternack, Barry A. (1998), Applied Management Science A Computer-Integrated Approach for Decision Making, John Wiley and Sons Inc.

vii FrontLineSolvers, https://www.solver.com/excel-solver-linear-programming, 2018.

viii Pich, Michael T, Christoph H Loch, and Arnoud De Meyer. (2002) "On uncertainty, ambiguity, and complexity in project management." Management science 48.8: 1008-1023.

ix Pritsker, A. A. B. (1966), GERT: Graphical Evaluation and Review Technique, RM-4973-NASA. National Aeronautics and Space Administration, Rand Corporation, https://www.rand.org/content/dam/rand/pubs/research_memoranda/2006/RM4973.pdf as seen on 5/14/2014

x Triangular Distribution, https://en.wikipedia.org/wiki/Triangular_distribution as seen on 6/15/2017

APPENDICES

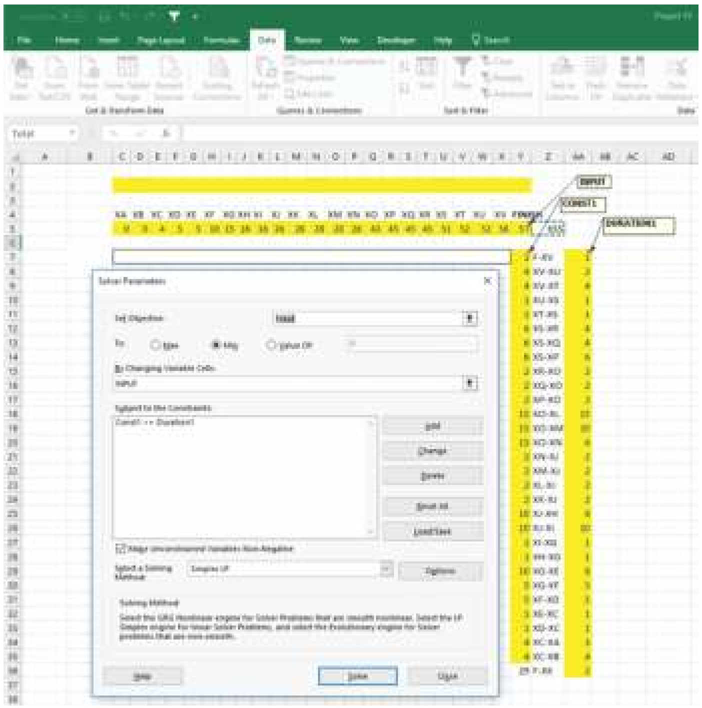
APPENDIX 1

```
'This is a function to randomly generate the activity times
'based on triangular distribution

Function ActTime(Maximum As Single, Minimum As Single, MostLikely As Single) As Single

Dim d As Single, x As Single

d = (MostLikely - Minimum) / (Maximum - Minimum)

x = Rnd()

If x < d Then

   ActTime = Minimum + (Maximum - Minimum) * Sqr(d * x)

  Else

   ActTime = Minimum + (Maximum - Minimum) * (1 - Sqr((1 - d) * (1 - x)))

End If

End Function
```

APPENDIX 2 A

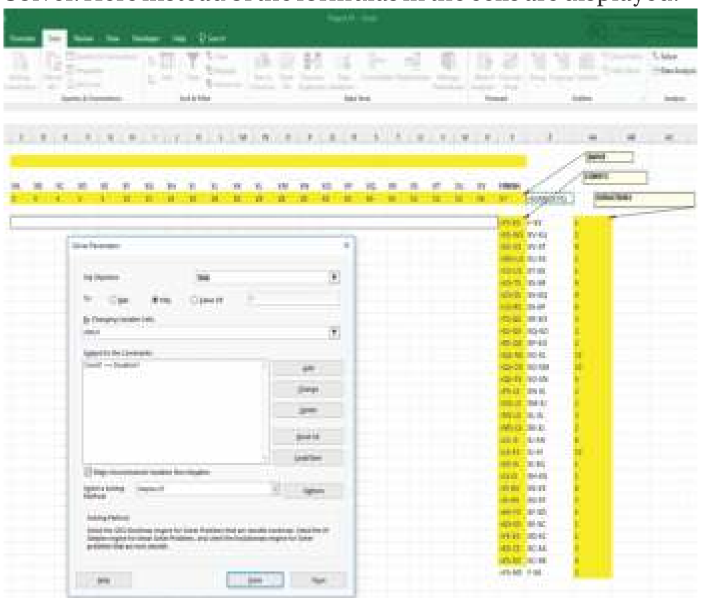
This shows the first LP formulation in Excel spreadsheet and Solver.

The numbers shown in the INPUT range are the early start time for each of the activities. The number under FINISH is the time required to complete the project.



APPENDIX 2 B

This shows the first LP formulation in Excel spreadsheet and Solver. Here instead of the formulas in the cells are displayed.



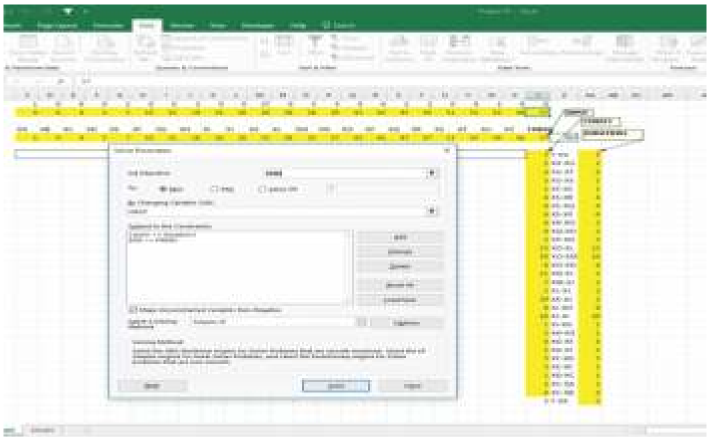
APPENDIX 2 C

```
'There are two outcomes of this sub
'to find the minimum time required to complete the project
'to find the early start time of each of the activities
Sub Minimize()
SolverReset
SolverOk SetCell:=Range("Total"), MaxMinVal:=2, ByChange:=Range("INPUT")
SolverAdd CellRef:=Range("CONST1"), Relation:=3, FormulaText:="Duration1"
SolverOptions AssumeLinear:=True, AssumeNonNeg:=True
SolverSolve UserFinish:=True
'Transfer the values of early start time in the next row
'This is clear the current row for the next run of LP
Range("INPUT").Copy Destination:=Range("C2:Y2")
'Copy the project completion time in another cell
```

Appendix 3 A

This shows the second LP formulation in Excel spreadsheet and Solver.

The numbers shown in the INPUT range are the late start time for each of the activities. The number under FINISH is the time required to complete the project.



APPENDIX 3 B

```
'This sub finds the late start time of each of the activities
Sub Maximize()
  SolverReset
        SolverOk SetCell:=Range("Total"), MaxMinVal:=1, ByChange:=Range("INPUT")
        SolverAdd CellRef:=Range("CONST1"), Relation:=3, FormulaText:="Duration1"
        'One constraint is added to make sure that project completion time remains the same
        SolverAdd CellRef:=Range("INPUT"), Relation:=1, FormulaText:="FINISH"
        SolverOptions AssumeLinear:=True, AssumeNonNeg:=True
    SolverSolve UserFinish:=True
End Sub
```