

A Generalized PERT/CPM Implementation in a Spreadsheet

Kala C. Seal
College of Business Administration
Loyola Marymount University
Los Angeles, CA 90045, USA

kseal@lmumail.lmu.edu

Abstract

This paper describes the implementation of the traditional PERT/CPM algorithm for finding the critical path in a project network in a spreadsheet. The problem is of importance due to the recent shift of attention to using the spreadsheet environment as a vehicle for delivering MS/OR techniques to end-users.

Introduction

Samuel Bodily (Bodily, 1986) suggested the use of spreadsheets for solving the Management Science (MS) and Operations Research (OR) problems so that the techniques become more accessible to practitioners. Spreadsheets provide a natural interface for model building, are easy to use in terms of inputs, solutions and report generation, and allow users to perform what-if analysis. The ensuing decade has seen a tremendous growth in the use

of spreadsheets for solving various MS/OR problems both by the practitioners and by the academic community (Leon et. al, 1995, Seal et. al, 2000). However, none of the work so far has presented a generalized method for implementation of the PERT/CPM method in a spreadsheet using the critical path algorithm. The few available approaches that use the well known critical path method for solving the problem and use spreadsheet as the medium (Eppen et. al, 1998; Hillier et. al, 2000; Ragsdale, 1998) do not permit easy extendibility of the model when activities are added or deleted from the network or when the predecessor relationships are altered. Some authors circumvent this issue by transferring the PERT/CPM problem to the underlying linear programming (LP) and then solving this LP through the solver module available in modern spreadsheets (Plane, 1994; Hesse, 1997; and Ragsdale, 2001). This paper shows a way of solving PERT/CPM problems in a spreadsheet using the critical path (the longest path) algorithm. The approach is easily extendible and does not use LP transformation.

The Problem

We will use a simple example problem of organizing a seminar with various activities. The completion time of each activity is not known with certainty; only estimates are available. The details of the activities, their time estimates and the predecessor relationships are presented in Table 1.

Table 1: The data for the example problem.

Using the approaches found in most textbooks (e.g. Hillier et. al, 2000, Ragsdale 2001), the approximate expected duration and approximate variance of activity duration for each of the activities are calculated and are shown in last two columns of Table 1.

Methods

The project network is shown in Figure 1. The method used for calculating the critical path revolves around the simple concept of representing a network with a matrix. Any graph or network consisting of n nodes can be represented by an $n \times n$ matrix where each row and each column of the matrix represents a node in the graph and an entry of 1 in a cell of the matrix represents a directed arc between the corresponding nodes. An example of the matrix representation is provided in Table 2 for the project network given in Figure 1. The network representation here uses the activity-on-node convention.

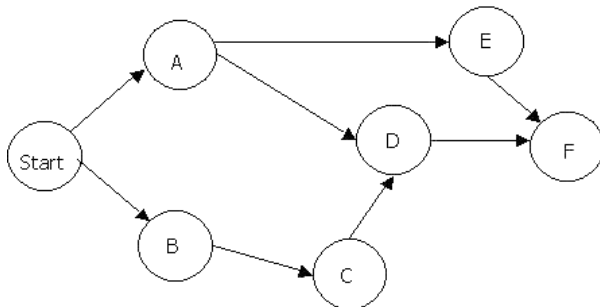


Figure 1: Project Network

We will adopt the convention of representing the predecessors in the columns and successors in rows throughout this paper.

The implementation of the PERT/CPM method in the spreadsheet is achieved by using a total of four worksheets in an Excel Workbook. The first one is used for the input data; the next two are used respectively to calculate the Earliest Start (EST) and the Earliest Finish Time (EFT), and the Latest Start (LST) and Latest Finish Time (LFT) of the activities; and the fourth sheet is used for reporting the critical path as well as for calculating the average project completion time and the project variance.

Two additional worksheets are used after solving the problem for displaying a Gantt chart for the project and the range names used in the spreadsheet.

ACTIVITIES	Immediate	Predecessors Relationships					
	Predecessors	A	B	C	D	E	F
A							
B							
C	B		1				
D	A, C	1		1			
E	A	1					
F	D, E				1	1	

Table 2: Matrix representation of the project network and the predecessor relationships

The first step of the process is to organize the input data in the spreadsheet. This entails laying out all the given information in Table 1 as well as representation of the network with the predecessor relationships as shown in Table 2, in the spreadsheet. The spreadsheet layouts of the inputs for the example problem are shown in Figures 2 and 3 below.

After entering the input data, the procedure starts calculating the EST and the EFT pairs for the activities in the network. This is accomplished by copying the network matrix into another sheet (named EST-EFT) and then adding two more columns to keep track of the final EST and EFT values of the activities. The layout of this sheet is shown in Figure 4.

The earliest start time of an activity is the maximum of all the EFTs of its immediate predecessor activities. The spreadsheet is constructed following that principle. For any activity pair i and j , i in the column and j in the row, the cell (i,j) in the matrix contains the EFT of activity i if i is an immediate predecessor of j , zero otherwise. We will look at activity D in the EST-EFT sheet as an example. The cell E7, representing the pair A and D, has a value of 3.00, the EFT of A, because A is an immediate predecessor of D. The cell F7, on the other hand, contains zero because B is not an immediate predecessor of D. Conditional formatting is used to make the zero entries appear as hatched cells to improve

1	A	B	C	D	E	F	G	H
2	ACTIVITIES	Description	Immediate Predecessors	Optimistic Time (a)	Most Likely Time (m)	Pessimistic Time (b)	Expected Value $(a + 4m + b)/6$	Variance $[(b-a)/6]^2$
3	A	Plan Topic		1	3	5	3.00	0.44
4	B	Obtain Speakers		2	4	6	4.00	0.44
5	C	List Meeting Locations	B	3	6	9	6.00	1.00
6	D	Select Location	A,C	1	5	9	5.00	1.78
7	E	Finalize Speakers Schedule	A	1	2	3	2.00	0.11
8	F	Prepare and mail Seminar Brochures	D,E	2	8	20	9.00	9.00

Figure 2: Layout of the Input data in the spreadsheet

1	A	B	C	Predecessors Relationships					
2	ACTIVITIES	Description	Immediate Predecessors	A	B	C	D	E	F
3	A	Plan Topic							
4	B	Obtain Speakers							
5	C	List Meeting Locations	B		1				
6	D	Select Location	A,C	1		1			
7	E	Finalize Speakers Schedule	A	1					
8	F	Prepare and mail Seminar Brochures	D,E				1	1	

Figure 3: The Predecessor Relationship Representation in Spreadsheet

the appearance of the EST-EFT results. The formula used for calculating the entries in the cells of the matrix is shown below using the cell E7, representing the activity pair A and D, as an example.

Cell E7:
 $=IF(Inputs!I6=1,VLOOKUP(E\$7,EST_EFTtable,3,False),0)$

Here EST_EFTtable refers to range A4:C203 in the EST-EFT sheet. The range up to C203 is used to accommodate up to 200 activities in the spreadsheet without renaming the range.

The EST for an activity is the maximum of all the EFTs of its immediate predecessors. Therefore, the

EST for activity A in cell B4 is given by the formula

Cell B4: $=MAX(E4:GV4),$

which is copied to the cells B4 through B11 for finding the EST of the remaining activities (<http://ite.informs.org/vol2no1/Seal/pertcpm.xls>).

Once the ESTs are obtained, the EFT is found by adding the expected activity time to the EST. For example, the formula used for finding the EFT for activity A is

Cell C43:
 $=B4+VLOOKUP(A4,Inputs,7,False)$

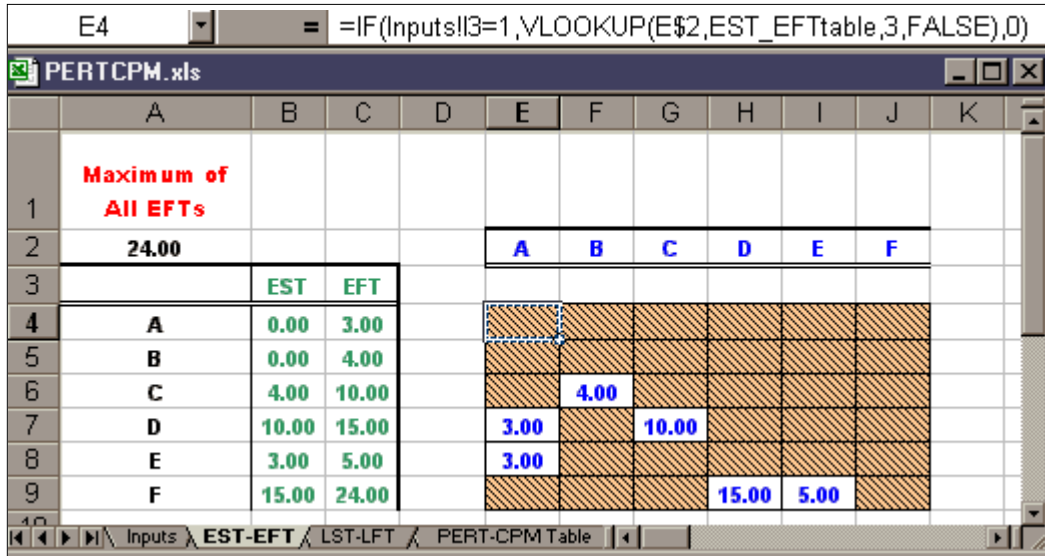


Figure 4: The EST-EFT Calculations

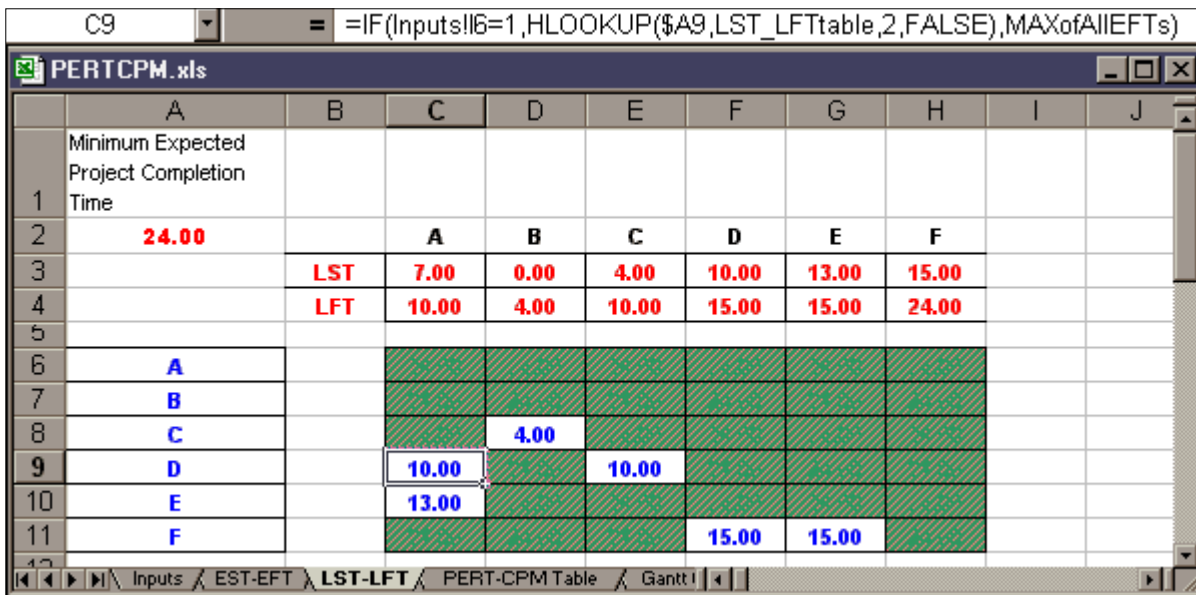


Figure 5: The LST-LFT Calculations

which is then copied to the cells C4 through C11. The VLOOKUP function is used to extract the expected completion time of the activity from the input data in the range A3:H203 (named as Inputs).

The next step in the process is to find the latest start time (LST) and latest finish time (LFT) for each activity. Noting that the LFT of an activity with successors must be the minimum of the LSTs of all the successor activities, we can use the same logic

that we employed to calculate the EST-EFT pairs for an activity, only in reverse. This is accomplished in the LST-LFT sheet shown in Figure 5. Once again a matrix with the activities in the rows as well as in the columns is created. The activities in the rows represent the successors. This is consistent with the predecessor representation followed in the paper. Two extra rows are inserted in this sheet to store the values of the LST-LFT pairs for each activity. In

addition, a cell (A2 in the EST-EFT sheet) is used that stores the value of the maximum of the EFTs of all the activities.

For an activity pair i and j , i in the column and j in the row, the cell(i,j) in the matrix is the LST of j if j is an immediate successor of i , maximum of all the EFTs (content of cell A2 in the EST-EFT sheet) otherwise. The following formula demonstrates the calculation for the activity pair A and D.

```
Cell C9: =IF(Inputs!I6=1,
HLOOKUP($A9,LST_LFTtable,2,FALSE),MAXofALLEFTs)
```

The formula is copied to the other cells of the matrix in the LST-LFT sheet. The LST_LFTtable here refers to cell range C2:GT4 in the LST-LFT sheet. Conditional formatting has been used again to hatch out the cells with the LST equal to the maximum of all EFTs to improve the presentation of the results. A horizontal lookup function is used here instead of the vertical lookup.

Once all the cells of the matrix are calculated the final LFT for an activity can easily be selected by finding the minimum of all the LSTs of its immediate successors. The formula for the LFT of activity A, for example, is given by

```
Cell C4: =MIN(C6:C205)
```

The LST is calculated by subtracting the activity completion time from the LFT. The formula used for the activity A is

```
Cell C3: =
C4-VLOOKUP(C2,Inputs,7,FALSE)
```

which is then copied to other cells in the range C3:H4 of the LST-LFT sheet. The VLOOKUP function is used to extract the activity completion time from the inputs (Excel Workbook).

Once the LST_LFTtable is completed, the minimum expected project completion time is obtained by taking the maximum of all the LFTs. Cell A2 in the LST-LFT sheet contains that value and the formula for that cell is

```
Cell A2: =MAX(C4:GT4).
```

If there is only one critical path, then the construction of the final CPM table and the determination of

the critical activities is straightforward. The (EST, EFT) and the (LST, LFT) pairs for the activities are extracted from sheets EST-EFT (Figure 4) and LST-LFT (Figure 5) and the total slacks are calculated by subtracting the EST from LST. If the result is zero, then the activity is a critical activity. The variances of the critical activities are added to get the overall variance in the project completion time. The NORMDIST function and the NORMINV function are then used to calculate the probability for a target completion time and the possible completion time for a target probability, respectively. Figure 6 and Table 3 show the layout of the final result and the corresponding cell formulas, respectively.

It is to be noted that the slack in this table refers to total slack (or total float) for an activity as opposed to the free slack. The total slack of an activity is the maximum amount of time the activity can be delayed from its earliest start time *without delaying the critical path*. The free slack, on the other hand, refers to the maximum allowable delay from the earliest start time of an activity *without delaying the earliest start time of any of its immediate successor activities*. The formula for automatically calculating the free slack in the spreadsheet can get quite complicated. We thus advise the readers to calculate the free slack for each activity manually after the CPM table is obtained from the spreadsheet.

The above method for the final construction of the CPM table will not be fully correct when there are multiple critical paths for the problem. In such a case the basic method will end up correctly identifying all the critical activities on all critical paths and thus will add the variances of all of them to get the overall project variance. Clearly that would lead to a wrong answer. It is possible to develop formulas in the spreadsheet for identifying the existence of multiple critical paths, but such a procedure makes the overall spreadsheet quite complex. We have, therefore, decided to leave it out from the paper. Our advise to the readers who may use the method presented in this paper is to identify all the critical activities in the pictorial representation of the project network to see if there are multiple critical paths and then calculate the final critical path length and the variance accordingly.

Usage and Extendibility

The method presented in this paper of using a spreadsheet to solve PERT/CPM problems by the critical path algorithm is a useful addition to professors teaching MS/OR courses. Spreadsheet programs are widely available now and managers are familiar and comfortable with the software. As a result, MS/OR courses in many business schools are being taught using spreadsheets as the software

tool. This paper makes a definite contribution to that effort. Depending on the goals, level, and the pace of the course, the model can either be used as a black box or as a detailed illustration of the mechanics of the PERT/CPM method.

Unlike the existing spreadsheet based PERT/CPM approach, which require the rebuilding of the model for each new network through a new linear programming formulation, our model is easily extendable

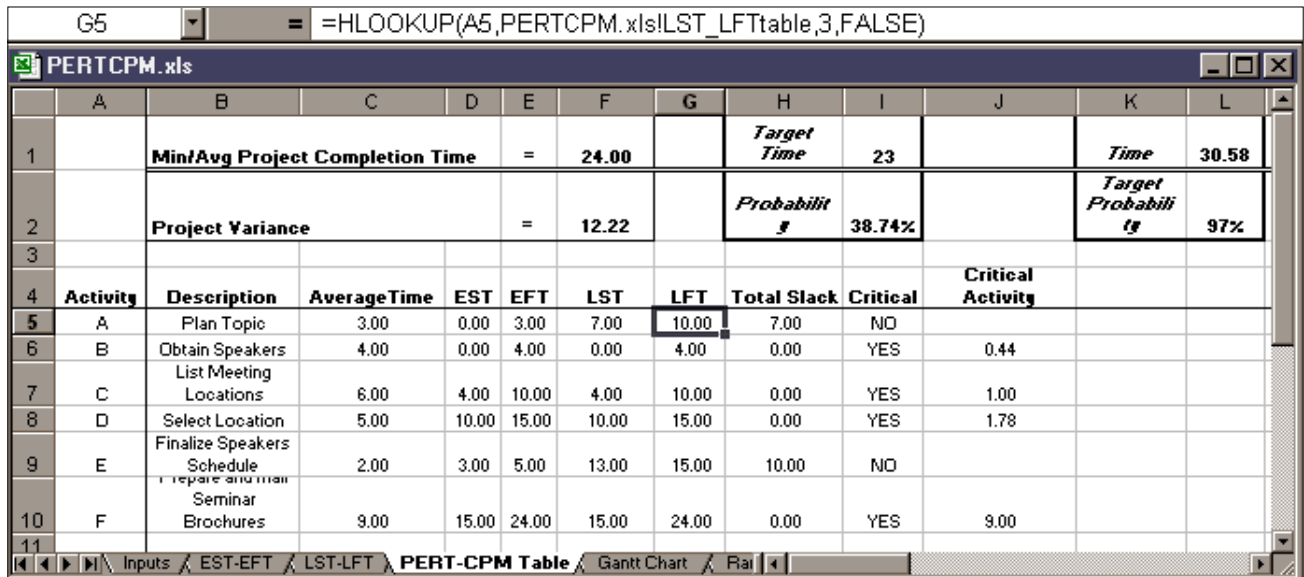


Figure 6: The Final CPM Table

Cell Reference	Formula	Copied To
F1	=MAX(G5:G204)	--
F2	=SUM(covariance))	--
I1	=23	--
I2	=NORMDIST(I1,F1,SQRT(F2),TRUE)	--
L1	=0.97	--
L2	=NORMINV(L2,F1,SQRT(F2))	--

A5	=Inputs!A3	A6 through A13
B5	=Inputs!B3	B6 through B13
C5	=Inputs!G3	C6 through C13
D5	=EST-EFT!B3	D6 through D13
E5	=D5+C5	E6 through E13
F5	=G5-C5	F6 through F13
G5	=HLOOKUP(A5,PERTCPM.xls!LST_LFTtable,3,FALSE)	G6 through G13
H5	=F5-D5	H6 through H13
I5	=IF(H5>0,"NO", "YES")	I6 through I13
J5	=IF('PERT-CPM Table'!I5="Yes", (VLOOKUP('PERT-CPM Table'!A5,Inputs,8,FALSE)), " ")	J6 through J13

Table 3: The formulas used in the "CPM Table" sheet

and can accommodate changes in the number of activities and changes in the predecessor relationships. If the range-names are defined at the initial set up anticipating a larger number of project activities then the extension from n activities to $n+1$ activities takes just a few copy commands. An example of extending the project network used in this paper with addition of another activity "G" after the current last activity "F" is shown in the Appendix (also refer to Excel Workbook with Extension). For introductory MS/OR courses, the spreadsheet can be set up to accommodate a large number of activities appropriate for the largest size of problem considered in the course. This would avoid the explanation of the extendibility issues to the students since it would only entail the students to provide the new data in the Inputs sheet. This is in contrast to any of the existing spreadsheet based PERT/CPM approaches where the set up is problem specific and thus has to be redone for each new problem.

Our method can also address an inherent problem with the PERT calculations. The time estimates that are used for calculating the mean and variance of the completion time of each activity are only approximations and this approach tends to provide a pretty rough approximation of the mean and variance. As a result, the assumption that the mean critical path identified by the method used in this paper will turn out to be the longest path through the project network may not hold. There is a significant chance that some other path will turn out to be longer than the mean critical path and thus the project manager who relies on the calculated probability from the spreadsheet can be badly misled (see Hillier et. al, 2000 for an extensive discussion on the evaluation of the assumptions in PERT). However, with the approaches described in this paper it is very easy to simulate a project completion time instead of using the probability calculation based on the approximate critical path. One can replace the formulas for average activity durations with formulas that draw random samples from a specified distribution for each activity duration. For example, one can use a triangular distribution for the mean activity duration with minimum = a , mode = m , and maximum = b . The simulation could then be replicated using the standard data table approach

described in many management science textbooks or using an add-in package such as @Risk or Crystal Ball. It is much simpler to simulate a project in this way than with the Linear Programming formulation since the later needs running Solver during each replication.

Another area of interest in the project management is evaluation of the effect of project crashing and the corresponding time-cost trade offs. The optimum value of time-cost trade-offs is an LP formulation and many authors have shown how to do that in the spreadsheet (e.g. Hillier et. al, 2000; Ragsdale, 2001). All the approaches suffer from the incapability to accommodate changes in the network structure (either due to change in precedence relationship or for addition or deletion of activities) without extensive manual changes in the formulas. We could not find a way to incorporate the crashing and the time-cost trade-off calculations in our method without encountering the same limitations and therefore, it is not described in this paper.

Teaching Experience

We used our model in introductory MBA level Management Science class for three semesters. The model was presented to the students as a black box with the EST-EFT sheet and the LST-LFT sheet hidden. For the interested students, the details of the method were available through the course web site. The students used the Excel model for both PERT as well as CPM calculations. For CPM, the times estimates were all made equal. The responses from the students have been very positive. The students liked the fact that they did not have to use any other program other than spreadsheet for any part of the course. In the first use of the model, the students in the class themselves identified the issue of the existence of multiple critical paths in the solution and the problems associated with it. Some students who were familiar with the MS Project software also liked our method as they could appreciate that our method eliminated the learning curve for a new software application and made it easy to simulate the project completion time. The simulation in MS Project needs either the @Risk add-in for MS Project or VBA coding.

Conclusions

Spreadsheet software has gained popularity as a modeling environment for managers interested in building and using MS/OR models for decision-making. Various techniques have been proposed for different areas of the MS/OR, however, the area of PERT/CPM lacked a generalized framework for implementation in a spreadsheet. This paper shows a way of accomplishing that task by using some simple but innovative spreadsheet layouts and formulas. The model presented not only solves the PERT/CPM problem by following the critical path method, it is also easily extendable when activities are added and deleted or predecessor relationships are changed. With a few simple copies the entire model can be extended from n activities to $n+1$ activities. It is, of course, possible to avoid the use of the critical path method in solving PERT/CPM problems by using LP formulation and then invoking the built-in spreadsheet LP solver. However, that introduces an extra step in the modeling process and also masks the more efficient critical path algorithm. This paper addresses these issues by maintaining the critical path algorithm yet providing a general framework for solving PERT/CPM problems.

Acknowledgement

The author would wish to thank the Associate Editor, three anonymous referees and Dr. Zbigniew Przasnyski for their valuable suggestions on the paper.

References

- Bodily, S. (1986), "Spreadsheet modeling as a stepping stone," *Interfaces*, Vol. 16, No. 5, pp. 34-52.
- Eppen, G. D., F. J. Gould, C. P. Schmidt, J. H. Moore, and L. R. Weatherford (1998), *Introductory Management Science: Decision Modeling with Spreadsheets*, 5th Edition, Prentice Hall.
- Hesse, R. (1997), *Managerial Spreadsheet Modeling and Analysis*, IRWIN.
- Hillier, F. S., M. S. Hillier and G. J. Lieberman (2000), *Introduction to Management Science*, Irwin-McGraw-Hill, Boston, MA.
- Leon, L., Z. H. Przasnyski and K. C. Seal (1995), "Spreadsheets and MS/OR Models: An End-User Perspective," *Interfaces*, Vol. 26, pp. 92-104.
- Plane, D. R. (1994), *Management Science: A Spreadsheet Approach*, Boyd and Fraser.
- Ragsdale, C. T. (2001), *Spreadsheet Modeling and Decision Analysis: A Practical Introduction to Management Science*, 3rd edition, South-Western College Publishing.
- Seal, K. C., Przasnyski, Z. H., and Leon, L. (2000), "A Literature Review of MS/OR Models in Spreadsheets", *OR Insight*, Vol 13, Issue 4, pp. 21-31.

Appendix - Example of Extendibility

Let us assume that the example problem presented in this paper needs to be revised with the addition of one more activity "G" and the resulting project network after the addition of this new activity is as shown in Figure A1. It is further assumed that the optimistic, most probable, and the pessimistic time estimates for the completion of "G" are given and they are 5, 10, and 15 weeks, respectively.

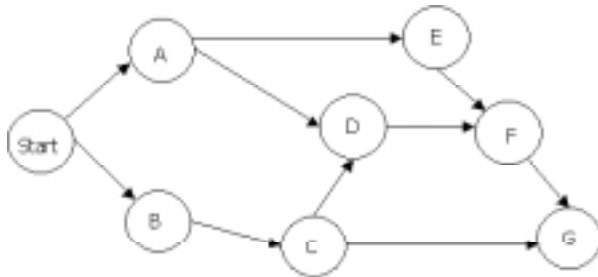


Figure A1: Project Network with Additional Activity G

The following steps need to be carried out to extend the example problem with this new activity.

In the Inputs sheet, the content of row 8, the row with the current last activity (activity F) is copied to the next row down (to row 9) and the column with

the current last activity F (column N) is copied to its right (to column O). This extends the input matrix for accommodating one more activity.

The activity name in the new row (row 9) is changed from F to G and the time estimates and predecessor matrix are updated with proper information for this new activity. The new "Inputs" sheet is shown in Figure A2 and Figure A3.

In the "EST-EFT" sheet, row 9, the current last row with an activity in it, is copied to row 10 and then column J, the current last column with activity F, is copied to column H. The order of copying is important to avoid circular reference. Since the activity names in this sheet are connected to the activity names on the input sheet (the formula in cell A4 of this sheet is "Inputs!A3" which is copied down the rows for the other activities and the formula for cell E2 is "Inputs!I2" which refers to the activity names at the top of the predecessor matrix in the "Inputs" sheet. It is then copied across the other columns), no other editing is necessary. The resulting "EST-EFT" sheet with the updated EST-EFT pairs for all the activities is shown in Figure A4.

In the "LST-LFT" sheet, the current last column with an activity (Column H) is copied to the next column (column I) and then row 11 is copied down to the next row. Once again the order of copying must be

	A	B	C	D	E	F	G	H	Pi
1									
2	ACTIVITIES	Description	Immediate Predecessors	Optimistic Time (a)	Most Likely Time (m)	Pessimistic Time (b)	Expected Value (a + 4m + b)/6	Variance [(b-a)/6]²	
3	A	Plan Topic		1	3	5	3.00	0.44	
4	B	Obtain Speakers		2	4	6	4.00	0.44	
5	C	List Meeting Locations	B	3	6	9	6.00	1.00	
6	D	Select Location	A, C	1	5	9	5.00	1.78	
7	E	Finalize Speakers							
8	F	Schedule	A	1	2	3	2.00	0.11	
9	G	Prepare and mail Seminar Brochures	D, E	2	8	20	9.00	9.00	
10									
11									
12									
13									

Figure A2: Input Sheet with New Row for activity G

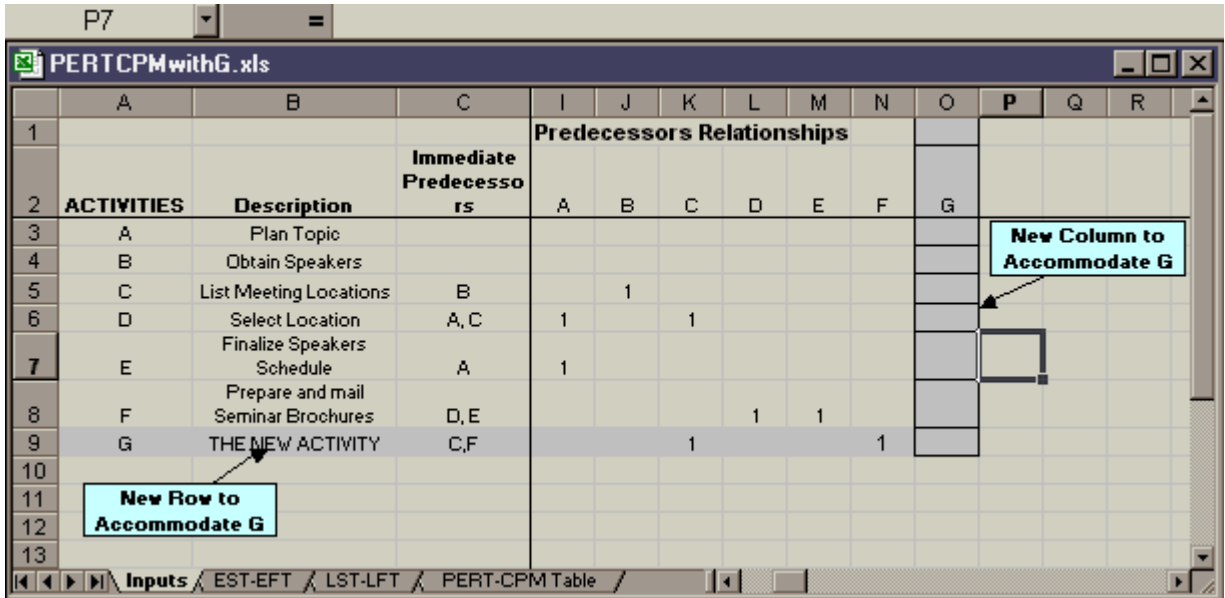


Figure A3: Input Sheet with New Column for activity G

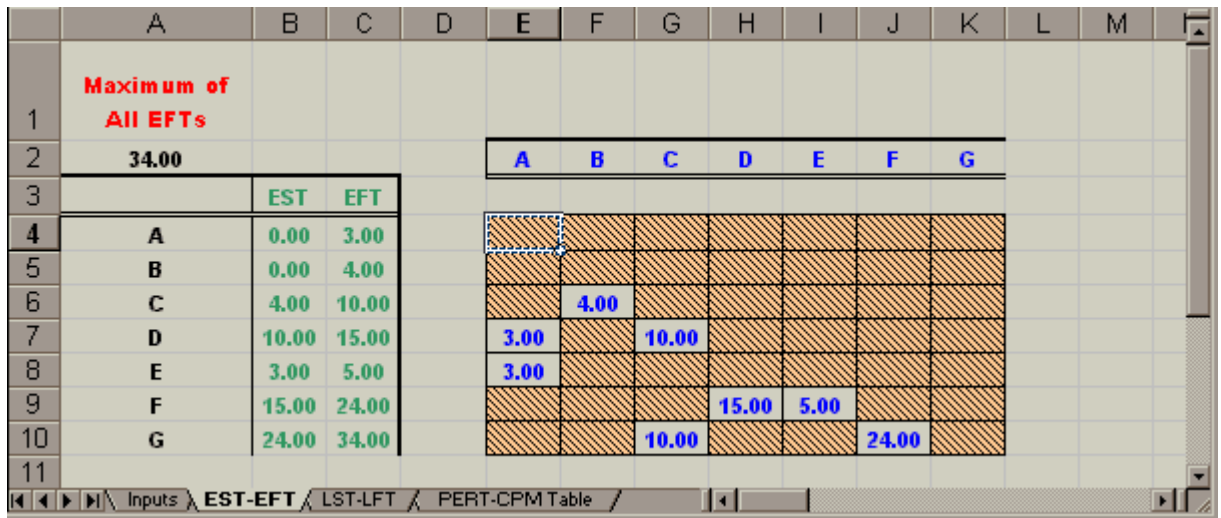


Figure A4: Updated EST-EFT after addition of Activity G

maintained to avoid circular references. Again no extra editing is needed to update the names of the activities since the cells used for the designating the names of the activities in the "LST-LFT" sheet are connected to the corresponding cells in the "Inputs" sheet. The resulting "LST-LFT" sheet with the updated LST-LFT pairs for all the activities is shown in Figure A5.

Finally the CPM Table is updated by copying of the current last row (row 10) in the "PERT-CPM Table" sheet to the next row down (row 11). The updated result is shown in Figure A6.

For the expansion of the network with the addition of more activities, the range-names do not have to be redefined because they are initially set up to accommodate a network of 200 activities. The

1	Minimum Expected Project Completion Time											
2	34.00											
3		A	B	C	D	E	F	G				
4		LST	7.00	0.00	4.00	10.00	13.00	15.00	15.00	24.00	24.00	
5		LFT	10.00	4.00	10.00	15.00	15.00	24.00	24.00	34.00	34.00	
6	A											
7	B											
8	C											
9	D											
10	E											
11	F											
12	G											

Figure A5: Updated LST-LFT sheet after addition of activity G

1		Min/Avg Project Completion Time	=	34.00				Target Time	35			Time	41.28
2		Project Variance	=	15.00				Probability	60.19%			Target Probability	97%
3													
4	Activity	Description	AverageTime	EST	EFT	LST	LFT	Total Slack	Critical	Critical Activity Variance			
5	A	Plan Topic	3.00	0.00	3.00	7.00	10.00	7.00	NO				
6	B	Obtain Speakers List Meeting	4.00	0.00	4.00	0.00	4.00	0.00	YES	0.44			
7	C	Locations	6.00	4.00	10.00	4.00	10.00	0.00	YES	1.00			
8	D	Select Location	5.00	10.00	15.00	10.00	15.00	0.00	YES	1.78			
9	E	Finalize Speakers Schedule	2.00	3.00	5.00	13.00	15.00	10.00	NO				
10	F	Prepare and mail Seminar Brochures	9.00	15.00	24.00	15.00	24.00	0.00	YES	9.00			
11	G	THE NEW ACTIVITY	10.00	24.00	34.00	24.00	34.00	0.00	YES	2.78			
12													

Figure A6: Updated CPM Table after addition of activity G

maximum number of activities in a project network that can be solved in a spreadsheet using this approach and the layout presented in this paper is the maximum number of columns possible in the predecessor matrix. Since the predecessor matrix can start only at column I, (the previous columns are needed for other information on the input data) the upper limit on the number of activities possible in modern spreadsheets is 248 (column IV - column I).