



## Software Estimation in PERT Networks

**D.Anuradha**

*Department of Mathematics,  
School of Advanced Sciences  
VIT University, Vellore-14,  
Tamil Nadu, India*

**S.Bhavani**

*School of Information Technology  
and Engineering  
VIT University, Vellore-14,  
Tamil Nadu, India*

---

**Abstract -** *The paper discusses the project management experience of using metrics to measure the growth of software system. A program is developed using Java programming for finding the project completion duration, variance and range estimate to Program Evaluation and Review Technique (PERT) network. An effort is made to measure the size of the developed system. A computer system installation project has been solved illustrating the validity of the proposed approach. This approach can be served as an essential tool for the software project managers when they are dealing different types of project networks.*

**Keywords:** *PERT, Source Lines of Code (SLOC), Function Points (FP).*

---

### I. INTRODUCTION

Project management is a complex attempt and growth of a metric for project management strength. PERT is a project management tool utilized to plan and coordinate activities within a project. The improvement and substantiation of methods used in software project estimation is calculated quantitatively with software metrics. The software metric is a measurement that commonly uses numerical evaluations to enumerate some qualities of a software entity. Metrics are classified by the duration and the software complexity. When the metrics are applied in software lifetime it is based on the duration and on the other hand, while dealing with computations it is based on the software complexity. The most popular metrics are counts of the source lines of code and function points. The familiar patterns have normally been weakly connected with design methodologies like data flow diagram, program flow chart, and PERT chart. An argument of metrics programs and metrics cannot be complete without a discussion of software size estimation which is an inherent characteristic of a piece of software. William and C.Terry [11] surveyed metrics and models of software reuse. Rachel Harrison [7] investigated a set of metrics for object-oriented design, called the MOOD metrics. LC Briand [5] rendered a framework for the comparison, estimation, and description of coupling measures in object-oriented systems. Khaled El Emam et al. [4] analyzed the confounding consequence of class size in the validation of object-oriented metrics. Bansiya [3] described an improved hierarchical design for the appraisal of high level design quality attributes in object oriented designs. J Marasovic and T Marasovic [2] framed a novel thought for useful E-learning support using critical path method and PERT project planning methods. Sastry et al. [8] presented various existing and new software metrics and then structured them into various stages of Software Development Life Cycle. Yongchang Ren [12] developed the computer program algorithm for the software project schedule planning using the critical path method. Gurdev Singh et al. [1] studied different types of software metrics which are used during the software development.

In this paper, we propose a programming approach to solve a PERT network. Also the classic metrics namely, source lines of code and function points are determined for the generated program. The output of this method includes expected duration of project completion time, variance and range estimate. Section 2 introduces the software estimation in project networks and its various metrics. The usage of metrics in PERT networks is shown in section 3. In Section 4 a simple project is chosen to illustrate the application of the proposed algorithm. The results and the graphical representation of SLOC and FP are discussed in section 5. Finally, the work is concluded in section 6.

### II. SOFTWARE ESTIMATION IN PROJECT NETWORKS

Software project management concentrates on people, product, process and project that begin before any proficient activity is started and proceeds throughout the development of computer software. The software process is designed using the critical path method (CPM) which is a gradual technique that determines critical and non-critical activities with the aim of preventing duration frame works. PERT takes critical path method network and adds distributions to symbolize the activity durations of the project. Software measure is a quantitative sign of dimension, capacity, amount, extent, or size of some quality of a product or process whereas metric is a quantitative assessment of degree to which a component, system or process owns a given quality. The software metrics are classified as product, process and project metrics. Among these metrics, product metrics has the feature size [9]. Size measures include source lines of code and function points. SLOC is one of the most former and simpler metrics for computing the size of computer program. The size of a large software product can be calculated in more beneficial way through a huge unit called module. A module can be determined as segment of code which may be compiled separately [1].

### III. CO-USE OF METRICS IN PERT NETWORKS

PERT approach employs the beta distribution as the distribution of task duration and computes the mean, variance and range of estimate of activity duration based on the Pessimistic (P), Optimistic (O) and Most-likely (M) duration estimates. A program is generated to obtain the expected duration, variance and range of estimate for all activities in a given PERT network by using functions. The program also identifies all the possible paths and calculates its corresponding total path duration. The path with maximum duration is called as the Critical Path (CP) of the PERT network and its duration is the total project completion duration. An exact estimate of software size is a crucial factor in the computation of estimated project schedules. The size of the software in terms of source line of code and function points are estimated for the given PERT network.

The proposed algorithm for PERT network proceeds as follows:

- Step1 : Start.
- Step2 : Initialize all variables for  $n$  nodes.
- Step3 : Initialize all values for nodes and activities: Values include optimistic, most-likely and pessimistic.
- Step4 : Calculate and print expected duration, variance and range for all activities.
- Step5 : Find root node in the structure.
- Step6 : Print all possible paths in network using recursive function.
- Step7 : Calculate sum of expected duration in all paths.
- Step8 : Return expected duration of the given activity.
- Step9 : Determine critical path.
- Step10 : Terminate overall activity by getting user input.
- Step11 : Print the stored paths and critical path provided by critical function.
- Step12 : End.

Fig.1 portrays the flowchart and description of the proposed algorithm.

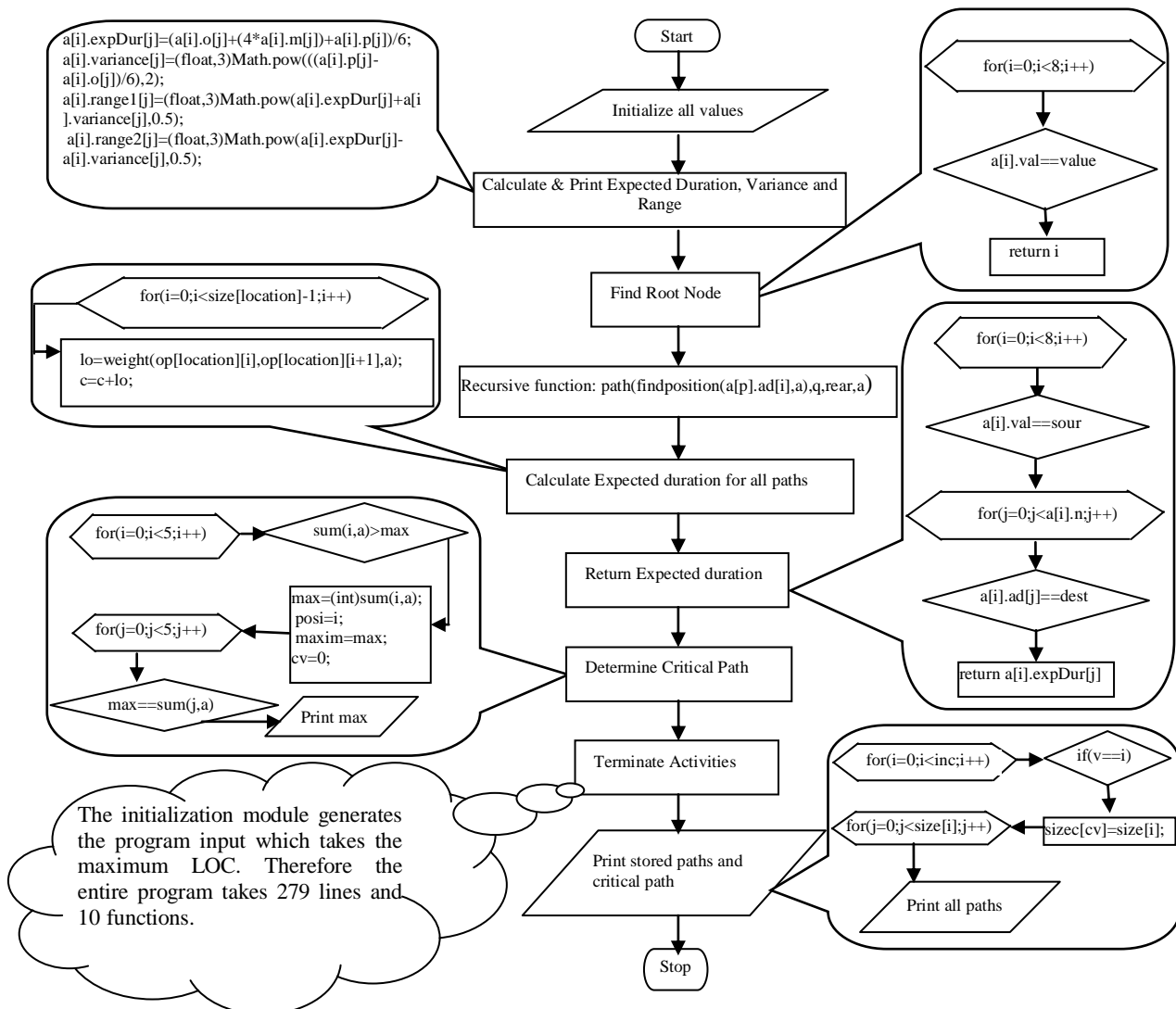


Fig.1 Flowchart and description

IV. ILLUSTRATION

A simple project is chosen to illustrate the application of the proposed algorithm and its metrics. Table 1 lists the various activities involved in the computer system installation. The installation begins with the selection of computer model followed by the design of input/output system and monitoring system. Assembling computer hardware and development of main programs takes very long time which starts once the system is designed. The activities involved depend on the previous activities for its commencement. The input/output routines are developed after the computer hardware assembling. A database has to be created as an extension to the main program. Finally the system can be installed and tested with the implementation.

TABLE 1  
COMPUTER SYSTEM INSTALLATION ACTIVITIES

Task	Description	Precedence task	Estimated Duration (Days)		
			O	M	P
A	Select the computer model	--	4	6	8
B	Design input/output system	A	5	7	15
C	Design monitoring system	A	4	8	12
D	Assemble computer hardware	B	15	20	25
E	Develop the main programmes	B	10	18	26
F	Develop input/output routines	C	8	9	16
G	Create database	E	4	8	12
H	Install the system	D,F	1	2	3
I	Test and implement	G,H	6	7	8

V. RESULTS AND DISCUSSION

Using the program generated for the software algorithm, we obtain the following expected duration, variance, range of the estimate of each activity in fig.2.

```

C:\WINDOWS\system32\cmd.exe - java Main1
C:\Program Files\Java\jdk1.6.0\bin>javac Main1.java
C:\Program Files\Java\jdk1.6.0\bin>java Main1
for 1->2 => Expdur:6.0 variance:0.4444448 range: 5.3333335 to 6.6666665
for 2->3 => Expdur:8.0 variance:2.7777777 range: 6.3333335 to 9.6666667
for 2->4 => Expdur:8.0 variance:1.7777779 range: 6.6666665 to 9.3333333
for 3->5 => Expdur:18.0 variance:7.1111116 range: 15.333333 to 20.666666
for 3->6 => Expdur:20.0 variance:2.7777777 range: 18.333334 to 21.666666
for 4->6 => Expdur:10.0 variance:1.7777779 range: 8.6666667 to 11.3333333
for 5->7 => Expdur:8.0 variance:1.7777779 range: 6.6666665 to 9.3333333
for 6->7 => Expdur:2.0 variance:0.1111112 range: 1.6666666 to 2.3333333
for 7->8 => Expdur:7.0 variance:0.1111112 range: 6.6666665 to 7.3333335
1-> 2-> 3-> 5-> 7-> 8->
1-> 2-> 3-> 6-> 7-> 8->
1-> 2-> 4-> 6-> 7-> 8->
Maximum is.....47
1->2->3->5->7->8->
    
```

Fig.2 Program Ouput

From fig.2 we observe that the critical path of given project is A → B → E → G → I . The expected duration of the critical path is 47 days, variance of the path is 12.222 and the Standard Deviation (SD) of the path is 3.496 (square root of variance). The network is drawn for the given scenario and the critical path is highlighted with dark lines in fig.3.

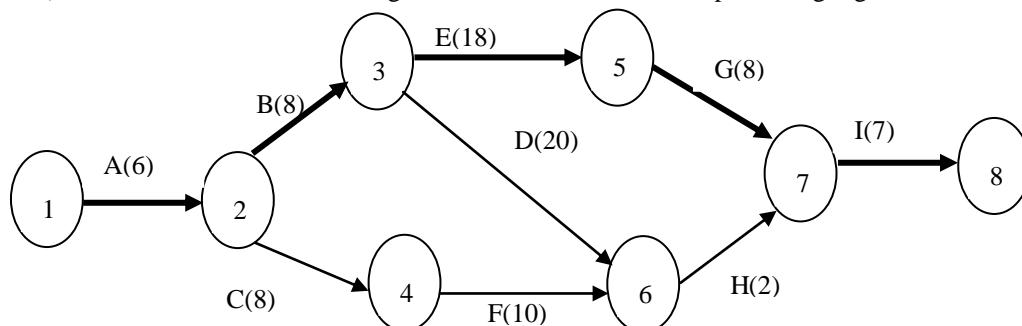


Fig. 3 PERT Network

#### A. Statistical Estimates of Total Time

Variance is mainly helpful for comparison to the expected values. However, the standard deviation just as simply, anticipates that we must recognize whether it is a one, two or three sigma limit deviation. One sigma limit represents 68 percent chance of completion within one standard deviation with a range of 43.504 – 50.496. Two sigma limits symbolizes 95 percent within two standard deviations with a range of 40.008 – 53.992 and three sigma limits represents 99 percent within three standard deviations with a range of 36.512 – 57.488.

#### B. Graphical Representation

Fig.4 clearly depicts the count of source line of code, coded to each block of the program. The program of the proposed algorithm has several functions to perform various tasks of PERT network. Appropriate header files and Class are included. Initialization of the necessary variables with its values is done. The main function of the program calls various sub functions to perform calculations.

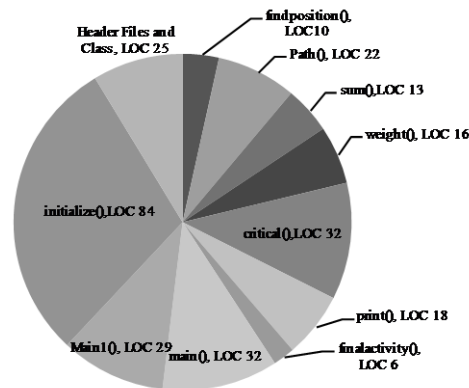


Fig. 4 Graphical representation

Suitable sub functions are defined in such a way that it calculates the expected duration, all possible paths and critical path of the PERT network. The program comprises of nine sub functions, namely, findposition(), path(), sum(), weight(), critical(), print(), finalactivity(), main1(), initialize() and a main function, namely, main() with 279 source line of code.

## VI. CONCLUSION

This paper, investigated the use of classic metrics in PERT network. By using classic metrics the proposed program of the given network can be estimated for its size. The validity of the proposed algorithm is examined with the illustration. Thus this approach paves an easy way for the software managers to complete the project within the estimated duration, range of the estimate and suggest ways in which the metrics might be used.

## REFERENCES

- [1] Gurdev Singh, Dilbag Singh and Vikram Singh, A study of software metrics, *International Journal of Computational Engineering and Management*, Vol. 11, pp.22-27, 2011.
- [2] Jadranka Marasovic and Tea Marasovic, CPM/PERT project planning methods as E-learning optional support, *International conference on SoftCOM*, pp. 352-356, 2006.
- [3] Jagdish Bansiya and Carl G. Davis, A hierarchical model for object-oriented design quality assessment, *IEEE Trans. Software Eng.*, Vol. 28, pp.4-15, 2002.
- [4] Khaled El Emam, Saida Benlarbi, Nishith Goel, and Shesh N. Rai, The confounding effect of class size on the validity of object-oriented metrics, *IEEE Trans. Software Eng.*, Vol. 27, pp. 630-650, 2001.
- [5] Lionel C. Briand, John W. Daly, and Jurgen K. Wust, A unified framework for coupling measurement in object-oriented systems, *IEEE Trans. Software Eng.*, Vol. 25, pp.91- 121, 1999.
- [6] Norman E. Fenton, Shari Lawrence Pfleeger, *Software Metrics A rigorous & Practical Approach*, PWS Publishing Company, Thomson Learning Inc., 2004.
- [7] Rachel Harrison, An evaluation of the mood set of object-oriented software metrics, *IEEE Transactions on Software Engineering*, Vol. 24, pp. 491-496, 1998.
- [8] Sastry, B.R., M.V. Vijaya Saradhi, impact of software metrics on object-oriented software development life cycle, *International Journal of Engineering Science and Technology*, Vol. 2, 67-76, 2010.
- [9] Stephen H. Kan, *Metrics and Models in Software Quality Engineering*, Pearson Education, Inc. 2003.
- [10] Walker Royce, *Software Project Management A unified Framework*, Pearson Education, Sixth Edition, 2003.
- [11] William Frakes and C. Terry, Software reuse: metrics and models, *ACM computing surveys*, Vol. 28, pp.415-435, 1996.
- [12] Yongchang Ren, Study of using critical path method to formulate the algorithm of software project schedule planning, *International Conference on Information Management, Innovation Management and Industrial Engineering*, 2010.