A Comparative Study on Fuzzy Approaches for COCOMO's Effort Estimation

Anupama Kaushik, A. K. Soni, and Rachna Soni

***Abstract*—The effort invested in a software project is one of the most challenging task and most analyzed variables in recent years in the process of project management. Many effort estimation models are developed. Each of these models has their own pros and cons in estimating the development cost and effort. This is because in initial stages project data is often incomplete and unclear. The most widely used model for effort estimation is Constructive Cost Model (COCOMO) but there is a great deal of imprecision present in its input which leads to imprecision in its output thereby resulting in erroneous effort estimation. Fuzzy logic based cost estimation models address the vagueness and imprecision present in these models to make reliable and accurate estimates of effort. The aim of this paper is to analyze the use of fuzzy logic in the COCOMO model and to provide indepth review and comparison of software effort estimation models.**

***Index Terms*—Project management, software effort estimation, fuzzy logic, COCOMO model**

I. Introduction

Software cost estimation is one of the most significant activities in software project management. Accurate cost estimation is important because it can help to classify and prioritize development projects to determine what resources to commit to the project and how well these resources will be used. Bulk of the cost of software development is due to the human effort and most cost estimation methods focus on this aspect and give estimates in terms of person-months which can then be converted into project duration and cost. However, estimates at the starting stages of the project are the most difficult to obtain due to limited details available.

In the last few decades many software cost estimation models have been developed. The algorithmic models use a mathematical formula to predict project cost based on the estimates of project size, the number of software engineers, and other process and product factors [1]. These models can be built by analysing the costs and attributes of completed projects and finding the closest fit formula to actual experience. COCOMO (Constructive Cost Model), is the best known algorithmic cost model published by Barry Boehm in 1981 [2]. It was developed from the analysis of sixty three software projects. Boehm provided three levels of the model called Basic COCOMO, Intermediate COCOMO and Detailed COCOMO. These conventional approaches lacks in terms of effectiveness and robustness in their results.

Non algorithmic models for cost estimation encompass methodologies on fuzzy logic (FL), artificial neural networks (ANN) and evolutionary computation (EC). Fuzzy logic based cost estimation is most appropriate when vague and imprecise information is to be handled.

The rest of the paper is organized as follows: Section 2 gives an introduction on fuzzy logic. Section 3 provides a review on application of fuzzy logic in COCOMO model. Section 4 discusses various criterions for assessment of software cost estimation models. Section 5 summarizes and compares the experimental results by different researchers. Section 6 gives conclusions and future research.

II. Fuzzy Logic

Fuzzy Logic is a methodology to solve problems which are too complex to be understood quantitatively. It is based on fuzzy set theory and introduced in 1965 by Prof. Zadeh in the paper fuzzy sets [3]. It is a theory of classes with unsharp boundaries, and considered as an extension of the classical set theory [4]. The membership $\mu_A(x)$ of an element x of a classical set A, as subset of the universe X, is defined by:

$$\mu_A(x) = \begin{cases} 1\ if\ fx \in A \\ 0\ if\ fx\ \notin A \end{cases}$$

That is, x is a member of set A ($\mu_A(x) = 1$) or not ($\mu_A(x) = 0$). The classical sets where the membership value is either zero or one are referred to as crisp sets.

Fuzzy sets allow partial membership. A fuzzy set A is defined by giving a reference set X, called the universe and a mapping; called the membership function of the fuzzy set A $\mu_A(x)$, for $x \in X$ is interpreted as the degree of membership of x in the fuzzy set A. A membership function is a curve that defines how each point in the input space is mapped to a membership value between 0 and 1. The higher the membership x has in the fuzzy set A, the more true that x is A. The membership functions (MFs) may be triangular, trapezoidal, Gaussian, parabolic etc. In this paper we have discussed three kinds of MFs i.e. triangular (Fig. 1), trapezoidal (Fig. 2), and Gaussian (Fig. 3) [5].

$$\mu_A: X \rightarrow [0,1]$$

The triangular membership function is specified by a triplet (a, b, c) as follows:

$$\text{Triangle } (x: a, b, c) = \begin{cases} 0 & x < 0 \\ \frac{x-a}{b-a} & a \leq x \leq b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x > 0 \end{cases}$$

Manuscript received July 27, 2012; revised September 27, 2012.

A. Kaushik is with the Department of Information Technology, Maharaja Surajmal Institute of Technology, Janakpuri, New Delhi, India (e-mail: thisisanupama@gmail.com).

A. K. Soni is with the Department of Information Technology, Sharda University, Greater Noida, U.P, India (e-mail: ak.soni@sharda.ac.in).

R. Soni is with the Department of Computer Science and Applications, D.A.VCollege, Yamunanagar, Haryana, India (e-mail:sonirachna67@gmail.com).

The parameters “a” and “c” locate the feet of the triangle and the parameter “b” locates the peak which is shown in Fig. 1.

The trapezoidal membership function is specified by four parameters (a, b, c, d) as follows:

$$\text{Trapezoidal } (x: a, b, c, d) = \begin{cases} 0 & x < 0 \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & x > d \end{cases}$$

The parameters “a“ and “d” locate the feet of the trapezoid and the parameters “b” and “c” locate the shoulders and shown in Fig. 2.

The Gaussian membership function is specified by two parameters (m,σ) as follows :

Gaussian(x: m, σ) = $\exp\left(-\frac{(x-m)^2}{\sigma^2}\right)$ as shown in Fig. 3.

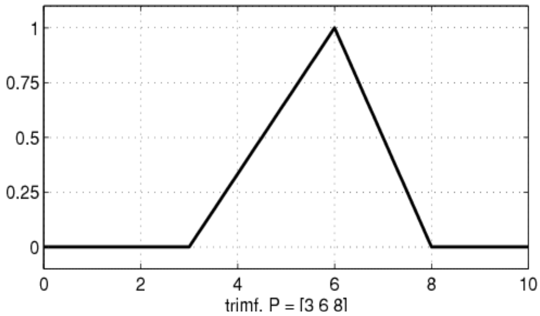

Fig. 1. A triangular MF specified by (3, 6, 8).

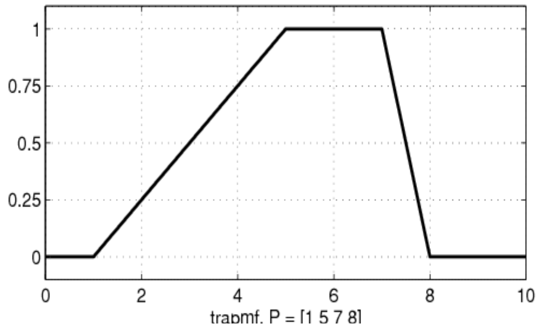

Fig. 2. A trapezoidal MF specified by (1, 5, 7, 8).

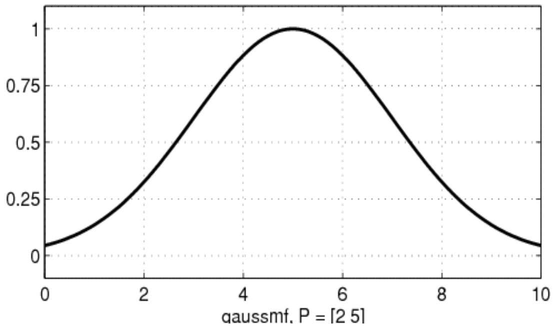

Fig. 3. A gaussian MF specified by (2, 5).

A general fuzzy system includes the following elements [6]:

1) Fuzzification Process: Here the membership functions are applied to the numerical value of input variables, to determine how much the input variables fit the linguistic terms.
2) Knowledge Base: It is a set of expert control rules needed to achieve a goal. The knowledge base is usually expressed as a number of ‘IF-THEN’ rules based on the domain expert’s knowledge.
3) Fuzzy Inference Mechanism: It performs various fuzzy logic operations by using knowledge base to convert fuzzy inputs to fuzzy outputs.
4) Defuzzification Process: Here the conclusion of the fuzzy rule set is translated into a crisp number before results can be used in practice.

III. Fuzzy Approaches in COCOMO

The first realization of the fuzziness in COCOMO was that of Fei and Liu [7] called f-COCOMO. They observed that an accurate estimate of delivered source instruction (KDSI) cannot be done before starting a project; so it is unreasonable to assign a determinate number to it. The reason for fuzziness to be considered in COCOMO is due to vague and imprecise nature of some involved factors, which have important influence upon development cost. They are always depicted by grades with natural language descriptions as very high, high and so on.

Jack Ryder [8] applied fuzzy modelling techniques to COCOMO and Function Points model respectively. This was due to the fact that COCOMO and function point models were similar in the sense that they use several inputs to make an unadjusted prediction, calculate an adjustment factor and then scale the unadjusted prediction by the factor. Jack Ryder provided two alternatives. In the first alternative he inserted a fuzzy expert system to calculate the adjustment factor. In his second alternative he reduced the number of inputs of both the models and inserted the fuzzy expert system at the top layer.

Musflek et. al. [9] fuzzified the basic COCOMO model at two different levels of detail. The first level called F-COCOMO was concerned about representing the size of software project as fuzzy set while the coefficients representing mode remain crisp values. The second level called the augmented f- COCOMO provided a representation of the modes of software development as singleton fuzzy sets. Their approach although, reduced the sensitivity to imprecision in the input data but the discrete number representation of mode does not take care of imprecision.

Idri and Abran [10] provided a two stage model for intermediate COCOMO, called simple F-COCOMO model and augmented F-COCOMO model respectively. They fuzzified only the cost drivers and not the other inputs i.e. mode and size of intermediate COCOMO. The fuzzified cost drivers took care of the very sharp transition between two different intervals defined for a single cost driver.

Ahmed, M.A., Omolade Sailu et. al. [11] presented an adaptive fuzzy logic based framework for software development effort prediction. Their framework used intermediate COCOMO as the base cost model. They integrated the nominal effort and the cost drivers of the COCOMO model and generated the artificial datasets for the experiments. They fuzzified mode of development and size and the corresponding effort was calculated using the COCOMO model equation. They used triangular membership functions and rules were formulated based on the relationship between size and effort. They developed a Fuzzy Inference System (FIS) for each cost driver. The rules were formulated using the cost drivers in the antecedent and their effects on effort in the consequent. The defuzzified values from each of the FIS were aggregated as effort adjustment factor (EAF) to adjust the predicted output from the trained FIS for nominal effort. For adaptive training approach, they used the Mamdani max-min fuzzy reasoning and fuzzy perceptron structure introduced by Nauck et al. [12].

Prasad Reddy P.V.G.D et. al. [13] provided a Fuzzy based approach for Predicting Software Development Effort. They used intermediate COCOMO model for developing the FIS. They fuzzified the size and mode and estimated the fuzzy nominal effort. This effort multiplied by the EAF gave the estimated effort. The membership functions used by them were triangular and GBell functions.

Ch. Satyananda Reddy et. al. [1] provided an Improved Approach for COCOMO's effort estimation using Gaussian Membership function. They evaluated the COCOMO model using the following equation:

$$\text{Effort} = A \times [size]^{1.01+\sum_{i=1}^{5} SF_i} \times \prod_{i=1}^{17} EM_i \qquad (1)$$

where A is a multiplicative constant, SF is the set of 5 scale factors and EM is the set of 17 effort multipliers [14].They calculated the effort by fuzzifying the cost drivers using Gaussian Membership function.

Harish Kumar Verma and Vishal Sharma [15] presented Handling Imprecision in Inputs using fuzzy logic to predict effort in Software Development. They used Gaussian Membership function to fuzzify the software development mode and size of a project. Separate independent FIS is used for every cost driver. The defuzzified value for each of the effort multiplier is obtained from individual FIS's and comprehensive EAF is obtained after multiplying them together. Comprehensive software effort is obtained by multiplication of crisp effort from the basic part and the crisp EAF from the cost driver part.

IV. Assessment Criteria

The evaluation of software effort estimation models consist in comparing the accuracy of the estimated effort with the actual effort. There are many evaluation criteria's for software effort estimation in the literature. A few of them are discussed below [16]:

1) Variance Accounted For (VAF)

$$\text{VAF}(\%) = \left(1 - \frac{var\,(E-\widehat{E})}{var\ E}\right) \times 100$$

2) Mean Absolute Relative Error (MARE)

$$\text{MARE}\ (\%) = \frac{\sum f\,(R_E)}{\sum f} \times 100$$

3) Variance Absolute Relative Error (VARE)

$$\text{VARE}\ (\%) = \frac{\sum f\,(R_E - mean\ R_E)^2}{\sum f} \times 100$$

4) Prediction (n):

Prediction at level n is defined as the % of projects that have absolute relative error less than n.

5) Balance Relative Error (BRE)

$$\text{BRE} = \frac{|E-\hat{E}|}{\min\,(E,\widehat{E})}$$

6) Magnitude of Relative Error (MRE)

$$\text{MRE} = \frac{|E-\hat{E}|}{E} \times 100$$

7) Mean Magnitude of Relative Error (MMRE)

$$\text{MMRE} = \frac{1}{N}\sum_i \frac{E_i - \widehat{E_i}}{E_i}$$

where E = Estimated Effort, $\hat{E}$ = Actual Effort

$$R_E = \text{Absolute Relative Error}$$

and $R_E = \frac{|E-\hat{E}|}{|E|}$

A model which gives higher VAF is better than that which gives lower VAF. A model which gives higher Prediction (n) is better than which gives lower prediction (n). A model which gives lower MARE is better than which gives higher MARE. A model which gives lower VARE is better than which gives higher VARE. A model which gives lower BRE is better than which gives higher BRE. A model which gives lower MRE and MMRE is better than which gives higher values.

V. Results

Experiments were done by taking original data from COCOMO dataset [17]. The software development efforts obtained when using COCOMO and other fuzzy membership functions were observed.

TABLE I: Effort Assessment Using COCOMO, Triangular And GBELL Membership Function

Model	VAF %	MARE%	VARE%	Mean BRE	Pred (25)%
COCOMO	87.16	21.41	5.48	0.25	72
Triangular MF	95.83	18.63	4.35	0.23	68
GBell MF	92.25	20.35	4.24	0.26	62

TABLE II: Effort Assessment Using COCOMO And Increasing Membership Functions

Model	MMRE
COCOMO	38.83
FIS with 3MFs	64.26
FIS with 5MFs	41.06
FIS with 7MFs	38.38
FIS with 11MFs	34.15

TABLE III: Effort Assessement Using COCOMO, Trapezoidal and Gaussian Membership Functions

Model	MRE	MMRE
COCOMO	25.20	32.65
Trapezoidal MF	20.51	22.09
Gaussian MF	15.82	17.02

Referring Table I [13] we found that triangular membership function model is giving better results for maximum criteria in effort assessment when compared with COCOMO and GBell MF models.

In Table II [15] we found that the effort estimation improved with increased membership functions. The MFs used here are Gaussian MF for size, mode and effort. FIS for cost drivers are developed using triangle MF and trapezoidal MF.

Referring Table III [1] we found that Gaussian membership function model in effort estimation is providing better results than COCOMO and Trapezoidal MF models.

Accurate cost estimation is very important as they help to classify and prioritize development projects with respect to an overall business plan. The most important is that customers expect actual development costs to be in line with the estimated costs. It is observed that there can be many fuzzy estimation models which are providing alternatives to estimate software development effort by improving estimation accuracies.

VI. Conclusions

This paper has presented an overview of fuzzy approaches in COCOMO's effort estimation. In all these approaches, fuzzy logic techniques are applied on COCOMO's equation for estimating the software effort. Some researchers have fuzzified size only, some have fuzzified size and mode, and some have fuzzified all the three components i.e. size, mode and cost driver part. They have seen the effect of fuzzifying these components on the software effort.

Future work includes estimation of software effort with different forms of membership functions and comparison of results with the earlier approaches. Newer techniques like Type-2 fuzzy can also be applied for more accurate predictions of software effort.

Acknowledgment

The authors would like to thank the anonymous reviewers for their careful reading of this paper and for their helpful comments.

References

[1] C. S. Reddy and K. Raju, "An improved fuzzy approach for COCOMO's effort estimation using gaussian membership function," *Journal of software*, vol. 4, no. 5, July 2009.

[2] B. W. Boehm, *Software Engineering Economics*, Prentice Hall, Englewood Cliffs, NJ, 1981.

[3] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

[4] M. W. Nisar, Y.-J. Wang, and M. Elahi, "Software development effort estimation using fuzzy logic – A survey," *Fifth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 18-20, pp. 421-427, October 2008.

[5] Matlabs description of fuzzy Inference Systems. [Online]. Available: http://www.mathworks.com/help/toolbox/fuzzy

[6] S. N. Sivanandam and S. N. Deepa, *Principles of Soft Computing*, Wiley India, 2007, vol. 9, pp. 369-388.

[7] Z. Fei and X. Liu, "f-COCOMO-fuzzy constructive cost model in software engineering," *Proceedings of IEEE International Conference on Fuzzy System*, pp. 331-337.

[8] J. Ryder, "Fuzzy modeling of software effort prediction," *Proceedings of IEEE Information Technology Conference*, Syracuse, NY, 1998.

[9] P. Musflek, W. Pedrycz, G. Succi, and M. Reformat, "Software cost estimation with fuzzy models," *Applied Computing Review*, vol. 8, no. 2, pp. 24-29, 2000.

[10] A. Idri and A. Abran, "COCOMO cost model using fuzzy logic," *7th International Conference on Fuzzy Theory and Technology*, Atlantic City, New Jersey, March 2000.

[11] M. A. Ahmed, M. O. Saliu, and J. AlGhamdi, "Adaptive fuzzy logic based framework for software development effort prediction," *Information and Software Technology, ELSEVIER*, pp. 31-48, 2005.

[12] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*, Wiley, Chichester, 1997.

[13] P. V. Reddy, K. R. Sudha, P. R. Sree, and S. N. Ramesh, "Fuzzy based approach for predicting software development effort," *International Journal of Software Engineering (IJSE)*, vol. 1, no. 1.

[14] B. W. Boehm, *et al.*, "Cost models for future software life cycle processes: COCOMO 2.0," *Annals of Software Engineering on Software Process and Product Measurement, Amsterdam,* 1995.

[15] H. K. Verma and V. Sharma, "Handling imprecision in inputs using fuzzy logic to predict effort in software developmetn," *International Advance Computing Conference IEEE,* 2010.

[16] H. Mittal and P. Bhatia, "A comparative of conventional effort estimation and fuzzy effort estimation based on triangular fuzzy numbers," *International Journal of Computer Science and Security*, vol. 1, no. 4.

[17] M. Korte and D. Port, "Confidence in software cost estimation results based on MMRE and PRED," *PROMISE'08,* Leipzig, Germany, pp. 12-13, May 2008.

Anupama Kaushik received her B.E (Computer Science) from Bharathiyar University, M.Tech (Information Technology) from Tezpur University and doing PhD from Sharda University. She joined Maharaja Surajmal Institute of Technology as an Assistant Professor in 2004. Her research area includes Software Engineering, Object Oriented Software Engineering and Soft Computing.

Dr. A. K Soni has done his Ph.D. & M.S.(Computer Science) both from Bowling Green State University in Ohio, USA. He is the Professor and Head, Department of Information Technology, Sharda University. He has more than seventeen years of teaching experience. He has published many papers in national and international journals. His research area includes Software engineering, Datamining, Database management systems and object oriented systems.

Dr. Rachna Soni did her M. Phil from IIT, Roorkee and Ph.D. from Kururukshetra University, Kurukshetra. She is the Associate Professor and Head, Dept. of Computer Science & Applications. She has more than eighteen years of teaching experience in the institution of repute. She has published several papers in International/national journals /conferences. Her area of interest includes Software Risk Management, Project Management, Requirement Engineering, Simulation and Component based Software Engineering.