Analyzing Effort Estimation in Multistage based FL-COCOMO II Framework using various Fuzzy Membership Functions

Kunal Gaurav
Dept. of Computer Science & Engineering,
BIT Mesra, Ranchi, India

Pooja Jha
Dept. of Computer Science & Engineering,
BIT Mesra, Ranchi, India

K.S.Patnaik, PhD.
Dept. of Computer Science & Engineering,
BIT Mesra, Ranchi, ndia

ABSTRACT
Software development has always been characterized by some metrics. One of the greatest challenges for software developers lies in predicting the development effort for a software system which is based on developer abilities, size, complexity and other metrics. Several algorithmic cost estimation models such as Boehm's COCOMO, Albrecht's' Function Point Analysis, Putnam's SLIM, ESTIMACS etc. are available but every model has its own pros and cons in estimating development cost and effort. Most common reason being project data which is available in the initial stages of project is often incomplete, inconsistent, uncertain and unclear. In this paper, soft computing based technique is explored to overcome the problems of uncertainty and imprecision resulting in improved process of software development effort estimation. The paper considers a multistage software estimation approach using six key cost drivers in COCOMO II model. The selected cost drivers are the inputs to fuzzy expert systems and the effort multiplier is obtained as output.

Keywords
Software Engineering, software cost estimation models, COCOMO II, soft computing, effort estimation.

1. INTRODUCTION
Accuracy in software estimation is among the greatest challenges for software developers. Software effort estimation deals with the prediction of the probable amount of time and cost required to complete the specific development task. Software metric and especially software estimation is based on measuring of software attributes which are typically related to the product, the process and the resources of software development [3]. Many quantitative models of software cost estimation have been developed. Most of these models are based on the size measure, such as Line of Code (LOC) and Function Point (FP), obtained from size estimation [2]. It is obvious that the accuracy of size estimation directly has impact on the accuracy of cost estimation. Based on this context, fuzzy logic can be an alternative choice to estimate task effort in software development.

This paper extends the Constructive Cost Model (COCOMO II) by incorporating the concept of fuzziness. Here, the key cost drivers are identified and fuzzy logic approach is used. The paper is organized as follows: section II briefly outlines the cost estimation models and COCOMO II model. Section III discusses implementation of fuzzy logic methodology in COCOMO II model and the detailed fuzzy logic based multistage software estimation framework is given using Gaussian and Bell shaped membership function. Section IV concludes with evaluation of numerical simulation result.

2. SOFTWARE COST ESTIMATION MODELS
Software developers estimates time of software tasks by comparing similar tasks that have already been developed. Although, this task has an uncertain nature, due to its dependency on several and usually not clear factors and which is hard to be modeled mathematically [2]. For considerable financial and strategic planning, the reliable and accurate cost estimation is an ongoing challenge. Software cost estimation techniques can be classified as algorithmic and non-algorithmic models.

2.1 Algorithmic Model
Algorithmic models are based on mathematical equation which is based upon the statistical analysis of data gathered from previously developed projects, e.g. Software Life Cycle Management (SLIM) and COCOMO and Albrecht's Function Point. The mathematical equations involved in these models use inputs such as Source Lines of Code (SLOC), number of functions to perform number of user screen, interfaces, complexity, and other cost drivers such as language, design methodology, skill-levels, risk assessments, etc. at a time when uncertainty is mostly present in the software [1]. Understanding and calculation of these models are difficult due to inherent complex relationships between the related attributes, and inability to handle categorical data as well as lack of reasoning capabilities [2]. The inability of algorithmic model to handle categorical data (which are specified by a range of values) and most importantly lack of reasoning capabilities contributed to the number of studies exploring non-algorithmic methods [6].

2.2 Non-algorithmic Model:
Non-algorithmic models came in 1990's and widely used in software cost estimation. Software researchers looked for new approaches which were based on soft computing approach such as artificial neural networks, fuzzy logic, and genetic algorithms [2]. Fuzzy Logic offers a powerful linguistic representation that able to represent imprecision in the model inputs and outputs, while providing a more knowledge base approach to establish an effective model. Research shows that using Fuzzy Logic can result in good performance in terms of reducing imprecision of inputs and outputs parameters.

2.2.1 COCOMO II
The COCOMO I model is a regression-based stable software cost estimation model developed by Boehm in 1981. One of the problems with the use of COCOMO I today is that it does not match the development environment of the late 1990's. Therefore, in 1997, Boehm developed the COCOMO II which solved most of the COCOMO I problems [4] [5]. The

estimated effort is given by equation 1 and equation 2 refers to the scaling exponent used in COCOMO II. The COCOMO II includes several software attributes such as: 17 Effort Multipliers (EMs), 5 Scale Factors (SFs), Software Size (SS), and Effort estimation that are used in the Post Architecture Model of the COCOMO II. The description of the 17 EMs and 5 SFs based upon their numerical values and productivity ranges are shown in Table 1 and Table 2 [6].

$$\mathrm{PM} = \mathrm{A} \times [\mathrm{Size}]^{E} \times \prod_{i=1}^{17} EM_i \qquad (1)$$

$$E = B + 0.01 \times \sum_{j=1}^{5} SF_j \qquad (2)$$

where, PM is the estimated effort expressed in person-months; A=2.94 is effort coefficient that can be calibrated; E is scaling exponent consisting of 5 scale factors, EM represents 17 effort multipliers; B=0.91 is constant that can be calibrated; SF represents scale factors.

The development time is calibrated as given in equation 3.

$$TDEV = [C \times (PM_{NS})^{F}] \times \frac{SCED\%}{100} \qquad (3)$$

$$F = (D + 0.2 \times [E - B])$$

where, TDEV is time to develop; C is coefficient that can he calibrated; PM_{NS} is the estimated effort expressed in person-months with nominal schedule; SCED is one of the effort multipliers, which is required schedule compression; D is scaling base-exponent that can he calibrated; B is scaling base-exponent for the effort equation; E is scaling exponent consisting of 5 scale factors [7]. Vagueness and uncertainty of the software attributes has impact on the software estimates. Thus, accurate software attributes resulted in producing accurate software estimates as desired by software project managers and organizations.

Table 1: The range of COCOMO II EMs

Effort Multiplier	Range
Required s/w reliability (RELY)	0.75 - 1.39
Database size (DATA)	0.93 - 1.19
Product complexity (CPLX)	0.75 - 1.66
Developed for reusability (RUSE)	0.91 - 1.49
Documentation match to life-cycle need (DOCU)	0.89 - 1.13
Execution time constraint (TIME)	1.00 - 1.67
Main Storage constraint (STOR)	1.00 - 1.57
Platform volatility (PVOL)	0.87 - 1.30
Analyst capability (ACAP)	1.50 - 0.67
Programmer Capability (PCAP)	1.37 - 0.74
Personnel Continuity (PCON)	1.24 - 0.84
Application experience (APEX)	1.22 - 0.81
Platform experience (PEXP)	1.25 - 0.81
Language and tool experience (LTEX)	1.22 - 0.84
Use of software tools (TOOL)	1.24 - 0.72
Multi site development (SITE)	1.25 - 0.78
Required development schedule (SCED)	1.29 - 1.00

Table 2: The Range of COCOMO II Scale Factors.

Precedentness	6.20 - 1.24
Development Flexibility	5.07 - 1.01
Architecture/Risk resolution	7.07 - 1.41
Team cohesion	5.48 - 1.10
Process maturity	7.80 - 1.56

Above scale factors ranges from very low to very high. Extra high value of the scale factors is 0.

3. Multistage framework of software estimation using FL-COCOMO II

According to Bohem, each Post-Architecture cost driver in COCOMO II model is measured using a rating scale of 6 linguistic values, such as “Very Low”, “Low”, “Nominal”, “High”, “Very High, and “Extra High. The corresponding linguistic values uses the conventional quantification approach when it is assigned, and it is represented by crisp intervals [7][9]. For example, the Execution Time Constraint (TIME) cost driver [6], is measured in percentage use of available execution time. Table 3 shows the linguistic value assignment for TIME [6].

Table 3: TIME Cost Driver Description

TIME (Execution Time Constraint)						
Descriptor	-	-	≤50%	70%	85%	95%
Rating	Very Low	Low	Nominal	High	Very High	Extra High
Effort multipliers	N/A	N/A	1.00	1.11	1.31	1.67

In the Fuzzy Logic-COCOMO II (FL-COCOMO II), fuzzy sets in EMs, SFs, and software size attributes can be specified by distribution of their possible values instead of using fixed values. This reduces the traditional problem of software effort estimation dependency on single value of size, cost driver and scale factors [1]. According to [7], all the effort multipliers are not equally important, hence only 6 key cost drivers among 17 cost drivers is considered here. The six key cost drivers are categorized under 2 factor: RELY (Required s/w reliability), CPLX (Product complexity) and TIME (Execution time constraint) together form Product Factor; ACAP(Analyst Capability), PCAP (Programmer Capability) and PCON (Personnel Continuity) together form Personnel Factor [7]. The CPLX, TIME and RELY are the inputs to Fuzzy Expert System (FES 1) and ACAP, PCAP and PCON are inputs to Fuzzy Expert System (FES 2). The framework deals with fuzzy logic modeling of the key cost drivers. The Product factor and Personnel Factor are inputs to the respective fuzzy expert systems and obtained output is the input to another fuzzy expert system (FES 3) which gives the final output as adjusted estimated total effort multiplier [7].

The framework which is used for the estimation is shown in [7] figure 1. The two membership functions used are GaussianMF and GBellMF.

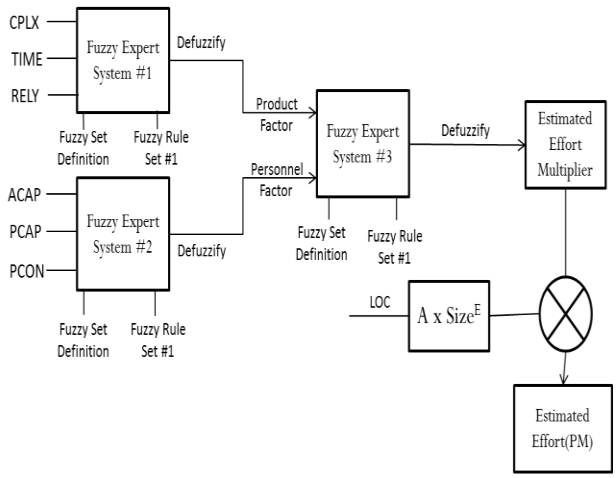

Fig.1. Multistage Software Estimation Framework

The Gaussian membership function is specified by two parameters {m, σ } as given in equation follows:

$$gaussian(x : m, \sigma) = \exp\left(-\frac{(x-m)^2}{\sigma^2}\right) \qquad (5)$$

A GBell membership function is specified by three parameters {a, b, c} as follows:

$$bell(x : a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}} \qquad (6)$$

Figure 2 and figure 3 shows the CPLX cost drivers which is defined in linguistic variables as Low, Medium and High in Gaussian and GBell membership functions.

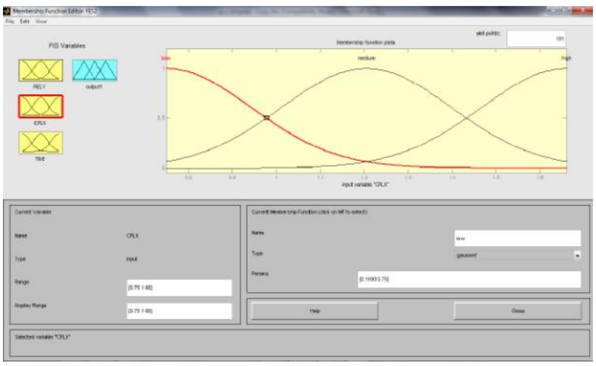

Fig.2. Gaussian Membership Function for CPLX cost driver

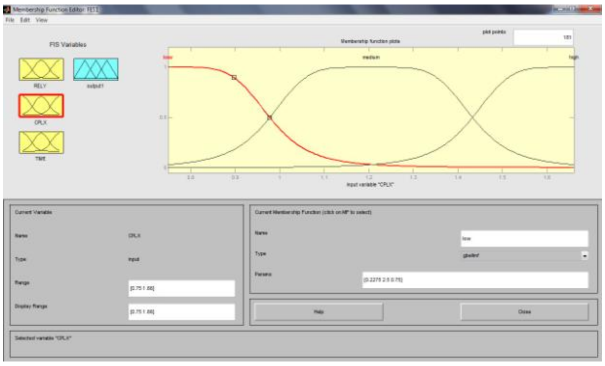

Fig.3. GBell Membership Function for CPLX cost driver

In FES 1 and FES 2 have three inputs with three linguistic variables, such as Low, Medium and High. 27 fuzzy expert rules are created on considering all possible relation among the inputs as shown in following figure.

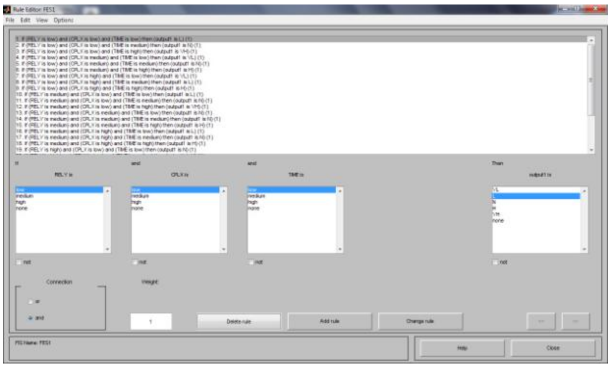

Fig.4 Rules in FES 1

Some fuzzy rules of FES 1:
If (RELY is low) and (CPLX is low) and (TIME is low) then (output1 is low)
If (RELY is low) and (CPLX is low) and (TIME is medium) then (output1 is nominal)
If (RELY is low) and (CPLX is low) and (TIME is high) then (output1 is very high)
…

FES 3 has two fuzzy inputs, Product factor which is output of FES 1 and Personnel factor which is output of FES 2. Each of them is associated with five linguistic variables, such as Very Low, Low, Medium, High and Very High. 25 FES rules is created on considering relations among two inputs as shown in Figure 5.

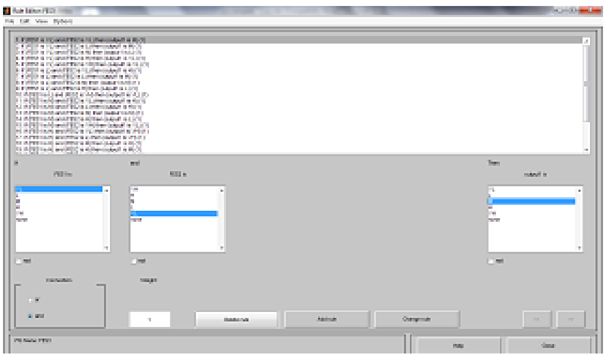

Fig 5 : Rules in FES 3

Some fuzzy rules of FES 3:
If (FES1 is very low) and (FES2 is very low) then (output is medium)
If (FES1 is very low) and (FES2 is medium) then (output is low)
If (FES1 is very low) and (FES2 is high) then (output is very low)
…

4. RESULTS AND DISCUSSIONS

FL-COCOMO II was evaluated using public domain COCOMO II data set. This data set is in public domain and available in http://promisedata.org.

4.1 EVALUATION

The Mean Magnitude of Relative Error (MMRE) and the probability of a project having a relative error of less than or equal to L (PRED(L)) are used in evaluation process.
The Magnitude of Relative Error (MRE) is defined in equation (7) as follows:

$$MRE_i = \frac{|\text{Actual Effort}_i - \text{Predicted Effort}_i|}{\text{Actual Effort}_i} \quad \textbf{(7)}$$

The MRE value was calculated for each observation i that effort is estimated at that observation. The aggregation of MRE over multiple observations (N) can be achieved through the Mean MRE (MMRE) in equation (8) as follows:

$$MMRE = \frac{1}{N}\sum_{i}^{N} MRE_i \quad (8)$$

Prediction at level L is given by PRED(L) = k / N, where k is number of observation where MRE is less than or equal to L. N gives the number of observations. In this paper, PRED (25) is considered. Table 4 shows comparison of results among various models in terms of MMRE and PRED (25). Figure 6 shows graphical representation for the same.

Table 4: Comparison between obtained results from COCOMO II and FL-COCOMO II in terms of MMRE and PRED(25)

MODEL	EVALUATION	
	MMRE	**PRED(25)**
COCOMO II	0.539	0.57
FL-COCOMO II (using Gaussian MF)	0.5804	0.429
FL-COCOMO II (using GBell MF)	0.3197	0.57

In Table 4, MMRE is 0.3197 which is lowest when fuzzy using GBell function is used. MMRE in case of fuzzy using Gaussian is 0.5804, whereas in COCOMO II, it is 0.539. The PRED (25) is improved when GBell function is used and it is about 57 % when compared to Gaussian function of 42.9 %. In COCOMO II, it is 57 %.

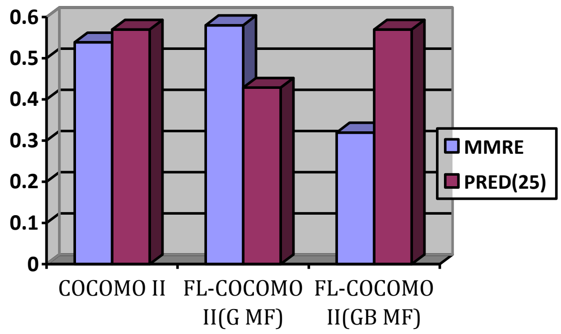

Figure 4: Comparison between obtained results from COCOMO II and FL-COCOMO II in terms of MMRE and PRED(25)

5. CONCLUSION

Software attributes usually have properties of uncertainty and vagueness when they are based on human judgment. A software cost estimation model incorporates fuzzy logic can overcome the uncertainty and vagueness of software attributes. However, determination of the suitable fuzzy rule sets for fuzzy inference system plays an important role in coming up with accurate and reliable software estimates. The objective of this paper was to examine the application of applying fuzzy logic in software cost estimation that can perform more accurate result and overcoming the problems of uncertainty and imprecision. FL-COCOMO II produced better estimation results than the COCOMO. The evaluation criterion MMRE and PRED (25%) shows that estimation could approximate the estimation process in a better way. FL-COCOMO II using GBell membership function resulted in MMRE of 31.9 % and Pred(25) of 57 %.

6. FUTURE WORK

The numerical analysis results show that proposed model can deal with uncertain linguistic variables directly with much less input variables. So the fuzzy system based software cost estimation model can overcome uncertainty exist in software cost drivers. The proposed framework can be used in terms of feasibility and acceptance in the industry. New technologies such as type-2 fuzzy can be incorporated to handle the uncertainty even more closely to make the predictions more accurate.

7. REFERENCES

[1] Sandeep Kad, Vinay Chopra, "Fuzzy Logic Based Framework For Software Development Effort Estimation", An International Journal of Engineering Sciences ISSN: 2229-6913 Issue July 2011, Vol. 1.

[2] Iman Attarzadeh, Siew Hock Ow, "Proposing a New High Performance Model for Software Cost Estimation", 2009 Second International Conference on Computer and Electrical Engineering, page 112-116.

[3] N. E. Fenton, S. L. Pfleeger, Software Metrics, A Rigorous and Practical Approach, 2nd Edition, PWS Publishing Company, Thomson Publishing, Boston, 1997.

[4] Iman Attarzadeh, Siew Hock Ow, "Improving Estimation Accuracy of the COCOMO II using an Adaptive Fuzzy

Logic Model". 2011 IEEE International Conference on fuzzy Systems, June 27-30, 2011, Taipei, Taiwan.

[5] B. W. Boehm, "Cost models for future software life cycle processes: COCOMO 2.0," Annals of Software Engineering Special Volume on Software Process and Product Measurement, Science Publisher, Amsterdam, Netherlands, 1(3) , pp. 45 – 60., 1995.

[6] Software engineering, K K Aggarwal & Yogesh Singh.

[7] Liang tian, Afzel Noore, "Multistage Software Estimation". 2003 IEEE, 232-236.

[8] A. Lotfi Zadeh, "The future of soft computing," The 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, Canada, pp. 217-228, 2001.

[9] A. Idri, A. Abran, and L. Kjiri, "COCOMO cost model using fuzzy logic," in Proceedings of The International Conference on Fuzzy Theory & Technology, 2000.