International Journal of Scientific & Engineering Research, Volume 8, Issue 1, January-2017 1332
ISSN 2229-5518

Analysis of Software Development Effort Estimation Using Fuzzy Logic Functionswith COCOMO II Estimation

Rituraj Jain, Mohd. Khalid Kaleem, Yohannes Bekuma

Abstract-Software effort Estimation is the task of estimation of schedule and the work-effort required to develop and/or maintain a software system. Software effort estimation is a most challenging and onerous task that software developers need to perform. Due to the insufficient information available during the early stage of any software development process, it is hard to estimate effort, cost and schedule correctly. This inaccuracy in estimation leads to delay in product development and delivery which in turn leads to loss. Efforts estimation during the development process are useful for the validation and monitoring of the project's progress. At the time of project closure, these estimates may be useful for project productivity assessment. This paper proposed a fuzzy logic based method applied to different parameters of Constructive Cost Model (COCOMO) II to accurately estimate effort to support the project manager during software development process and overcome the problems of uncertainty and imprecision resulting in improved process of software development effort estimation.

Index Terms- COCOMO II, Estimation, Fuzzy logic, Membership function, Soft Computing, Software Effort Estimation, Gaussian Membership Function

1 Introduction

Most important and crucial activity for software project management isthe estimation of work effort and schedule as well as track the ongoing process of the software development. Estimation schedule and effort is known by software cost estimation process [1]. These estimations play a vital role for initial validation and the tracking the progress of the progress while after the closure of project these estimates will lead as software metrics and used during further projects.

In the Software Technology Conference keynote address, Dr. Patricia Sanders, Director of Test Systems Engineering and Evaluation at OUSD said that 40% of the DoD's software development costs are spent on reworking the software. Sanders also stated that only 16% of software development would finish on time and on budget [2].

Complexity of the computer based systems developed noticeably during the past few decades [3], [4], [5], [6] and will certainly continue in near future in almost all major and huge organizations and enterprises.

Many models have been developed since 1960 to help effort and cost estimations for software projects, but still it is a challenge due to many reasons. Some of these are:

1) the uncertainty in collected measurement,
2) the estimation methods used which might have many drawbacks and
3) the cost drivers which come with various characteristics based on the methodology of development.

- Rituraj Jain, Department of Electrical & Computer Engineering, Wollega University, Nekemte – Ethiopia. Email: jainrituraj@yahoo.com
- Mohm. Khalid Kaleem, Department of Informatics, Wollega University, Nekemte – Ethiopia. Email: khalid_ccbc@yahoo.ca
- Yohannes Bekuma, Department of Electrical & Computer Engineering, Wollega University, Nekemte – Ethiopia. Email: yohbek@gmail.com

Models for effort and cost estimation are categorized as algorithmic and non-algorithmic models [7]. Methematical computation based menthoscatrgorised as algorithmic estimation while machine learning based approaches categorized as non algorithmic estimation.

The paper is divided into 8 sections as follows. Section 2 discusses about software cost estimation models. Section 3 discusses the related work. Section 4,discusses about COCOMO model. Section 5 describes the key features of the Fuzzy logic. In section 6, the proposed method is explored. Evaluation and result of proposed method is illustrated in section 7 followed by section 8 briefing about conclusion of the findings.

2 Software Cost Estimation Models

Most cost estimation models attempt to generate an effort estimate, which can then be converted into the projectduration and cost. Generally, there are many methods for software cost estimation. As stated above these methods are divided into two groups: Algorithmic and Non-algorithmic. Using of the both groups is required for performing the accurate estimation. If the requirements are known better, their performance will be better.

Algorithmic models: Boehm's COCOMO'81, COCOMO II, Albrecht's Function Point and Putnam's SLIM are algorithmic models which estimates based on the factors like inputs, estimated specific attributes such as Lines of Code (LOC), number of user screen, interfaces and other cost drivers. At the early stage of software development process, it is not easy to acquire all these factors. At the early stages of development, it is more likely to be inaccurate identification of these factors because not much information of the project to be developed is available at that time.

Lacking of handling categorical data and reasoning capabilities in algorithmic models,non algorithmic models explored and software researchers have turned their attention to these new approaches. These are based on soft computing such as Artificial Neural Networks (ANN), Fuzzy Logic (FL) models and Genetic Algorithms (GA) [8]. To predict the software effort more correctly many researchers used their different non algorithmic models and different data sets. Artificial Neural Network is a network of nonlinear computing elements called neurons which model the functionality of human brain. Neural networks are able to generalize from trained data set. A set of rules designed with a set of training data and a specific learning algorithm that fit the data and fits previously unseen data in a rational manner. Fuzzy logic offers a powerful linguistic representation that able to represent imprecision in inputs and outputs, while providing a more knowledge based approach to model building. Research shows that fuzzy logic model achieved good performance, being outperformed in terms of accuracy only by neural network model with considerably more input variables. Use of the Fuzzy logic to derive a cost estimation model is advantageous because it interprets the linguistic values very much similar to the human way of interpretation. It is more suitable for projects with indistinct and imprecise information [9].

3 Related Work

Effort estimation during the initial stages of project development is invariably essential for the software industry to cope with the unrelenting and competitive demands of today's world. The estimation should also be accurate, reliable and precise to meet the growing demands of the industry.

W. Pedrycz et al. [10] found that the concept of information granularity and fuzzy sets, in particular, plays an important role in making software cost estimation models more users friendly. The methodology of fuzzy sets giving rise to f-COCOMO [11] is sufficiently general to be applied to other models of software cost estimation such as function point method [12]. Harish Mittal et al. [13] used triangular fuzzy numbers for fuzzy logic sizing. Ali Idri et al. [14] proposed the use of fuzzy sets in the COCOMO-81 models [15]. Lima, O.S.J. et al. [16] proposed the use of concepts and properties from fuzzy set theory to extend function point analysis to Fuzzy function point analysis, using trapezoid shaped fuzzy numbers for the linguistic variables of function point analysis complexity matrixes.

Wei Lin Du et al. [17] proposed a methodology combining the neuro-fuzzy technique and SEER-SEM that can function with various algorithmic models. A transparent and improved Fuzzy logic based framework [18] is proposed for effectively dealing with the imprecision and uncertainty problem. A multi agent system has been employed [19] to deal with the characteristics of the team members in a fuzzy system. A new framework has been elucidated [20] stage itself, especially for projects representing linguistic variables. Many studies have been carried out [21] which utilize the fuzzy systems to deal with the ambiguous and linguistic inputs of software cost estimation. The Gaussian MFs [22] have been used in the fuzzy framework, which show good results while handling the imprecision in inputs. The ability of this method to adapt itself with the varying environment as much as its efficient handling of the inherent imprecision and uncertainty problem makes it a valid choice for representing fuzzy sets. In [23], it is noted that homogeneous dataset results in better and more accurate effort estimates while the irrelevant and disordered dataset results in lesser accuracy in effort estimation.

4 COCOMO Models

Barry Bohem introduced one of the mostused techniques for cost estimation is COCOMO (constructive cost model). COCOMOconsists of hierarchy of three forms which are basic COCOMO, intermediate COCOMO and detailed COCOMO. Basic COCOMO is good for early stage estimates but estimation accuracy impacted because of missing of factors which affect the cost incurred during actual development. While intermediate COCOMO,estimate cost by considering factors and detail COCOMO adds more factors to influence of individual project phases [24].

Basic COCOMO computes software development effort (and cost) as a function of program size (in kilo source lines of code (KSLOC)).COCOMO deals with three classes of software projects: Organic projects, Semi-detached projects and Embedxed projects.

TABLE 1
BASIC COCOMO MODEL [24]

Software Project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

The basic COCOMO equations are:
Effort Applied (E) = ab (KLOC) bb [man- months]
Development Time (D) = cb (Effort Applied) db [months]
People required (P) = Effort Applied / Development Time [count]

Where E is the effort applied in person-months, D is the development time in chronological months and KLOC is the estimated number of delivered lines of code for the project (express in thousands). The coefficients ab and cb and the exponent's bb and db are given in table 1.

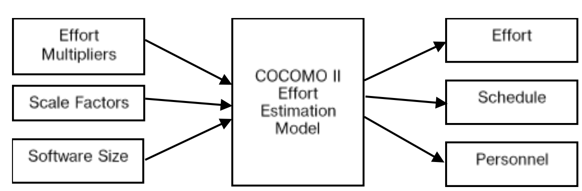


Fig. 1: COCOMO II Estimation Process

COCOMO-II is the advance version of COCOMO that predicts the amount of effort based on Person-Month (PM) in the software projects. As shown in fig.1 it uses the size metrics and composes of 22 factors (17 Effort Multipliers and 5 scale factors as shown in table 2). For all 22 factors rating levels (very low, low, nominal, high, very high and extra high) are defined and their respective quantative value as its weight is also given in table 2. The Usage of this method is very wide and its results usually are accurate. Equations used to estimate effort, Schedule and Personnel are as below [25]:

$$PM_{NS} = A * Size^{E} * \prod_{i=1}^{n} \mathrm{EM}_i \quad (1)$$

where $E = B + 0.01 * \sum_{j=1}^{5} \mathrm{SF}_j$ and A = 2.95, B = 0.91

$$TDEV_{NS} = C * (PM_{NS})^{F} \quad (2)$$

where $F = D + 0.02 * 0.01 * \sum_{j=1}^{5} \mathrm{SF}_j = D + 0.2 * (E - B)$

and C = 3.67, D = 0.28

Personnel = Effort/Schedule (3)

TABLE 2

EFFORT MULTIPLIERS AND SCALE FACTORS USED IN COCOMO II AND THEIR RATING SCALES [26]

COCOMO II Parameters			Rating Scales					
			Very Low	Low	Nomina 1	High	Very High	Extra High
SCALE FACTORS	PREC	Precedentedness	6.2	4.96	3.72	2.48	1.24	0
	FLEX	Development Flexibility	5.07	4.05	3.04	2.03	1.01	0
	RESL	Risk Resolution	7.07	5.65	4.24	2.83	1.41	0
	TEAM	Team Cohesion	5.48	4.38	3.29	2.19	1.1	0
	PMAT	Process maturity	7.8	6.24	4.68	3.12	1.56	0
EFFORT MULTIPLIERS	RELY	Required Software Reliability	0.82	0.92	1	1.1	1.26	
	DATA	Database Size		0.9	1	1.14	1.28	
	CPLX	Software Product Complexity	0.73	0.87	1	1.17	1.34	1.74
	RUSE	Required Reusability	0.95	1	1.07	1.15	1.24	
	DOCU	Documentation Match to Life-Cycle Needs		0.81	0.91	1	1.11	1.23
	TIME	Execution Time Constraint			1	1.11	1.29	1.63
	STOR	Main Storage Constraint			1	1.05	1.17	1.46
	PVOL	Platform Volatility		0.87	1	1.15	1.3	
	ACAP	Analyst Capability	1.42	1.19	1	0.85	0.71	
	PCAP	Programmer Capability	1.34	1.15	1	0.88	0.76	
	AEXP	Applications Experience	1.22	1.1	1	0.88	0.81	
	PEXP	Platform Experience	1.19	1.09	1	0.91	0.85	
	LTEX	Language and Tool Experience	1.2	1.09	1	0.91	0.84	
	PCON	Personnel Continuity	1.29	1.12	1	0.9	0.81	
	TOOL	Use of Software Tools	1.17	1.09	1	0.9	0.78	
	SITE	Multisite Development	1.22	1.09	1	0.93	0.86	0.8
	SCED	Required Development Schedule	1.43	1.14	1	1	1	

5 FUZZY LOGIC

Fuzzy logic is originated from the Fuzzy set theory and can be classify as an extensive form of the classical logical system. These techniques have found mass appeal in various computational and manufacturing engineering domains. In numerous problems of different domain, fuzzy logic has been successfully applied and also gave the useful results [27-29]. The popular fuzzy logic systems can be categorized into three types: viz. Pure fuzzy logic systems, Takagi and Sugeno's fuzzy system, and fuzzy logic system with fuzzifier and defuzzifier. Since most of the engineering applications produce crisp data as input and expects crisp data as output, the last type is the most widely used type of fuzzy logic systems. In the software engineering domain also, fuzzy logic was applied in various development phases and on the artifacts released through these phases.

A fuzzy model structure can be represented by a set of fuzzy If-Then rules [30]. It serves as a conceptual framework which works to caterthe uncertainty in the knowledge representation. In the fuzzy logic, intermediate values will be defined between conventional evaluations like yes or no, true or false, good or bad, low – medium - high, etc. and these notions can be formulated mathematically and processed by computers [31].

Fuzzy logic based approach, to solve any problem, is divided into three steps which are Fuzzification, Development of Fuzzy Rules, and Defuzzification. Fuzzification process is carried out by developing membership functions generated from different input sources. The fuzzy rule base is usually constructed from the experience of the decision maker, which will be applied over fuzzy input and arriving at the fuzzy output. They encode knowledge about a system in statement of the form:

IF (x_1 is X_1 , x_2 is X_2 ,……..x_n is X_n) THEN (y_1 is Y_1, y_2 is Y_2,……..y_n is Y_n)

where linguistic variables x_i, y_j take the value of fuzzy sets X_i and Y_j respectively.Defuzzification is the reverse procedure of the fuzzification and used to take crisp decision by applying membership functions like Max membership principle, Centroid Method (Center of Gravity Method), Weighted Average Method, Mean–max Membership etc. on the fuzzy outputs and used to represent them in a single scalar quantity [32].

6 FUZZY BASED PROPOSED APPROACH

Proposed model is established based on the COCOMO II and Fuzzy Logic. The COCOMO II includes a set of input software attributes: Effort Multipliers (EMs), Scale Factors (SFs), Size in KLOC (SZ) and one output, Effort.

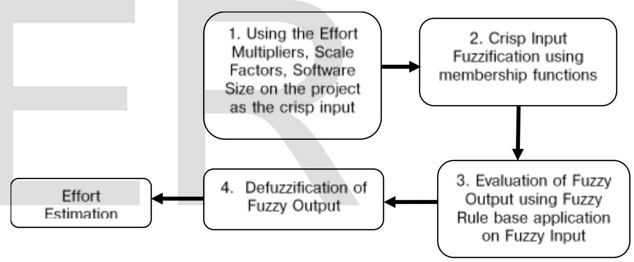
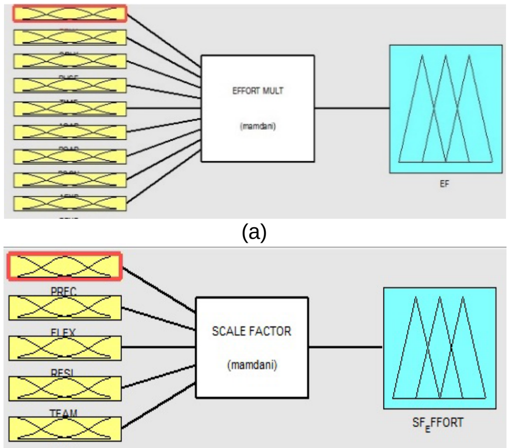


Fig.2 Architecture Flowchart for Fuzzy Approach

All these inputs to proposed system are uncertain and can be expressed in qualitative terms like Very Low, Low, Nominal, High, Very High and extra High. Therefore, it is needed to apply fuzzy based approach to quantify these qualitative terms by deriving suitable membership functions to find suitable methodology for software development process for current project. The flowchart of the used fuzzy approach is shown in fig 2. Mamdani's fuzzy inference method is used in the proposed approach. The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The factors, can be interpreted as linguistic variables, are given as input to the Fuzzy Inference System (FIS).

Not all the effort multipliers are equally important, hence only 9 key cost drivers among 17 cost drivers is considered here [32]. The nine key cost drivers are RELY (Required s/w reliability), CPLX (Product complexity), TIME (Execution time constraint), RUSE Required Reusability, ACAP(Analyst Capability), PCAP (Programmer Capability), PCON (Personnel Continuity), AEXP Applications Experience together, PEXP Platform Experience form. With these all 5 scale factors and Size we are passing as input to our approach.

The effects of effort multipliers and scale factors are the linguistic variables for output as shown in fig 3(a), (b).



(a)

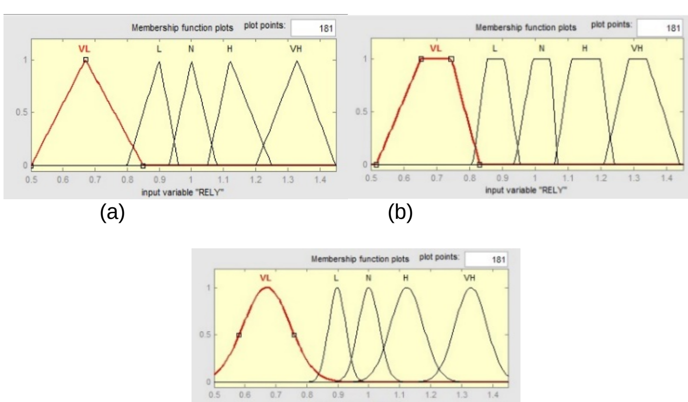
(b)

Fig. 3 Fuzzy Inference System for (a) Effort Multipliers (b) Scale Factors

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The membership functions chosen in proposed method are triangular membership functions, trapezoidal membership function and Gaussian membership function. After getting effort adjustable factor value by applying all three memberships one by one to all factors we will compare the individual effort estimated by these three membership functions with the actual effort and effort estimated by COCOMO II model.

Using the impact of effort multipliers, on the effort for project development and effort evaluation in qualitative terms such as very low, low, nominal, high, very high and extra high as the crisp input, membership functions can be generated by fuzzifing them as shown in fig 4, 5, &6.Membership Functions and their parameter values for all these variables used for the experiment are as given in table 3&4.

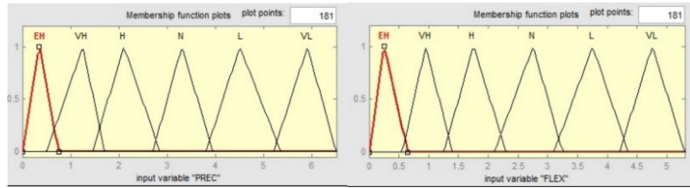
After generating the membership values of effort multipliers and scale factors, fuzzy rule base to show effects of effort multipliers and scale factors are as shown in fig 7, is constructed to arrive at the fuzzy output.



(a) (b)

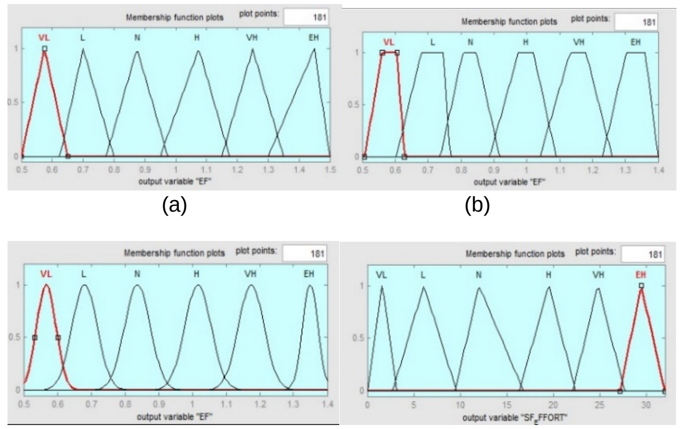
(c)

Fig. 4 Membership Function of Inputs for Effort Multiplier RELY using (a) triangular MF (b) Trapezoidal MF (c) Gaussian MF



(a) (b)

Fig. 5 Membership Function of Inputs for Scale Factors (a) PREC (b) FLEX



(a) (b)

(c) (d)

Fig. 6 Membership Function of Outputs by (a) Effort Multiplier effects using Triangular MF (b) Effort Multiplier effects using Trapezoidal MF (c) Effort Multiplier effects using Gaussian MF (d) from Scale Factor effects

Fuzzified input gives the degree to which each part of the antecedent has been satisfied for each rule. With effort multiplier fuzzification, we have nine inputs as well as with sale factor fuzzification we have 5 inputs on which the fuzzy operator AND / OR is applied to obtain one number that represents the result of the antecedent for that rule. This number will then be applied to the output function to get effect of effort multipliers in the current scenario.

The Rule Viewer displays a roadmap of the whole fuzzy inference process for both effort multipliers effects and scale factors effects. Form the rule viewer, as shown in fig 8(a) and (b), we can study the relationship of specific parameter to output and analyze the change in output function membership as the changes happened in the specific factor.

TABLE 3

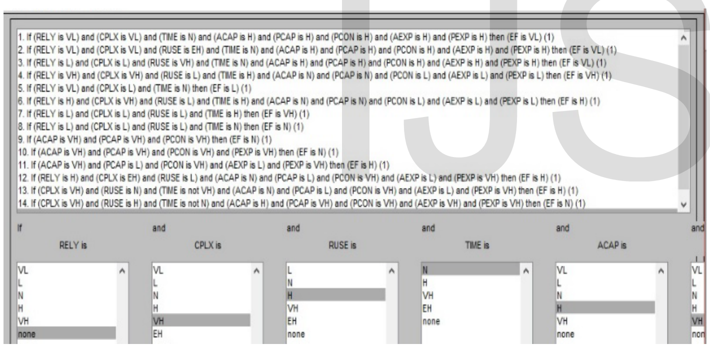
Triangular Membership Functions and their Parameter Values for all Scale Factor Variables

		Extra High (EH)	Very High (VH)	High (H)	Nominal (N)	Low (L)	Very Low (VL)
SCALE FACTORS	PREC	[0 0.35 0.75]	[0.5 1.25 1.7]	[1.45 2.1 2.85]	[2.7 3.3 3.95]	[3.8 4.5 5.35]	[5.2 5.9 6.5]
	FLEX	[0 0.25 0.65]	[0.55 0.95 1.4]	[1.25 1.75 2.3]	[2.1 2.75 3.25]	[3.1 3.75 4.35]	[4.2 4.75 5.3]
	RESL	[0 0.25 0.65]	[0.55 1.1 1.7]	[1.5 2.25 3.05]	[2.95 3.75 4.45]	[4.3 5.15 5.85]	[5.7 6.5 7.25]
	TEAM	[0 0.25 0.65]	[0.55 1 1.45]	[1.3 1.95 2.4]	[2.3 2.9 3.45]	[3.3 4 4.6]	[4.5 5 5.5]
	PMAT	[0 0.35 0.75]	[0.6 1.25 1.9]	[1.75 2.6 3.5]	[3.35 4.25 4.95]	[4.8 5.65 6.5]	[6.35 7 7.8]

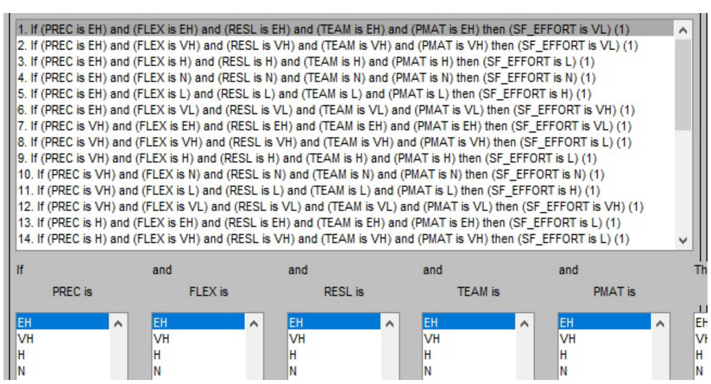
TABLE 4

TringularMembership Functions and their Parameter Values for all Effort Multiplier Variables

		Very Low (VL)	Low (L)	Nominal (N)	High (H)	Very High (VH)	Extra High (EH)
EFFORT MULTIPLIERS	RELY	[0.5 0.67 0.85]	[0.8 0.9 0.96]	[0.93 1 1.08]	[1.05 1.12 1.25]	[1.2 1.33 1.45]	
	CPLX	[0.5 0.67 0.82]	[0.78 0.85 0.93]	[0.9 0.98 1.08]	[1.05 1.12 1.25]	[1.2 1.33 1.45]	[1.4 1.6 1.75]
	TIME			[0.9 0.98 1.08]	[1.05 1.12 1.25]	[1.2 1.33 1.45]	[1.4 1.6 1.75]
	RUSE		[0.8 0.88 0.96]	[0.94 1.02 1.08]	[1.05 1.12 1.2]	[1.18 1.28 1.4]	[1.35 1.5 1.6]
	ACAP	[1.29 1.39 1.5]	[1.08 1.18 1.33]	[0.9 0.98 1.1]	[0.78 0.85 0.93]	[0.5 0.67 0.8]	
	PCAP	[1.29 1.39 1.5]	[1.08 1.18 1.33]	[0.9 0.98 1.1]	[0.78 0.85 0.93]	[0.5 0.67 0.8]	
	PCON	[1.16 1.2 1.3]	[1.04 1.12 1.3]	[0.95 1 1.06]	[0.87 0.92 0.98]	[0.5 0.8 0.9]	
	AEXP	[1.16 1.2 1.3]	[1.04 1.12 1.18]	[0.95 1 1.06]	[0.87 0.92 0.98]	[0.5 0.8 0.9]	
	PEXP	[1.16 1.2 1.3]	[1.04 1.12 1.16]	[0.95 1 1.16]	[0.87 0.92 0.98]	[0.5 0.8 0.9]	

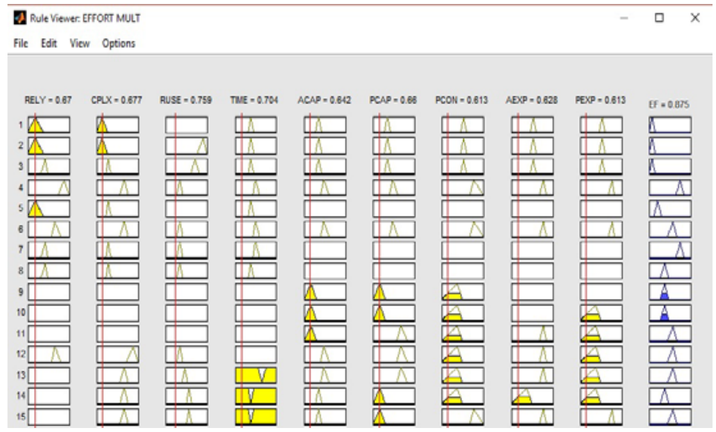


(a)

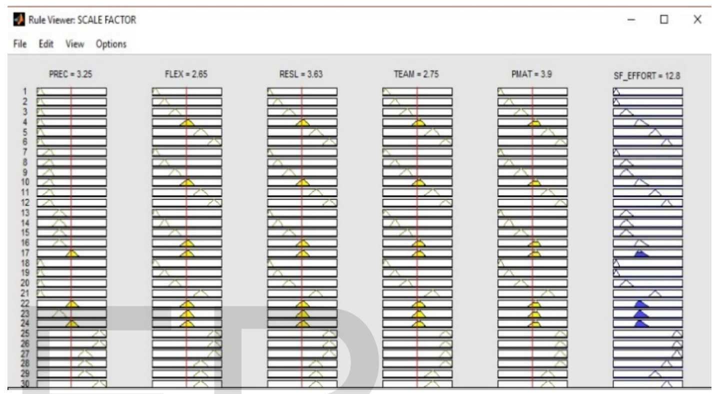


(b)

Fig.7 Fuzzy rule base to show effects of (a) Effort Multipliers and (b) Scale Factors



(a)



(b)

Fig.8 Rule View of Input /output Membership Functions (a) with effort multipliers (b) scale factors

As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number for showing suitability of that output variable representing for a methodology. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

7 Evaluation of Effort

In the next step, we evaluate the COCOMO II model using the (1) and cost drivers obtained from fuzzy sets Fuzzy_EMij rather than from the classical EMij. Fuzzy_EMij is calculated from (5), the classical EMij and the membership functions μ defined for the various fuzzy sets associated with the cost drivers. F is a linear function, where the $\mu_{A_1}^{V_i}$ is the membership function of the fuzzy set Aj associated with the cost driver Vi is shown in (5) [34].

$$Fuzzy\_EM_{ij} = F(\mu_{A_1}^{V_i}(P) \ldots \ldots \mu_{A_j}^{V_i}(P),\ C_{1j} \ldots \ldots C_{ij} \quad (4)$$

$$Fuzzy\_EM_{ij} = \sum_{j=1}^{k_i} \mu_{A_i}^{V_i}(P) * EM_{ij} \quad (5)$$

TABLE 5

Estimated Effort Using Different MF's

Sr. No.	Actual_ Effort	Triangular MF_Effort	Trapezoidal MF_Effort	Gaussian MF_Effort	COCOMO II_ Effort
1	360	420.62	441.65	378.56	381.82
2	324	364.12	390.13	348.52	352.35
3	60	96.64	108.72	67.65	70.73
4	48	78.83	101.36	56.31	60.08
5	60	70.78	80.13	68.11	70.92
6	60	78.78	92.68	64.88	68.78
7	300	345.63	382.01	316.52	324.02
8	120	157.86	165.76	134.18	140.59
9	90	106.11	121.27	99.04	101.22
10	210	235.76	251.78	224.32	237.52
11	48	73.1	87.72	56.28	57.35
12	70	92.3	105.49	73.84	85.44
13	239	267.78	281.17	257.07	260.27
14	82	113.23	122.66	84.92	91.71
15	62	96.04	104.04	75.23	77.79
16	170	228.53	238.05	180.92	185.1
17	192	232.04	263.68	200.4	205.03
18	18	28.56	30.56	22.24	25.08
19	50	77.22	92.67	61.78	68.77
20	60	78.06	93.67	64.01	69.52

Data set used for the evaluation is using public domain data set [35]. This data set provides data for standard COCOMO attributes in the range Very Low to Extra High; one lines of code measure (KLOC), the actual effort in person months, total defects and last being development time in months. Results obtained by using the proposed approach were compared by effort values given in data set for conventional COCOMO II. The results obtained by means of applying all three membership function**s**of fuzzy logic which are Triangular, Trapezoidal and Gaussian MF's were analyzed. It was observed that by fuzzifying 17 effortsmultiply, scale factors and size, Gaussian MF gives the better results than the Triangular and Trapezoidal MFs. The estimated efforts using COCOMO II, Triangular Membership Function, Trapezoidal Membership Function and Gaussian Membership Function obtained are tabulated and compared as show in table 5.

Finally**,** the accuracy of the estimated effort with the actual effort is done for the evaluation process. For the evaluation process**,** the Mean Magnitude of Relative Error (MMRE) evaluation criteria and Prediction PRED(L) is used as shown in (6)and (8)is used.Prediction PRED(L) is the probability of the model having relative Error less than or equal to L.

$$MMRE = \frac{1}{N}\sum_{1}^{N} \mathrm{MRE}_i \quad (6)$$

where $$MRE_i = \frac{|EM_{est}^i - EM_{act}^i|}{EM_{act}^i} * 100 \quad (7)$$

$$PRED(L) = {}^{M_{obs}}/_{N_{obs}} \quad (8)$$

EM_{est}^i is the value of estimated effort and EM_{act}^i is the actual effort used during the project development. M_{obs} is the number of observations where MRE is less than or equal to L and N_{obs} is the total number of observations. MRE is calculated for each case for which effort is estimated. MRE over multiple observations N is calculated by MMRE as shown in (6).

TABLE 6

Comparision of MRE from all 3 MF's & COCOMO II

Table 6 shows comparison of MRE of efforts estimated by using triangular, trapezoidal and gaussian membership functions for every esrtimate. Comparison of the model results in table 7 shows that the Gaussian member function has better estimation accuracy as compared to other models having 10.92 MMRE. In order to further verify the prediction, analys**is** the results for 20%, 15%, 10% prediction. The results show that the effort estimation based on gaussian membership function is 55% confident to have its average MRE, which shows the stability of its estimates.

TABLE 7

Sr. No.	MRE_Tringular MF	MRE_Trapezodial MF	MRE_Gaussian MF	MRE_COCOMO II
1	16.84	22.68	5.16	6.06
2	12.38	20.41	7.57	8.75
3	61.07	81.20	12.75	17.88
4	64.23	111.17	17.31	25.17
5	17.97	33.55	13.52	18.20
6	31.30	54.47	8.13	14.63
7	15.21	27.34	5.51	8.01
8	31.55	38.13	11.82	17.16
9	17.90	34.74	10.04	12.47
10	12.27	19.90	6.82	13.10
11	52.29	82.75	17.25	19.48
12	31.86	50.70	5.49	22.06
13	12.04	17.64	7.56	8.90
14	38.09	49.59	3.56	11.84
15	54.90	67.81	21.34	25.47
16	34.43	40.03	6.42	8.88
17	20.85	37.33	4.38	6.79
18	58.67	69.78	23.56	39.33
19	54.44	85.34	23.56	37.54
20	30.10	56.12	6.68	15.87

Comparision of MMRE and PRED(L) for all MF's &COCOMO II

8 Conclusion

Referring to table 5, 6 & 7, Gaussian member function based estimation gives the better results for maximum criterions when compared with the other methods. Because of smoother transition in the interval in Gaussian MF it is analyzed that it is performing better than trapezoidal MF and triangular MF, and gives results which were closer to the actual effort (as given in table 5 and represented graphically in fig.9).Thus it is concluded that the new approach using Gaussian MF is better than other methods. By suitably adjusting the values of the parameters in FIS, we can optimize the estimated effort. Future work includes multistage evaluation of effort using multistage fuzzy system and also with neuro-fuzzy system to get more stable and efficient results.

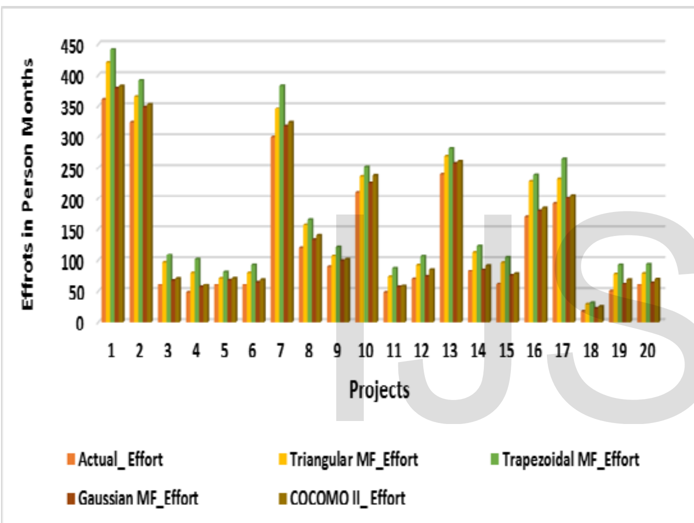


Fig.9 Graphical representation of comparison of Fuzzy based estimated effort with COCOMO II and Actual Effort

References

[1] Z. Zia, A. Rashid and K. U. Zaman, "Software cost estimation for component based fourth-generation-language software applications", I T Software, vol. 5, no. 1, pp. 103-110, 2011

[2] Alaa F. Sheta, Sultan Aljahdali, "Software effort estimation by tuning COOCMO model parameters using differential evolution", 2013 ACS International Conference on Computer Systems and Applications (AICCSA), vol. 00, no. , pp. 1-6, 2010, doi:10.1109/AICCSA.2010.5586985

[3] C. F. Kemere, "An empirical validation of software cost estimation models," Communication ACM, vol. 30, pp. 416–429, 1987.

[4] J. W. Park R, W. Goethert, "Software cost and schedule estimating: A process improvement initiative," tech. rep., 1994.

[5] M. Boraso, C. Montangero, and H. Sedehi, "Software cost estimation: An experimental study of model performances," tech. rep., 1996.

[6] K. Pillai and S. Nair, "A model for software development effort and cost estimation," IEEE Trans. on Software Engineering, vol. 23, pp. 485– 497, 1997.

[7] VahidKhatibi, Dayang N. A. Jawawi.: Software Cost Estimation Method A Review. Journal of Emerging Trends in Computing and Information Sciences. Vol.2, No.1, 2010.

[8] ImanAttarzadeh, Siew Hock Ow, "Proposing a New High Performance Model for Software Cost Estimation", 2009 Second International Conference on Computer and Electrical Engineering, page 112-11

[9] Ch. Satyananda Reddy and KVSVN Raju.: An Improved Fuzzy Approach for COCOMO's Effort Estimation using Gaussian Membership Function. Journal of Software, Vol. 4, No. 5, Pp. 452-459, 2009.

[10] W. Pedrycz, F. F. Peters and S. Ramanna, "A fuzzy Set Approach to Cost estimation of Software Projects", Proceedings of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering Shaw Conference Center, Edmonton Alberta, Canada, (1999).

[11] P. Musílek, W. Pedrycz, G. Succi and M. Reformat, "Software Cost estimation with fuzzy models", ACM SIGAPP Applied Computing Review, vol. 8, no. 2, (2000), pp. 24-29.

[12] J. E. Matson, B. E. Barrett and J. M. ellichamp, "Software Development Cost estimation Using unction Points", I Trans. on Software Engineering, vol. 20, no. 4, (1994), pp. 275-287.

[13] H. Mittal and P. Bhatia, "Optimization Criterion for effort estimation using fuzzy Technique", CL I Electronic Journal, vol. 10, no. 1, (2007), pp. 2.

[14] A. Idri, A. Abran and L. Kjiri, "COCO O cost model using fuzzy Logic", 7th International Conference on Fuzzy Theory & Technology Atlantic, New Jersey, (2000).

[15] B. Boehm, "Software engineering Economics", Englewood Cliffs, NJ Prentice-Hall, (1981).

[16] O. S. J. Lima, P. P. Farias and A. D. Belchor, "A fuzzy model for Function Point Analysis to Development and Enhancement Project Assessments", CL I J, vol. , no. 2, (2002).

[17] Wei Lin Du, Danny Ho and Luiz Fernando Capretz.: Improving Software Effort Estimation Using Neuro-Fuzzy Model with SEER-SEM. Global Journal of Computer Science and Technology. Vol. 10, No. 12, Pp. 52-64, 2010.

[18] M.A. Ahmeda, Z. Muzaffar,: Handling imprecision and uncertainty in software development effort prediction: a type-2 fuzzy logic based framework. Information and Software Technology 51, 640–654 (2009)

[19] M.Kazemifard, A.Zaeri, N.ghasem-ghaee, M.A.Nematbakhsh, F.Mardukhi.: Fuzzy Emotional COCOMO II Software Cost Estimation (FECSCE) using Multi-Agent Systems. Applied Soft Computing, Elsevier.pg.2260-2270, 2011.

[20] RoheetBhatnagar, VandanaBhattacharjee and MrinalKantiGhose.: A Proposed Novel Framework for Early Effort Estimation using Fuzzy Logic Techniques. Global Journal of Computer Science and Technology. Vol. 10, No. 14, Pp. 66-72, 2010.

[21] ImanAttarzadeh and Siew Hock Ow.: Improving the Accuracy of Software Cost Estimation Model Based on a new Fuzzy Logic Model, World applied sciences Journal. Vol. 8, No. 2, pp. 177-184, 2010.

[22] Jacky W. Keung.: Theoretical Maximum Prediction Accuracy for

Sr. No.	MRE_Tringular MF	MRE_Trapezodial MF	MRE_Gaussian MF	MRE_COCOMO II
MMRE	33.42	50.03	10.92	16.88
PRED (20)	35	10	90	80
PRED (15)	15	0	70	50
PRED (10)	0	0	55	30

Analogy-based Software Cost Estimation.15th Asia-Pacific Software Engineering Conference, 2008.

[23] Harsh Kumar Verma, Vishal Sharma.: Handling Imprecision in Inputs using Fuzzy Logic to Predict Effort in Software Development. IEEE , 2010.

[24] http://www.mhhe.com/engcs/compsci/pressman/information/olc/COCOMO.html [accessed on 02/01/2017]

[25] VahidKhatibi, Dayang N. A. Jawawi, Software Cost Estimation Methods: A Review, Journal of Emerging Trends in Computing and Information Sciences Volume 2 No. 1 pp 21-29, 2010-11

[26] Baik, J., Chulani, S., Horowitz, S.: Software Effort And Schedule Estimation Using The Constructive Cost Model: COCOMO II, Center for Systems and Software Engineering, University of Southern California, USA, https://pdfs.semanticscholar.org/0998/1c62358da9336340a77100e1a97d127579c4.pdf [accessed on 02/01/2017]

[27] DA White and DA Sofge, Handbook of Intelligent Control, Neural, Fuzzy, and Adaptive Approaches, Van Nostrand Reinhold, New York, 1992.

[28] LX Wang and JM Mendel, Fuzzy Adaptive Filter: with Application to Nonlinear Channel Equalization, Proceeding of 2nd International Conference on Fuzzy System, 1993, 895–900.

[29] M Brown and CJ Harris, Neurofuzzy Adaptive Modeling and Control, Hemel Hempstead: Prentice Hall, 1994.

[30] B Kosko, Fuzzy Systems as Universal Approximators, Proceeding of International Conference on Fuzzy System, 1998, 1153–1162.

[31] B Peter, N Stephan and W Roman, A Brief Course in Fuzzy Logic and Fuzzy Control, Department of Knowledge based Mathematical Systems, www.esru.strath.ac.uk/Reference/concepts/fuzzy/fuzzy.htm, 1996.

[32] Ranjan Srivastava, Praveen Kumar, Sirish Singh, AjitPratap and G Raghurama, Software Testing Effort: An Assessment Through Fuzzy Criteria Approach, Journal of Uncertain Systems, 2011, 5 (3), 183-201.

[33] Liang tian, AfzelNoore, “Multistage Software Estimation”. 2003 IEEE, 232-236.

[34] GeetikaBatra and Mahima Trivedi "A Fuzzy Approach for Software Effort Estimation", International Journal on Cybernetics &Informatics (IJCI) Vol.2, No.1, Feb 2013

[35] http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa_2.arff [accessed on 04/01/2017]