

# Application of Fuzzy Logic Approach to Software Effort Estimation

Prasad Reddy P.V.G.D  
Dept. of CSSE  
Andhra University  
Visakhapatnam, INDIA

Sudha K. R  
Dept. of EE  
Andhra University  
Visakhapatnam, INDIA

Rama Sree P  
Dept. of CSSE  
Aditya Engineering College, JNTUK  
Kakinada, INDIA

**Abstract**— The most significant activity in software project management is Software development effort prediction. The literature shows several algorithmic cost estimation models such as Boehm's COCOMO, Albrecht's' Function Point Analysis, Putnam's SLIM, ESTIMACS etc., but each model do have their own pros and cons in estimating development cost and effort. This is because project data, available in the initial stages of project is often incomplete, inconsistent, uncertain and unclear. The need for accurate effort prediction in software project management is an ongoing challenge. A fuzzy model is more apt when the systems are not suitable for analysis by conventional approach or when the available data is uncertain, inaccurate or vague. Fuzzy logic is a convenient way to map an input space to an output space. Fuzzy Logic is based on fuzzy set theory. A fuzzy set is a set without a crisp, clearly defined boundary. It is characterized by a membership function, which associates with each point in the fuzzy set a real number in the interval [0, 1], called degree or grade of membership. The membership functions may be Triangular, GBell, Gauss and Trapezoidal etc. In the present paper, software development effort prediction using Fuzzy Triangular Membership Function and GBell Membership Function is implemented and compared with COCOMO. A case study based on the NASA93 dataset compares the proposed fuzzy model with the Intermediate COCOMO. The results were analyzed using different criterions like VAF, MARE, VARE, MMRE, Prediction and Mean BRE. It is observed that the Fuzzy Logic Model using Triangular Membership Function provided better results than the other models.

**Keywords**- *Development Effort; EAF; Cost Drivers; Fuzzy Identification; Membership Functions; Fuzzy Rules; NASA93 dataset.*

## I. INTRODUCTION

In algorithmic cost estimation [1], costs and efforts are predicted using mathematical formulae. The formulae are derived based on some historical data [2,19]. The best known algorithmic cost model called COCOMO (COnstructive COSt MOdel) was published by Barry Boehm in 1981[3]. It was developed from the analysis of sixty three (63) software projects. Boehm projected three levels of the model called Basic COCOMO, Intermediate COCOMO and Detailed COCOMO [3, 5]. In the present paper we mainly focus on the Intermediate COCOMO.

### A. Intermediate COCOMO

The Basic COCOMO model [3] is based on the relationship: Development Effort,  $DE = a*(SIZE)^b$ ; where, SIZE is measured in thousand delivered source instructions. The constants a, b are dependent upon the 'mode' of development of projects. DE is measured in man-months. Boehm proposed 3 modes of projects [3]:

- Organic mode* – simple projects that engage small teams working in known and stable environments.
- Semi-detached mode* – projects that engage teams with a mixture of experience. It is in between organic and embedded modes.
- Embedded mode* – complex projects that are developed under tight constraints with changing requirements.

The accuracy of Basic COCOMO is limited because it does not consider the factors like hardware, personnel, use of modern tools and other attributes that affect the project cost. Further, Boehm proposed the Intermediate COCOMO[3,4] that adds accuracy to the Basic COCOMO by multiplying 'Cost Drivers' into the equation with a new variable: EAF (Effort Adjustment Factor) shown in Table I.

TABLE I. DE FOR THE INTERMEDIATE COCOMO

Development mode	Intermediate Effort Equation
Organic	$DE = EAF * 3.2 * (SIZE)^{1.05}$
Semi-detached	$DE = EAF * 3.0 * (SIZE)^{1.12}$
Embedded	$DE = EAF * 2.8 * (SIZE)^{1.2}$

The EAF term is the product of 15 Cost Drivers [5, 11] that are listed in Table II. The multipliers of the cost drivers are Very Low, Low, Nominal, High, Very High and Extra High. For example, for a project, if RELY is Low, DATA is High, CPLX is extra high, TIME is Very High, STOR is High and rest parameters are nominal then  $EAF = 0.75 * 1.08 * 1.65 * 1.30 * 1.06 * 1.0$ . If the category values of all the 15 cost drivers are "Nominal", then EAF is equal to 1.

TABLE II. INTERMEDIATE COCOMO COST DRIVERS WITH MULTIPLIERS

S. No	Cost Driver Symbol	Very low	Low	Nominal	High	Very high	Extra high
1	RELY	0.75	0.88	1.00	1.15	1.40	—

2	DATA	—	0.94	1.00	1.08	1.16	—
3	CPLX	0.70	0.85	1.00	1.15	1.30	1.65
4	TIME	—	—	1.00	1.11	1.30	1.66
5	STOR	—	—	1.00	1.06	1.21	1.56
6	VIRT	—	0.87	1.00	1.15	1.30	—
7	TURN	—	0.87	1.00	1.07	1.15	—
8	ACAP	—	0.87	1.00	1.07	1.15	—
9	AEXP	1.29	1.13	1.00	0.91	0.82	—
10	PCAP	1.42	1.17	1.00	0.86	0.70	—
11	VEXP	1.21	1.10	1.00	0.90	—	—
12	LEXP	1.14	1.07	1.00	0.95	—	—
13	MODP	1.24	1.10	1.00	0.91	0.82	—
14	TOOL	1.24	1.10	1.00	0.91	0.83	—
15	SCED	1.23	1.08	1.00	1.04	1.10	—

The 15 cost drivers are broadly classified into 4 categories [3, 5].

- a) *Product*: RELY - Required software reliability  
DATA - Data base size  
CPLX - Product complexity
- b) *Platform*: TIME - Execution time  
STOR- Main storage constraint  
VIRT - Virtual machine volatility  
TURN - Computer turnaround time
- c) *Personnel*: ACAP - Analyst capability  
AEXP - Applications experience  
PCAP - Programmer capability  
VEXP - Virtual machine experience  
LEXP - Language experience
- d) *Project*: MODP - Modern programming  
TOOL - Use of software tools  
SCED - Required development schedule

Depending on the projects, multipliers of the cost drivers will vary and thereby the EAF may be greater than or less than 1, thus affecting the Effort [5].

## II. FUZZY IDENTIFICATION

A fuzzy model [13,15] is used when the systems are not suitable for analysis by conventional approach or when the available data is uncertain, inaccurate or vague [7]. The point of Fuzzy logic is to map an input space to an output space using a list of if-then statements called rules. All rules are evaluated in parallel, and the order of the rules is unimportant. For writing the rules, the inputs and outputs of the system are to be identified. To obtain [18] a fuzzy model from the data available, the steps to be followed are,

- Select a Sugeno type Fuzzy Inference System.
- Define the input variables mode, size and output variable effort.
- Set the type of the membership functions (TMF or GBellMF) for input variables.

- Set the type of the membership function as linear for output variable.
- The data is now translated into a set of if-then rules written in Rule editor.
- A certain model structure is created, and parameters of input and output variables can be tuned to get the desired output.

### A. Fuzzy Approach for Prediction of Effort

The Intermediate COCOMO model data is used for developing the Fuzzy Inference System (FIS)[10]. The inputs to this system are MODE and SIZE. The output is Fuzzy Nominal Effort. The framework [8] is shown in “Fig. 1”.

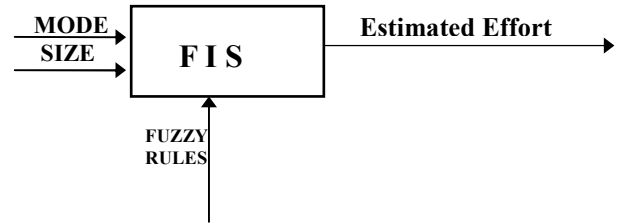


Figure 1: Fuzzy Framework

Fuzzy approach [9] specifies the SIZE of a project as a range of possible values rather than a specific number. The MODE of development is specified as a fuzzy range. The advantage of using the fuzzy ranges[14] is that we will be able to predict the effort for projects that do not come under a precise mode i.e. comes in between 2 modes. This situation cannot be handled using the COCOMO. The output of this FIS is the Fuzzy Nominal Effort. The Fuzzy Nominal Effort multiplied by the EAF gives the Estimated Effort. The FIS[16] needs appropriate membership functions and rules.

### B. Fuzzy Membership Functions

A membership function (MF) [9, 12] is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is also called as the universe of discourse. For our problem, we have used 2 types of membership functions:

- 1) Triangular membership function
- 2) Generalized Bell membership function

#### Triangular membership function (TMF):

It is a three-point function [17], defined by minimum ( $\alpha$ ), Maximum ( $\beta$ ) and modal ( $m$ ) values, that is, TMF ( $\alpha, m, \beta$ ), where ( $\alpha \leq m \leq \beta$ ). The parameters  $\alpha$  and  $\beta$  locate the "feet" of the triangle and the parameter  $m$  locates the peak. The triangular curve is a function of a vector,  $x$ , and depends on three scalar parameters  $\alpha$ ,  $m$ , and  $\beta$  as given by

$$f(x; \alpha, m, \beta) = \max \left( \min \left( \frac{x - \alpha}{m - \alpha}, \frac{\beta - x}{\beta - m} \right), 0 \right)$$

(1)

Please refer to “Fig. 2” for a sample triangular membership function.

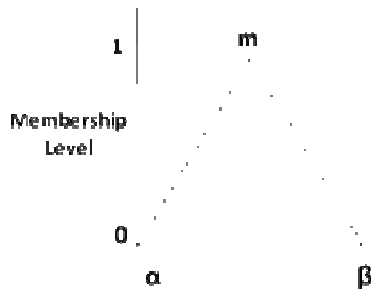


Figure 2: A Sample Triangular Membership Function

The fuzzy set definitions for the MODE of development appear in “Fig. 3” and the fuzzy set [8] for SIZE appear in “Fig. 4”.

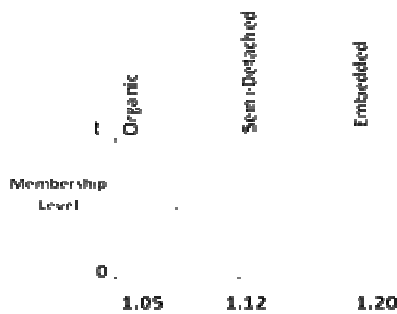


Figure 3: Fuzzy Set for Mode

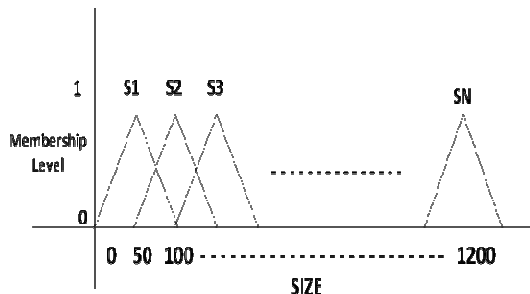


Figure 4: Fuzzy set for SIZE

**Generalized Bell Membership Function (GBellMF):**

It is a three-point function, defined by minimum ( $\alpha$ ), maximum ( $\beta$ ) and modal ( $m$ ) values, that is,  $GBellMF(\alpha, m, \beta)$ , where ( $\alpha \leq m \leq \beta$ ). Please refer to “Fig. 5” for a sample Generalized Bell membership function. The generalized bell function depends on three parameters  $\alpha$ ,  $m$ , and  $\beta$  as given by

$$f(x; \alpha, m, \beta) = \frac{1}{1 + \left| \frac{x - \beta}{\alpha} \right|^{2m}} \tag{2}$$

where the parameter  $m$  is usually positive. The parameter  $\beta$  locates the center of the curve.

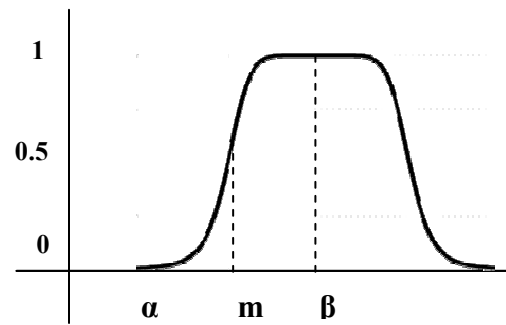


Figure 5: A Sample Generalized Bell Membership Function

We can get the Fuzzy sets for MODE, SIZE and Effort for GBellMF in the same way as in triangular method, but the difference is only in the shape of the curves.

*C. Fuzzy Rules*

Our rules based on the fuzzy sets [9, 20] of MODE, SIZE and EFFORT appears in the following form:

- If MODE is organic and SIZE is s1 then EFFORT is EF1
- If MODE is semidetached and SIZE is s1 then EFFORT is EF2
- If MODE is embedded and SIZE is s1 then EFFORT is EF3
- If MODE is organic and SIZE is s2 then EFFORT is EF4
- If MODE is semidetached and SIZE is s2 then EFFORT is EF5
- If MODE is embedded and SIZE is s3 then EFFORT is EF5
- If MODE is embedded and SIZE is s4 then EFFORT is EF3
- If MODE is organic and SIZE is s3 then EFFORT is EF4
- If MODE is embedded and SIZE is s5 then EFFORT is EF6
- If MODE is organic and SIZE is s4 then EFFORT is EF4
- .....

**III. VARIOUS CRITERIONS FOR ASSESSMENT OF SOFTWARE EFFORT**

*a) Variance Accounted For (VAF)*

$$VAF (\%) = \left( 1 - \frac{\text{var}(E - \hat{E})}{\text{var}E} \right) \times 100 \tag{3}$$

*b) Mean Absolute Relative Error (MARE)*

$$MARE (\%) = \frac{\sum f(R_E)}{\sum f} \times 100 \tag{4}$$

*c) Variance Absolute Relative Error (VARE)*

$$VARE (\%) = \frac{\sum f(R_E - \text{mean}R_E)^2}{\sum f} \times 100 \tag{5}$$

*d) Prediction (n)*

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

*e) Balance Relative Error (BRE)*

$$BRE = \frac{|E - \hat{E}|}{\min(E, \hat{E})} \tag{6}$$

Where,  $E$  = Estimated effort  $\hat{E}$  = Actual effort

$$\text{Absolute Relative Error (RE)} = \frac{|E - \hat{E}|}{|E|} \quad (7)$$

f) Mean Magnitude of Relative Error (MMRE)

$$\text{MMRE}(\%) = \frac{1}{N} \sum_{i=1}^N \text{MRE}_i \times 100 \quad (8)$$

$$\text{Where } \text{MRE} = \frac{|\hat{E} - E|}{\hat{E}} \quad (9)$$

$N$  = No. of Projects,  $E$  = Estimated effort,  $\hat{E}$  = Actual effort

A model which gives higher VAF is better than that which gives lower VAF. A model which gives higher Pred(n) is better than that which gives lower Pred(n). A model which gives lower MARE is better than that which gives higher MARE [11, 12]. A model which gives lower VARE is better than that which gives higher VARE [6]. A model which is having less MMRE is better than the model which is having higher MMRE. A model which gives lower BRE is better than that which gives higher BRE. A model which gives lower MMRE is better than that which gives higher MMRE.

#### IV. EXPERIMENTAL STUDY

In carrying out our experiments, we have chosen 93 projects of the NASA 93 dataset. The estimated efforts using Intermediate COCOMO, Fuzzy using TMF and GBellMF are shown in Table III. Table IV and “Fig. 6” to “Fig. 12” shows the comparisons of various models basing on different criterions.

TABLE III. ESTIMATED EFFORT IN MAN MONTHS OF VARIOUS MODELS

Project ID	Actual Effort	Estimated Effort using		
		COCOMO	Fuzzy-TriMF	Fuzzy-GbellMF
1	8.40	2.30	12.61	14.05
9	36.00	27.80	42.98	45.31
12	42.00	32.30	35.86	32.37
13	42.00	35.40	42.14	43.83
17	50.00	36.30	52.23	60.74
19	60.00	50.60	67.46	80.99
26	72.00	33.00	93.12	94.04
36	120.00	116.70	134.45	113.77
41	192.00	574.20	192.00	172.59
45	239.00	224.70	245.30	268.04
49	300.00	290.50	355.46	354.73
53	352.80	290.50	355.46	354.73
59	420.00	436.90	448.12	483.07
61	432.00	615.50	438.00	269.83
68	576.00	821.10	533.08	652.35
77	882.00	1332.30	882.03	881.34
81	1248.00	1113.60	1183.38	1004.62
83	1368.00	1139.60	1358.98	1358.40
92	4560.00	24726.50	4559.60	4581.44

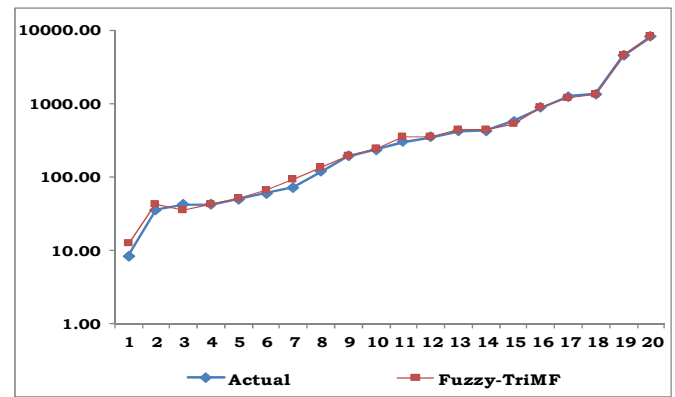


Figure 6: Estimated Effort using Fuzzy-TriMF versus Actual Effort

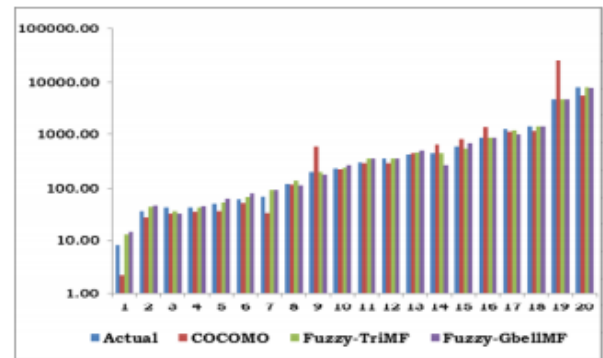


Figure 7: Estimated Effort using Various Models versus Actual Effort

TABLE IV. COMPARISON OF VARIOUS MODELS

Model	VAF	MARE	VARE	Mean BRE	MMRE	Pred(30)%
Fuzzy-TriMF	96.53	28.53	10.51	0.61	54.81	62
Fuzzy-GBellMF	95.90	23.78	32.59	0.59	63.16	65
COCOMO	33.65	47.22	46.89	0.78	59.50	53

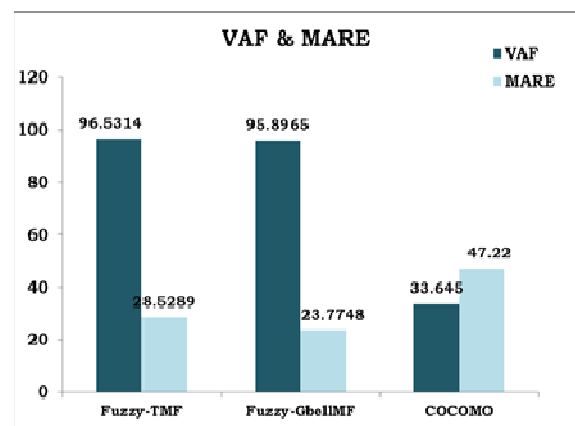


Figure 8: Comparison of VAF & MARE against various models

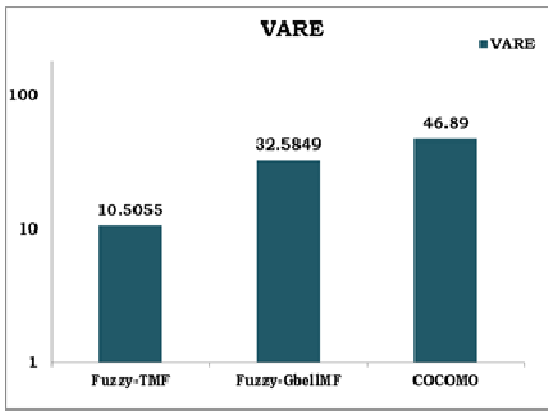


Figure 9: Comparison of VARE against various models

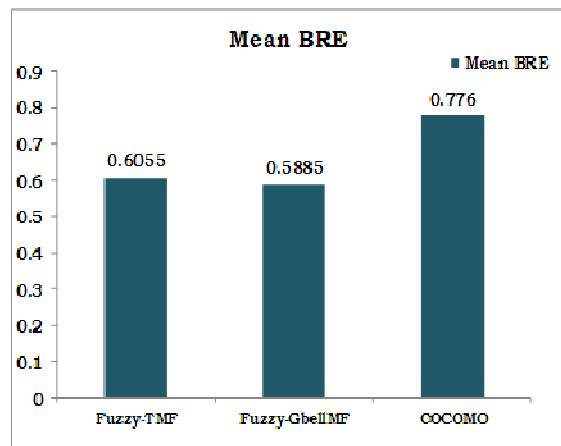


Figure 10: Comparison of Mean BRE against various models

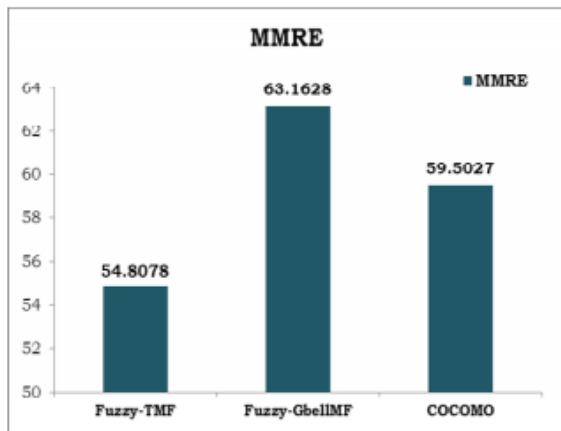


Figure 11: Comparison of MMRE against various models

## V. CONCLUSION

Referring to Table 4, we see that Fuzzy using TMF yields better results for maximum criterions when compared with the other methods. Thus, basing on VAF, MMRE, VARE, MARE, Pred(30) & Mean BRE, we come to a conclusion that the Fuzzy method using TMF (triangular membership function) is better than Fuzzy method using GbellMF or Intermediate COCOMO. It is not possible to evolve a method, which can give 100 % VAF. By suitably adjusting the values of the parameters in FIS we can optimize the estimated effort.

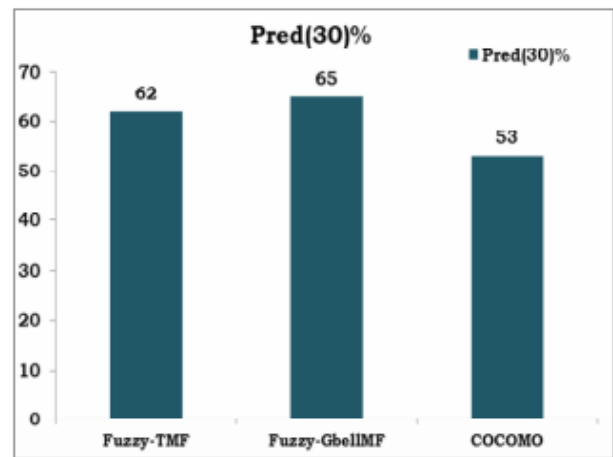


Figure 12: Comparison of Pred (30) % against various models

## REFERENCES

- [1] RamilJ.F., Algorithmic cost estimation for software evolution, Software Engg. (2000) 701-703.
- [2] Angelis L, Stamelos I, Morisio M, Building software cost estimation model based on categorical data, Software Metrics Symposium, 2001-Seventh International Volume (2001) 4-15.
- [3] B.W. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Clis, NJ, 1981
- [4] Kirti Seth, Arun Sharma & Ashish Seth, Component Selection Efforts Estimation– a Fuzzy Logic Based Approach, IJCSS-83, Vol (3), Issue (3).
- [5] Zhiwei Xu, Taghi M. Khoshgoftaar, Identification of fuzzy models of software cost estimation, Fuzzy Sets and Systems 145 (2004) 141–163
- [6] Harish Mittal, Harish Mittal, Optimization Criteria for Effort Estimation using Fuzzy Technique, CLEI Electronic Journal, Vol 10, No 1, Paper 2, 2007
- [7] R. Babuska, Fuzzy Modeling for Control, Kluwer Academic Publishers, Dordrecht, 1999
- [8] Moshood Omolade Saliu, Adaptive Fuzzy Logic Based Framework for Software Development Effort Prediction, King Fahd University of Petroleum & Minerals, April 2003
- [9] Iman Attarzadeh and Siew Hock Ow, Software Development Effort Estimation Based on a New Fuzzy Logic Model, IJCTE, Vol. 1, No. 4, October2009
- [10] Xishi Huang, Danny Ho, Jing Ren, Luiz F. Capretz, A soft computing framework for software effort estimation, Springer link, Vol 10, No 2 Jan-2006
- [11] Prasad Reddy P.V.G.D, Sudha K.R , Rama Sree P & Ramesh S.N.S.V.S.C, Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks, Journal of Computing, Vol 2, Issue 5 May 2010
- [12] Prasad Reddy P.V.G.D, Sudha K.R , Rama Sree P & Ramesh S.N.S.V.S.C, Fuzzy Based Approach for Predicting Software Development Effort, International Journal of Software Engineering, Vol 1, Issue 1, June2010
- [13] Zonglian F. and Xihui L., “f-COCOMO: Fuzzy Constructive Cost Model in Software Engineering”, Proc. of IEEE Int. Conf. On Fuzzy Systems, IEEE, 1992, 331-337.
- [14] Ryder J., “Fuzzy Modeling of Software Effort Prediction”, Proc. of IEEE Information Technology Conference, Syracuse, NY, 1998.
- [15] Idri A. and Abran A., “COCOMO Cost Model Using Fuzzy Logic”, 7th International Conference on Fuzzy Theory & Technology, Atlantic City, New Jersey, March 2000.
- [16] M. W. Nisar, Yong-Ji Wang, M. Elahi and I.A Khan, “Software Development Effort Estimation Using Fuzzy Logic”, Information

- Technology Journal, 2009 Asian Network for Scientific Information, 2009.
- [17] <http://www.mathworks.com/help/toolbox/fuzzy/fp351dup8.html>
- [18] M. Braz and S. Vergilio, "Using Fuzzy Theory for Effort Estimation of Object-Oriented Software", Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2004.
- [19] B. A. Kitchenham and N. R. Taylor "Software Cost Models", ICL Technical Journal, May 1984
- [20] W. Pedrycz, J.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 May 9-12, 1999.
- [21] Ahmad, Y., & Husain, S. (2010). Applying Intuitionistic Fuzzy Approach to Reduce Search Domain in an Accidental Case. International Journal of Advanced Computer Science and Applications - IJACSA, 1(4).
- [22] Barik, S., Mishra, D., Mishra, S., Satapathy, S. K., Rath, A. K., & Acharya, M. (2010). Pattern Discovery using Fuzzy FP-growth Algorithm from Gene Expression Data. International Journal of Advanced Computer Science and Applications - IJACSA, 1(5), 50-55.
- [23] Khan, A. R., Rehman, Z. U., & Amin, H. U. (2011). Application of Expert System with Fuzzy Logic in Teachers ' Performance Evaluation. International Journal of Advanced Computer Science and Applications - IJACSA, 2(2), 51-57.