

Optimization Criteria for Effort Estimation using Fuzzy Technique

Harish Mittal

Department of Computer Science
Vaish College of Engineering
Rohtak-124201 (INDIA)
harish.mittal@vcenggrtk.com

Pradeep Bhatia

Department of Computer Science and Engineering
G.J. University of Science and Technology
Hisar-125001 (INDIA)
pk_hatia2002@yahoo.com

Abstract

Effective cost estimation is the most challenging activity in software development. Software cost estimation is not an exact science. Cost estimation process involves a series of systematic steps that provide estimate with acceptable risk. Some prevalent LOC based models are- Bailey Basili model, Alaa F. Sheta G.E. Model, and Alaa F. Sheta, Model 2 [1]. Two new models, based on fuzzy logic sizing, are presented in this paper. Rather than using a single number, the software size is regarded as a triangular fuzzy number. We can optimize the estimated effort for any application by varying arbitrary constants for these models. The developed models are tested on 10 NASA software projects [2], on the basis of four criterions for assessment of software cost estimation models. Comparison of all the models, cited above, is done and it is found that the developed models provide better estimation.

Key Words: LOC based models, NASA Software Projects, Fuzzy logic, Triangular Fuzzy Number, Membership function and Fuzziness.

1. INTRODUCTION

Out of the three principal components of cost i.e., hardware costs, travel and training costs, and effort costs, the effort cost is dominant. Software cost estimation starts at the proposal state and continues throughout the life time of a project. There are seven techniques of software cost estimation:

- Algorithm Cost Model
- Expert Judgments
- Estimation by Analogy
- Parkinson's Law
- Pricing to win
- Top-down estimation
- Bottom-up estimation

If these predict radically different costs, more estimation should be sought and the costing process repeated. Algorithm model, also called parametric model, and is designed to provide some mathematical equations to provide software estimation. LOC-based models are algorithm models [3,12,13,14]. Ali Idri and Laila Kjjri [7] proposed the use of fuzzy sets in the COCOMO, 81 model [3]. Musilek, P., Pedrycz, W. and others [10] proposed f-COCOMO model, using fuzzy sets. The methodology of fuzzy sets giving rise to f-COCOMO is sufficiently general to be applied to other models of software cost estimation such as function point method [9]. W.Pedrycz, J.F.Peters, S. Ramanna [11] found that the concept of

information granularity and fuzzy sets, in particular, plays an important role in making software cost estimation models more user friendly.

To apply fuzzy logic first fuzzification is done using triangular fuzzy number and extension principle [4]. Fuzzy output is evaluated and then estimation is done by defuzzification technique given in this paper. Comparison of all the models, cited above, is done, in the experimental study section, and it is found that the developed models provide better estimation.

The paper is divided into five sections. Section 2 introduces related terms used to develop the new models. In section 3, the proposed models are presented, section 4 belongs to experimental study and the last section 5 is referred to conclusions and future research.

2. Related Terms

(a) Fuzzy Number

(b) Fuzzy Logic

(c) Fuzziness

(d) Fuzzy Effort Estimation

(i) Fuzzification

(ii) Defuzzification

(e) Various criteria for Assessment of Software Cost Estimation Models

(a) Fuzzy Number

A fuzzy number is a quantity whose value is imprecise, rather than exact as in the case of ordinary single valued numbers. Any fuzzy number can be thought of as a function, called membership function, whose domain is specified, usually the set of real numbers, and whose range is the span of positive numbers in the closed interval $[0, 1]$. Each numerical value of the domain is assigned a specific value and 0 represents the smallest possible value of the membership function, while the largest possible value is 1.

In many respects fuzzy numbers depict the physical world more realistically than single valued numbers. Suppose that we are driving along a highway where the speed limit is 80km/hr, we try to hold the speed at exactly 80km/hr, but our car lacks cruise control, so the speed varies from moment to moment. If we note the instantaneous speed over a period of several minutes and then plot the result in rectangular coordinates, we may get a curve that looks like one of the curves shown below, however there is no restriction on the shape of the curve. The curve in figure 1 is a triangular fuzzy number, the curve in figure 2 is a trapezoidal fuzzy number, and the curve in figure 3 is bell shaped fuzzy number.

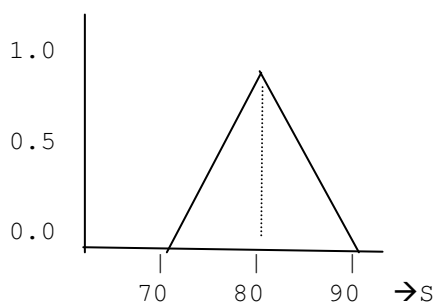


Fig 1: Triangular Fuzzy Number

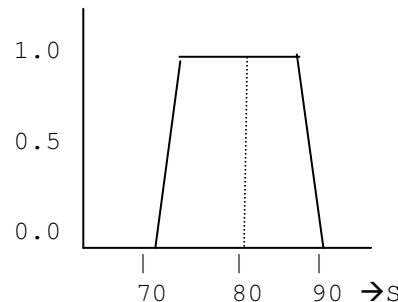


Fig 2: Trapezoidal Fuzzy Number

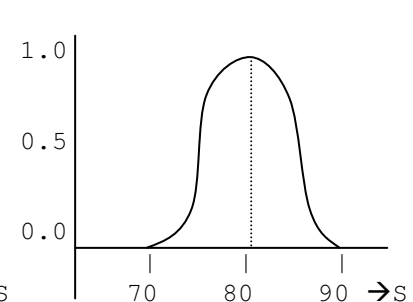


Fig 3: Bell shaped Fuzzy Number

(b) Fuzzy Logic

Fuzzy logic is a methodology, to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory [4,5]. Use of fuzzy sets in logical expression is known as fuzzy logic. A fuzzy set 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 function may be triangular, trapezoidal, parabolic etc. Fuzzy numbers are special convex and normal fuzzy sets, usually with single modal value, representing uncertain quantitative information. A triangular fuzzy number (TFN) is described by a triplet (α, m, β) , where m is the model value, α and β are the right and left boundary respectively.

Size of the project, especially in the beginning of the project, cannot be taken precisely. It can be taken as a triangular fuzzy number. Uncertainty at the input level yields uncertainty at the output level. The effort is estimated in terms of distribution described by membership function of the effort. A single fuzzy estimate of effort is calculated by defuzzification technique given in defuzzification section of the paper.

(c) Fuzziness

Fuzziness of a TFN (α, m, β) is defined as:

$$\text{Fuzziness of TFN (F)} = \frac{\beta - \alpha}{2m}, 0 < F < 1 \quad \dots\dots\dots (1)$$

The higher the value of fuzziness, the more fuzzy is TFN. The value of fuzziness to be taken depends upon the confidence of the estimator. A confident estimator can take smaller values of F.

(d) Fuzzy Effort Estimation

(i) Fuzzification:

We take the input as size S, output as effort E, then a triangular fuzzy number, $K(S)$, is defined as follows:

$$K(S) = \begin{cases} 0, & S \leq \alpha \\ \frac{S - \alpha}{m - \alpha}, & \alpha \leq S \leq m \\ \frac{\beta - S}{\beta - m}, & m \leq S \leq \beta \\ 0, & S \geq \beta \end{cases} \quad \dots\dots\dots (2)$$

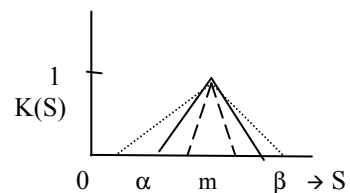


Fig 4a: Representation of $K(S)$

----- $K(S) = \text{TFN}(\alpha, m, \beta), F=0.1$

————— $K(S) = \text{TFN}(\alpha, m, \beta), F=0.2$

..... $K(S) = \text{TFN}(\alpha, m, \beta), F=0.3$

where α, m and β are the parameters of the membership function, $K(S)$, m is the model value, α and β are the right and left boundary respectively.

Let $(m, 0)$ divides, internally, the base of the triangle in ratio $k : 1$, where k is real positive number.

$$\text{So that } m = \frac{\alpha + k\beta}{k + 1} \quad \dots\dots (3)$$

As by definition of fuzziness

$$F = \frac{\beta - \alpha}{2m},$$

so

$$\alpha = \left(1 - \frac{2kF}{k+1}\right) * m \quad \dots (4)$$

and

$$\beta = \left(1 + \frac{2F}{k+1}\right) * m \quad \dots (5)$$

Similarly, the TFN $\mu(E)$ is defined as,

$$\mu(E) = \begin{cases} 0, E \leq a \alpha^b \\ \frac{(E/a)^{1/b} - \alpha}{m - \alpha}, a \alpha^b \leq E \leq a m^b \\ \frac{\beta - (E/a)^{1/b}}{\beta - m}, a m^b \leq E \leq a \beta^b \\ 0, E \geq a \beta^b \end{cases} \quad \dots (6)$$

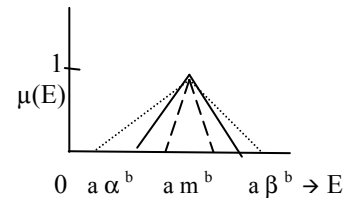


Fig: 4b: Representation of $\mu(E)$

- $\mu(E) = \text{TFN}(a \alpha^b, a m^b, a \beta^b), F=0.1$
- $\mu(E) = \text{TFN}(a \alpha^b, a m^b, a \beta^b), F=0.2$
- $\mu(E) = \text{TFN}(a \alpha^b, a m^b, a \beta^b), F=0.3$

Table 1, gives values of α and β for $F=0.1, 0.2$ and 0.3 for various values of k using equations (4) and (5), where m , size estimate in KDLOC.

Table 1: Values of α and β for different values of F and k .

F	k=1	k=2	k=3
0.1	$\alpha=0.9m$	$\alpha=(2.6/3)*m$	$\alpha=.85m$
	$\beta=1.1m$	$\beta=(3.2/3)*m$	$\beta=1.05m$
0.2	$\alpha=0.8m$	$\alpha=(2.2/3)*m$	$\alpha=.7m$
	$\beta=1.2m$	$\beta=(3.4/3)*m$	$\beta=1.1m$
0.3	$\alpha=0.7m$	$\alpha=0.6m$	$\alpha=0.55m$
	$\beta=1.3m$	$\beta=1.2m$	$\beta=1.15m$

(ii) Defuzzification:

The single output, fuzzy estimate of E , can be computed as a weighted average of the optimistic ($a\alpha^b$), most likely ($a m^b$) and pessimistic estimate ($a \beta^b$) [6]

$$\text{Fuzzy estimate of } E = \frac{w_1 (a \alpha^b) + w_2 (a m^b) + w_3 (a \beta^b)}{w_1 + w_2 + w_3} \quad \dots (7)$$

Where, w_1, w_2 and w_3 are weights of the optimistic, most likely and pessimistic estimate respectively. Maximum weight should be given to the most expected estimate.

(e) Various Criteria for Assessment of Software Cost Estimation Models

1. VAF (Variance Accounted For)

$$\text{VAF (\%)} = \left(1 - \frac{\text{var}(E - \hat{E})}{\text{var } E} \right) * 100 \quad \dots\dots\dots(8)$$

2. Mean Absolute Relative Error

$$\text{Mean absolute error (\%)} = \frac{\sum f(R_E)}{\sum f} * 100 \quad \dots\dots\dots(9)$$

3. Variance Absolute Relative Error

$$\text{Variance Absolute Relative Error (\%)} = \frac{\sum f(R_E - \text{mean } R_E)^2}{\sum f} * 100 \quad \dots\dots\dots(10)$$

4. Prediction (n)

Prediction at level n is defined as the % of projects that have absolute relative error less than n [8].

Where,

$$\text{var } x = \frac{\sum f(x - \bar{x})^2}{\sum f} \quad \dots\dots\dots(11)$$

\bar{x} = mean x

E = measured effort

\hat{E} = estimated effort

f = frequency

$$\text{Absolute Relative Error (R}_E\text{)} = \frac{|E - \hat{E}|}{|E|} \quad \dots\dots\dots(12)$$

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 Mean absolute Relative Error is better than that which gives higher Mean absolute Relative Error. A model which gives lower Variance Absolute Relative Error is better than that which gives higher Variance Absolute Relative Error.

3. Proposed Models**Model 1**

$$\text{Effort (E)} = \frac{w_1 (a \alpha^b) + w_2 (a m^b) + w_3 (a \beta^b)}{w_1 + w_2 + w_3}$$

where ,

a=3.41, b=0.795,

m represents size in KLOC,

$$\alpha = \left(1 - \frac{2kF}{k+1} \right) * m ,$$

$$\beta = \left(1 + \frac{2F}{k+1}\right) * m ,$$

k, F ,w₁,w₂ and w₃ are arbitrary constants.

The effort is obtained in Man Months (MM). Optimization of effort for an application is done by a suitable choice of arbitrary constants.

Model 2

$$\text{Effort (E)} = \frac{w_1 (a \alpha^b) + w_2 (a m^b) + w_3 (a \beta^b)}{w_1 + w_2 + w_3} + c (\text{ME}) + d$$

Where ,

a=3.41, b=0.795, m=size in KLOC,

$$\alpha = \left(1 - \frac{2kF}{k+1}\right) * m ,$$

$$\beta = \left(1 + \frac{2F}{k+1}\right) * m ,$$

k, F ,w₁,w₂ ,w₃ , c and d are arbitrary constants.

ME is methodology of the project.

The effort is obtained in Man Months. Optimization of effort for an application can be done by a suitable choice of arbitrary constants.

4. Experimental Study

For this study we have taken data of 10 projects of NASA [2], given in Table 2.

Table2: NASA software projects data

Project No	Size In KLOC	Methodology ME	Measured Effort
13	2.1	28	5
10	3.1	26	7
11	4.2	19	9
17	12.5	27	23.9
3	46.5	19	79
4	54.5	20	90.8
6	67.5	29	98.4
15	78.6	35	98.7
1	90.2	30	115.8
18	100.8	34	138.3

Let F=0.1 ,k=1, then from Table 1,

$\alpha=0.9m$ and $\beta=1.1m$

Suitable values of w_i (i=1,2,3) can be obtained by optimising effort on the basis of any one of the criteria, using algorithm given in appendix, for the application under study.

We have taken

$w_1=1$, $w_2=10$ and $w_3=10$ for Model 1. For Model 2, $w_1=1$, $w_2=10$ and $w_3=10$, when we take $c=-0.55$ and $d=13.3$

Experimental Results are given in **Table3**.

Table 3: Estimated Efforts in Man Months of Various Models

S. No.	Size in KLOC	x=Measured Effort.	Bailey –Basili Estimate	Alaa F. Sheta G.E. model Estimate	Alaa F. Sheta Model 2 Estimate	Model 1 Estimate	Model 2 Estimate
1	2.1	5	7.226224	8.440419	11.27120048	6.357633226	4.257633
2	3.1	7	8.21208	11.22081	14.45704091	8.664902435	7.664902
3	4.2	9	9.357377	14.01029	19.97636804	11.03098556	13.88099
4	12.5	23.9	19.16905	31.09857	31.68630498	26.25273718	24.70274
5	46.5	79	68.24343	81.25767	85.00702888	74.60298899	77.45299
6	54.5	90.8	80.92977	91.25759	94.97777782	84.63818656	86.93819
7	67.5	98.4	102.1752	106.7071	107.2547219	100.3292648	97.67926
8	78.6	98.7	120.8488	119.2705	118.0305042	113.2379646	107.288
9	90.2	115.8	140.8201	131.8988	134.0113916	126.3340235	123.134
10	100.8	138.3	159.4349	143.0604	144.4488097	138.0010063	132.601

Following graphic show comparison of experimental results and measured Efforts.

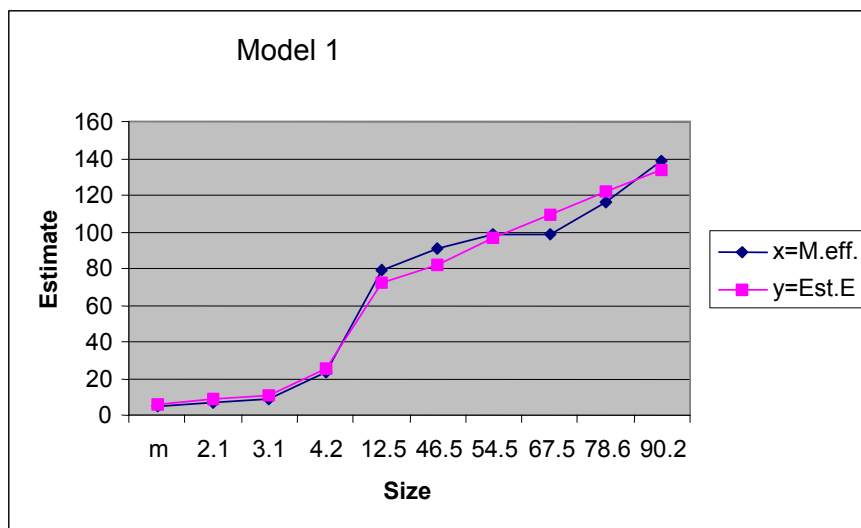


Fig 5: Effort from Model 1 versus measured effort.

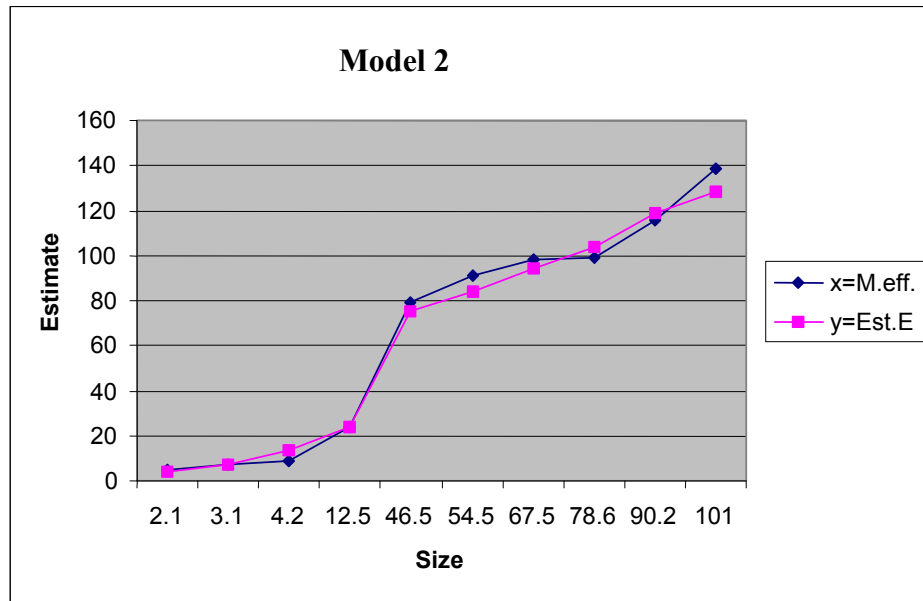


Fig 6: Effort from Model 2 versus measured effort.

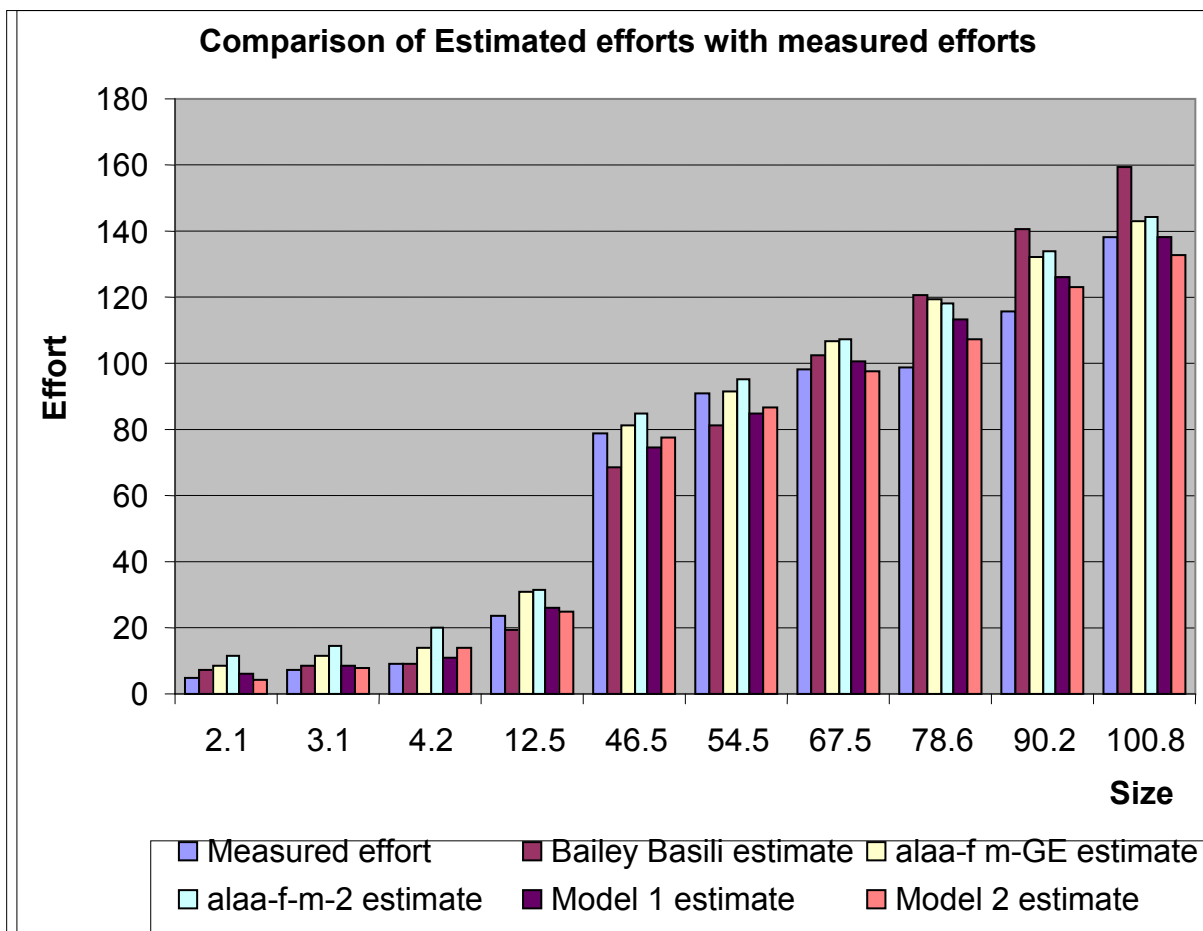


Fig 7: Effort from various models versus measured effort.

Comparison of various models on the basis of various criterion is given in Table 4.

Table 4: Comparison of various models

Model	$\left(1 - \frac{\text{var}(E - \hat{E})}{\text{var} E}\right) * 100$ VAF(%)	$\frac{\sum f(R_E)}{\sum f} * 100$ Mean Absolute Relative error (%)	$\frac{\sum f(R_E)}{\sum f}$ Variance Absolute Relative Error (%)	Pred (25)
Bailey-Basili Model	93.14782546	17.32570997	1.2102744	90
Alaa F. Sheta G E model	98.41081042	26.48847917	6.07945	60
Alaa F. Sheta , Model 2	98.92944579	44.74527991	23.8044018	60
Model 1	98.50090897	12.1702568	80.859119	90
Model 2	99.15069596	10.8036822	2.2512503	90

5. Conclusions and Future Research

The proposed study gives better results as compared to some earlier models. Proposed Model 2 is the best as per the above experimental study, on the basis of VAF, Mean Absolute Relative Error, Variance Absolute Relative error and Pred (25). It is not possible to evolve a method, which can give VAF (%) equal to 100. By suitably adjusting the values of the arbitrary constants we can optimise the estimated effort. The methodology of fuzzy sets used for developing the two new models is sufficiently general and can be applied to other models of software cost estimation models based on function point methods and to other areas of quantitative software engineering

References:

1. Alaa F. Sheta, 2006, Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects. *Journal of Computer Science* 2(2):118-123.
2. Bailey, J.W. and Basili, 1981. A Meta model for software development resource expenditure. *Proc. Intl. Conf. Software Engineering*, pp: 107-115
3. Boehm, B.,1981. *Software Engineering Economics*, Englewood Cliffs, NJ. Prentice-Hall.
4. L.A. Zadeh, 2002, From Computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions, *Int. J. Appl. Math.Comut.Sci.*,2002, Vol.12,No.3,307-324.
5. L.A. Zadeh, 1965, Fuzzy Sets, *Information and Control*, 8, 338-353.
6. Roger S. Pressman, *Software Engineering; A Practitioner Approach*, Mc Graw-Hill International Edition, Sixth Edition, 2005
7. Ali Idri , Alain Abran and Laila Kjiri (2000), COCOMO cost model using Fuzzy Logic, 7th International Conference on Fuzzy Theory & Technology Atlantic, New Jersey, March-2000.
8. Emilia Mendes, Nile Mosley Ch 8-Web Cost Estimation: An Introduction , *Web engineering : principles and techniques* , 2005
9. J.E. Matson, B.E. Barrett, J.M. Mellichamp, *Software Development Cost Estimation Using Function Points*, *IEEE Trans. on Software Engineering*,20, 4(1994), 275-287.
10. Musilek, P., Pedrycz, W., Succi, G., & Reformat, M. (2000). Software Cost Estimation with Fuzzy Models. *ACM SIGAPP Applied Computing Review*, 8(2), 24-29.

11. W.Pedrycz, J.F.Peters, 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.
12. A. J. Albrecht, Measuring application development productivity, SHARE/GUIDE IBM Application development Symposium, 1979, 83 -92
13. V. R. Basili, K. Freburger, Programming Measurement and Estimation in the Software Engineering Laboratory, Journal of System and Software, 2, 2(1981), 47-57
14. B. W. Boehm et al., Software Cost Estimation with COCOMO II, Prentice Hall, 2000.

Appendix 1**Optimisation process for weights for chosen k and Fuzziness F for VAF.**

Step 1: Assign Values to a, b, k and F

Step 2: Compute Values of α and β accordingly.

Step 3: Compute Optimistic, Pessimistic and most expected estimates for the projects under consideration and store in arrays opt[], pess[] and exp[].

Step4: mVAF= 0 ; w1=w2=w3=0;

for (I=1 ; I ≤ 10 ; I++)

for (J=1; J ≤ 10; j++)

for (K=1; k ≤ 10; k++)

{

/* wi= I, wj= J, wk =K */

a. for each project compute estimated effort as

$$E \text{ est} = (w_i * \text{opt} + w_j * \text{pess} + w_k * \text{exp}) / (w_i + w_j + w_k)$$

b. Compute VAF for the data generated,

if (mVAF < VAF)

{

mVAF=VAF;

w1= I, w2 = J, w3=k;

}

}

Step5: Optimised values of weights are given by w1, w2, and w3 for optimum value of VAF (Variance Accounted for)

Step6: End