# A comparative study of conventional effort estimation and fuzzy effort estimation based on Triangular Fuzzy Numbers

**Harish Mittal** harish.mittal@vcenggrtk.com
*Department of IT*
*Vaish College of Engineering,*
*Rohtak, 124001, India*

**Pradeep Bhatia** pk_bhatia20002@yahoo.com
*Department of Computer Science*
*G.J. University of Science & Technology*
*Hisar, 125001, India*

**Abstract**

Effective cost estimation is the most challenging activity in software development. Software cost estimation is not an exact science. However it can be transformed from a black art to a series of systematic steps that provide estimate with acceptable risk. Effort is a function of size. For estimating effort first we face sizing problem. In direct approach size is measured in lines of code (LOC). In indirect approach, size is represented as function points (FP). In this paper we use indirect approach. Fuzzy logic is used to find fuzzy functional points and then the result is defuzzified to get the functional points and hence the size estimation in person hours. Triangular fuzzy numbers are used to represent the linguistic terms in Function Point Analysis (FPA) complexity matrixes We can optimise the results for any application by varying the fuzziness of the triangular fuzzy numbers.

**Keywords:** FP, FFP, FPA, FFPA, LOC, 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, is designed to provide some mathematical equations to provide software estimation. LOC-based models are algorithm models such as [3, 13, 14, and 15]. Ali Idri and Laila Kjjri [7] proposed the use of fuzzy sets in the COCOMO, 81 models [3]. Musilek, P. and others [11] proposed f-COCOMO model, using fuzzy sets. 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 [9]. Software Functional size measurement is regarded as a key aspect in the production, calibration and use of software engineering productivity models because of its independence of technologies and of implementation decisions. W.Pedrycz and others [12] 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. Harish Mittal and Pradeep Bhatia [10] used triangular fuzzy numbers for fuzzy logic sizing. Lima, O.S.J. and Others [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.

In this paper we proposed triangular fuzzy numbers to represent the linguistic variables. The results can be optimised for the given application by varying fuzziness of the triangular fuzzy numbers. To apply fuzzy logic first fuzzification is done using triangular fuzzy number, Fuzzy output is evaluated and then estimation is done by defuzzification technique given in this paper.
The paper is divided into sections. Section 2 introduces related terms. In section 3, the technique of estimation of fuzzy functional points and optimisation technique is given; section 4 gives experimental results and section 5 gives conclusions and future research.

## 2. RELATED TERMS

(a) Fuzzy Number (b) Fuzzy Logic (c) Fuzziness

(d) Function Point Analysis (e) Various criterion 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 figure3 is bell shaped fuzzy number.

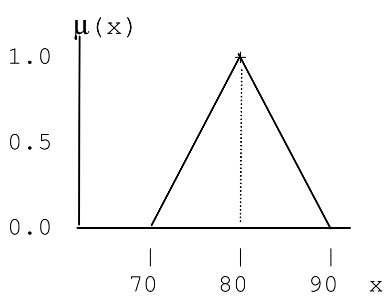


**Fig1:** Triangular Fuzzy Number

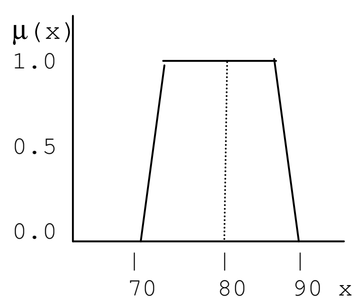


**Fig2**:Trapezoidal Fuzzy Number

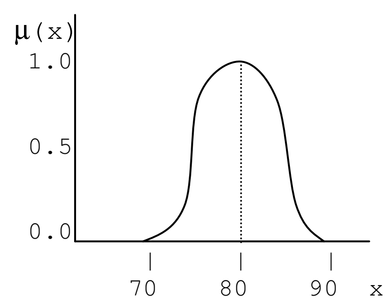


**Fig3:** Bell shaped Fuzzy Number

**(b)Fuzzy Logic**

Fuzzy logic is a methodology, to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory, and introduced in 1965 by Prof. Zadeh in the paper Fuzzy Sets [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 ($\alpha$ ,m, $\beta$), where m is the modal value, $\alpha$ and $\beta$ are the right and left boundary respectively.

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

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

The higher the value of fuzziness, the more fuzzy is TFN

**(d) Function Point Analysis (FPA):**

FPA begins with the decomposition of a project or application into its data and transactional functions. The data functions represent the functionality provided to the user by attending to their internal and external requirements in relation to the data, whereas the transactional functions describe the functionality provided to the user in relation to the processing this data by the application.

The data functions are:

1. Internal Logical File(ILF)
2. External Interface File (EIF)

The transactional functions are:

1. External Input(EI)
2. External Output(EO)
3. External Inquiry(EI)

Each function is classified according to its relative functional complexity as low, average or high. The data functions relative functional complexity is based on the number of data element types (DETs) and the number of record element types (RETs). The transactional functions are classified according to the number of file types referenced (FTRs) and the number of DETs. The number of FTRs is the sum of the number of ILFs and the number of EIFs updated or queried during an elementary process.

The actual calculation process consists of three steps:

1. Determination of unadjusted function points(UFP)
2. Calculation of value of adjustment factor(VAF)
3. Calculation of final adjusted functional points.

**Evaluation of Unadjusted FP:**

The unadjusted Functional points are evaluated in the following manner
UFP= $\Sigma\Sigma F_{ij} Z_{ij}$ , for j= 1 to 3 and i = 1 to 5, where $Z_{ij}$ denotes count for component i at level (low, average or high) j, and $F_{ij}$ is corresponding Function Points from table 1.

| **Level** | **Function Points** | | | | |
|---|---|---|---|---|---|
| | ILF | EIF | EI | EO | EQ |
| **Low** | 7 | 5 | 3 | 4 | 3 |
| **Average** | 10 | 7 | 4 | 5 | 4 |
| **High** | 15 | 10 | 6 | 7 | 6 |

**Table 1:** Translation table for the terms low, average and high

Value Adjustment Factor (VAF) is derived from the sum of the degree of influence (DI) of the 14 general system characteristics (GSCc). General System characteristics are:

1. Data communications
2. Distributed data processing
3. Performance
4. Heavily utilised configuration
5. Transaction rate
6. On-line data entry
7. End-user efficiency
8. On-line update

9. Complex processing
10. Reusability
11. Installations ease
12. Operational ease
13. Multiple sites/organisations
14. Facilitate change

The DI of each one of these characteristics ranges from 0 to 5 as follows:

(i) 0- no influence
(ii) 1 -Incidental influence
(iii) 2- Moderate influence
(iv) 3- Average influence
(v) 4- Significant influence
(vi) 5- Strong influence

Total Function Points = UFP * (0.65+ 0.01 * Value Adjustment Factor)

Function points can be converted to Effort in Person Hours. Numbers of studies have attempted to relate LOC and FP metrics [16]. The average number of source code statements per function point has been derived from historical data for numerous programming languages. Languages have been classified into different levels according to the relationship between LOC and FP. Programming language levels and Average numbers of source code statements per function point are given by [17].

Complexity matrix of an ILF or EIF is given in Table 2. Complexity matrix of EO or EQ is given in Table 3**.** Complexity matrix of EI is given in Table 4

| **RET** | **DET** | | |
|---|---|---|---|
| | 1 to 19 | 20 to 50 | 51 or more |
| **1** | Low | Low | Average |
| **2 to 5** | Low | Average | High |
| **6 or more** | Average | High | High |

**Table 2:** Complexity matrix of an ILF or EIF

| **FTR** | **DET** | | |
|---|---|---|---|
| | 1 to 5 | 6 to 19 | 20 or more |
| **Less than 2** | Low | Low | Average |
| **2 or 3** | Low | Average | High |
| **Greater than 3** | Average | High | High |

**Table 3: Complexity matrix of EO or EQ**

| **FTR** | **DET** | | |
|---|---|---|---|
| | 1 to 4 | 5 to 15 | 16 or more |
| **Less than 2** | Low | Low | Average |
| **2** | Low | Average | High |
| **More than 2** | Average | High | High |

**Table 4: Complexity matrix of EI**

The value of function points for the terms low, average and high to each FPA are given in Table1.

**(e) Various Criterions for Assessment of Software Cost Estimation Models**

There are 4 important criterions for assessment of software cost estimation models:

1. VAF (Variance Accounted For) (%):

$$\text{VAF (\%)} = \left(1 - \frac{\operatorname{var}(E - \hat{E})}{\operatorname{var} E}\right) * 100 \quad \ldots(2)$$

2. Mean absolute Relative Error (%):

$$\text{Mean absolute error (\%)} = \frac{\sum f(R_E)}{\sum f} * 100 \quad \ldots(3)$$

3. Variance Absolute Relative Error (%):

$$\text{Variance Absolute Relative Error (\%)} = \frac{\sum f(R_E - mean\ R_E)^2}{\sum f} * 100 \quad \ldots(4)$$

4. Pred (n): Prediction at level n((Pred (n))is defined as the % of projects that have absolute relative error under n[8].

   Where,

$$\text{Var x} = \frac{\sum f(x - \overline{x})^2}{\sum f} \quad \ldots(5)$$

   $\overline{x}$ = mean x

   E = measured effort

   $\hat{E}$ = estimated effort

   f = frequency

$$\text{Absolute Relative Error } (R_E) = \frac{|E - \hat{E}|}{|E|} \quad \ldots(6)$$

## 3. FUZZY FUNCTIONAL POINT ANALYSIS (FFPA)

FFPA consists of the following three stages:

1. Fuzzification
2. Defuzzification
3. Optimization

**1 Fuzzification**

We take each linguistic variables as a triangular Fuzzy numbers, TFN (α, m, β), α≤m, β≥ m. The membership function (μ(x)) for which is defined as:

$$\mu(x) = \begin{cases} 0, & x \le \alpha \\ x-\alpha / m - \alpha, & \alpha \le x \le m \\ \beta - x / \beta - m, & m \le x \le \beta \\ 0, & x \ge \beta \end{cases} \quad \ldots\ldots(7)$$

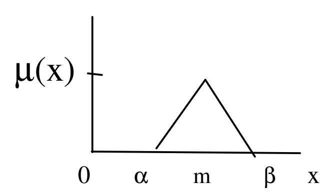


**Fig4:** representation of TFN (α, m,β)

We create a new linguistic variable, TFN (α, m, k), high or very high, where k is a positive integer. In case low and average are given, we create high variable. In case low, average and high are given, we create very high variable. In case average and high are given, we create very high variable. The creation of the new linguistic variable helps to deal better with larger systems.

Fuzziness of the created linguistic variable (F) = $(k-\alpha)/2m$, $0< F < 1$. …….. (8)

So that $k=2\ F\ m\ +\alpha$. …… (9)

We can estimate the function points for the new variable, very high, by extrapolation using Newton's interpolation formula [16]. The estimated values of function points are 22,14,9,10 and 9 for the functions ILF, EIF, EI, EO and EQ respectively.

Modified Complexity Matrices for various data and transaction functions are given in the following tables:

| DET | Complexity |
|---|---|
| 1-50 | Low |
| 51-k1<br>$k_1 \leq 102$ | Average |
| $K_1$+1 or more | High |

**Table 5**: Modified Complexity Matrix for ILF & EIF (RET =1)

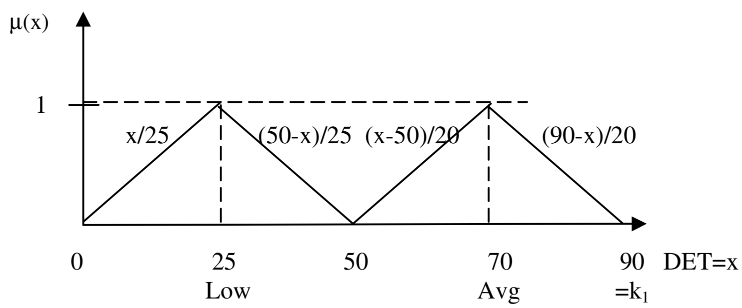


**Fig 5**

| DET | Complexity |
|---|---|
| 1-19 | Low |
| 20-50 | Average |
| 51-k2<br>$k_2 \leq 102$ | High |
| $K_2$+1 or more | V. High |

**Table 6:** Modified Complexity Matrix for ILF & EIF (RET = 2 to 5)

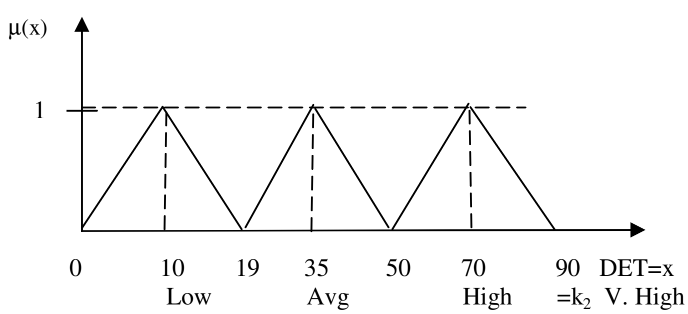


**Fig 6:**

| DET | Complexity |
|---|---|
| 1-19 | Average |
| 20-k3<br>$k_3 \leq 40$ | High |
| $K_3$+1 or more | V. High |

**Table 7:** Modified Complexity Matrix for ILF & EIF (RET $\geq 6$)

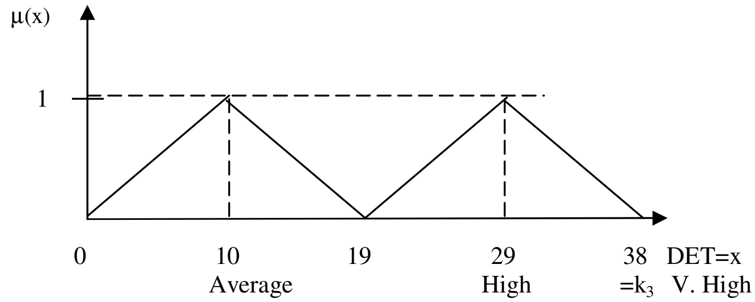


**Fig7:**

| DET | Complexity |
|---|---|
| 1-19 | Low |
| 20-k4 $k_4 \le 40$ | Average |
| $K_4$+1 or more | High |

**Table 8:** Modified Complexity Matrix for EO & EQ (FTR $\le$ 2)

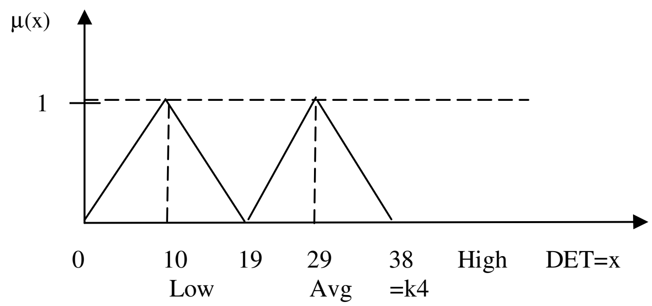


**Fig 8:**

| DET | Complexity |
|---|---|
| 1-5 | Low |
| 6-19 | Average |
| 20-k5 $k_5 \le 40$ | High |
| $K_5$+1 or more | V. High |

**Table 9:** Modified Complexity Matrix for EO & EQ (FTR = 2 or 3)

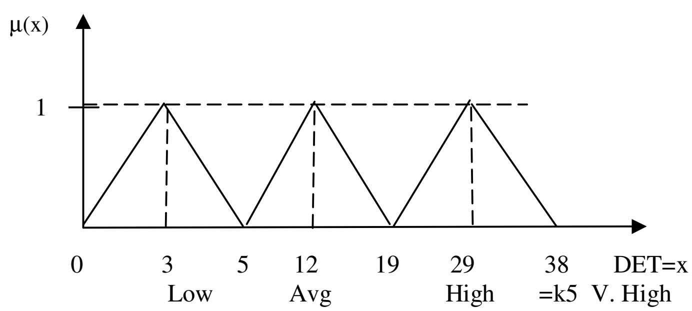


**Fig 9:**

| DET | Complexity |
|---|---|
| 1-5 | Average |
| 5-k6 $k_6 \le 12$ | High |
| $K_6$+1 or more | V.High |

**Table 10:** Modified Complexity Matrix for EO & EQ (FTR $\ge$ 4)

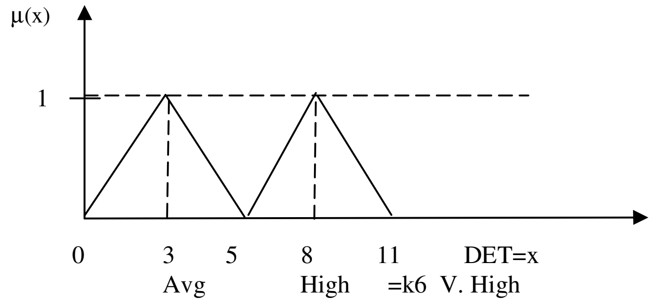


**Fig 10:**

| **DET** | **Complexity** |
|---|---|
| 1-15 | Low |
| 16-k7<br>$k_7 \le 32$ | Average |
| $K_7$+1 or more | High |

**Table 11:** Modified Complexity Matrix for EI ( FTR = 1)

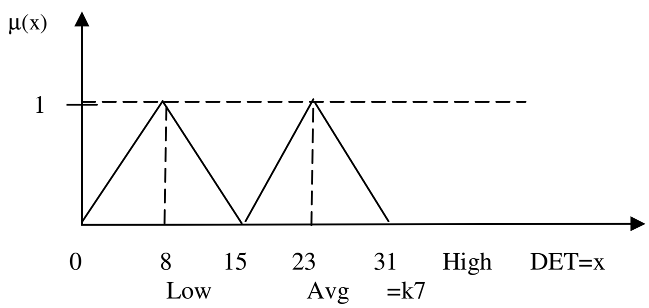


**Fig 11:**

| DET | Complexity |
|---|---|
| 1-4 | Low |
| 5-15 | Average |
| 16-k8<br>$k_8 \le 32$ | High |
| $K_8$+1 or more | V. High |

**Table 12:** Modified Complexity Matrix for EI (FTR = 2)

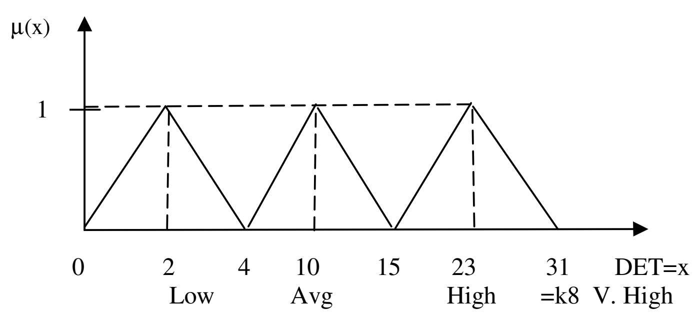


**Fig 12:**

| **DET** | **Complexity** |
|---|---|
| 1-4 | Average |
| 5-k9<br>$k_9 \le 10$ | High |
| $K_9$+1 or more | V.High |

**Table 13:** Modified Complexity Matrix for EI (FTR $\ge$ 2)

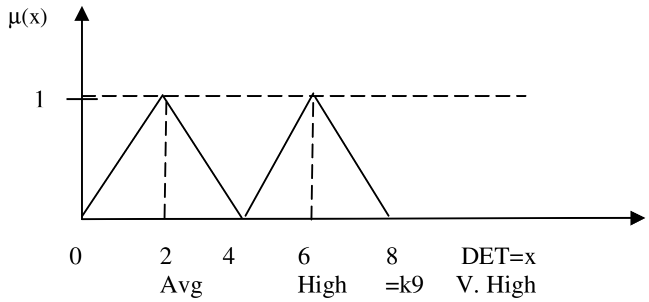


**Fig 13:**

**Defuzzification:**
Defuzzification rules for various data and transaction functions are given in the following tables.

**Defuzzification for ILF and EIF**

| DET / FFP | 1-25 | 25-50 | 50-70 | 70-90 |
|---|---|---|---|---|
| ILF | μ*7 | (μ*7)+ (1-μ)*10 | (μ*10)+ (1-μ)*7 | (μ*10)+ (1-μ)*15 |
| EIF | μ*5 | (μ*5)+ (1-μ)*7 | (μ*7)+ (1-μ)*5 | (μ*7)+ (1-μ)*10 |

**Table 14:** Case 1 for RET =1

| DET / FFP | 1-10 | 10-19 | 19-35 | 35-50 | 50-70 | 70-90 |
|---|---|---|---|---|---|---|
| ILF | μ*7 | (μ*7)+ (1-μ)*10 | (μ*10)+ (1-μ)*7 | (μ*10)+ (1-μ)*15 | (μ*15)+ (1-μ)*10 | (μ*15)+ (1-μ)*22 |
| EIF | μ*5 | (μ*5)+ (1-μ)*7 | (μ*7)+ (1-μ)*5 | (μ*7)+ (1-μ)*10 | (μ*10)+ (1-μ)*7 | (μ*10)+ (1-μ)*14 |

**Table 15:** Case 2 for $2 \leq$ RET $\leq 5$

| DET / FFP | 1-10 | 10-19 | 19-29 | 29-38 |
|---|---|---|---|---|
| ILF | μ*10 | (μ*10)+ (1-μ)*15 | (μ*15)+ (1-μ)*10 | (μ*15)+ (1-μ)*22 |
| EIF | μ*7 | (μ*7)+ (1-μ)*10 | (μ*10)+ (1-μ)*7 | (μ*10)+ (1-μ)*14 |

**Table 16:** Case 3 for RET $\geq 6$

**Defuzzification for EO and EQ:**

| DET / FFP | 1-10 | 10-19 | 19-29 | 29-38 |
|---|---|---|---|---|
| EO | μ*4 | (μ*4)+ (1-μ)*5 | (μ*5)+ (1-μ)*4 | (μ*5)+ (1-μ)*7 |
| EQ | μ*3 | (μ*3)+ (1-μ)*4 | (μ*4)+ (1-μ)*3 | (μ*4)+ (1-μ)*6 |

**Table 17: Case 1 FTR < 2**

| DET / FFP | 1-3 | 3-5 | 5-12 | 12-19 | 19-29 | 29-38 |
|---|---|---|---|---|---|---|
| EO | μ*4 | (μ*4)+ (1-μ)*5 | (μ*5)+ (1-μ)*4 | (μ*5)+ (1-μ)*7 | (μ*7)+ (1-μ)*5 | (μ*7)+ (1-μ)*10 |
| EQ | μ*3 | (μ*3)+ (1-μ)*4 | (μ*4)+ (1-μ)*3 | (μ*4)+ (1-μ)*6 | (μ*6)+ (1-μ)*4 | (μ*6)+ (1-μ)*9 |

**Table 18: Case 2 FTR = 2 or 3**

| RET / FFP | 1-3 | 3-5 | 5-8 | 8-11 |
|---|---|---|---|---|
| EO | μ*5 | (μ*5)+ (1-μ)*7 | (μ*7)+ (1-μ)*5 | (μ*7)+ (1-μ)*10 |
| EQ | μ*4 | (μ*4)+ (1-μ)*6 | (μ*6)+ (1-μ)*4 | (μ*6)+ (1-μ)*9 |

**Table 19: Case 3 FTR > 3**

**Defuzzification for EI:**

| DET / FFP | 1-8 | 8-15 | 15-23 | 23-37 |
|---|---|---|---|---|
| EI | μ*3 | (μ*3)+ (1-μ)*4 | (μ*4)+ (1-μ)*3 | (μ*4)+ (1-μ)*6 |

**Table 20: Case 1 FTR < 2**

| DET / FFP | 1-2 | 2-4 | 4-10 | 10-15 | 15-23 | 23-31 |
|---|---|---|---|---|---|---|
| EI | μ*3 | (μ*3)+ (1-μ)*4 | (μ*4)+ (1-μ)*3 | (μ*4)+ (1-μ)*6 | (μ*6)+ (1-μ)*4 | (μ*6)+ (1-μ)*9 |

**Table 21: Case 2 FTR= 2**

| RET / FFP | 1-2 | 2-4 | 4-6 | 6-8 |
|---|---|---|---|---|
| EI | μ*4 | (μ*4)+ (1-μ)*6 | (μ*6)+ (1-μ)*4 | (μ*6)+ (1-μ)*9 |

**Table 22: Case 3 FTR > 2**

**Optimisation**

Optimization of result for an application can be done on the basis any of the four criteria given in section 2, by varying one or more variables $k_1$, $k_2$, $k_3$, $k_4$, $k_5$, $k_6$, $k_7$, $k_8$ and $k_9$.

## 4. EXPERIMENTAL STUDY

The data for experimental study is taken from [18]. Calculation of Unadjusted Fuzzy Function points for real life application is given in tables 23 to 26.

| K | DET | RET | μ | Count | FFP | FP |
|---|---|---|---|---|---|---|
| $K_2$=90 | 60 | 3 | 0.5 | 2 | 17.00 | 20 |
| | 75 | 3 | 0.75 | 1 | 11.00 | 10 |
| | | | Total | 3 | 28.00 | 30 |

**Table 23: Calculation of FP and FFP for EIF**

| K | DET | FTR | μ | Count | FFP | FP |
|---|---|---|---|---|---|---|
| $K_4$=38 | 22 | 1 | 0.30 | 3 | 12.90 | 15 |
| | 10 | 2 | 0.71 | 4 | 18.86 | 20 |
| | 22 | 3 | 0.30 | 3 | 16.80 | 21 |
| | | | Total | 10 | 48.56 | 56 |

**Table 24: Calculation of FP and FFP for EO**

| K | DET | FTR | μ | Count | FFP | FP |
|---|---|---|---|---|---|---|
| $K_4$=38 | 2 | 1 | 0.20 | 1 | 0.80 | 3 |
| | 21 | 1 | 0.20 | 3 | 12.60 | 12 |
| $K_5$=38 | 1 | 2 | 0.33 | 1 | 1.33 | 3 |
| | 7 | 2 | 0.29 | 2 | 8.57 | 8 |
| $K_6$=11 | 2 | 4 | 0.67 | 2 | 5.33 | 8 |
| | | | Total | 9 | 28.64 | 34 |

**Table 25: Calculation of FP and FFP for EQ**

| K | DET | FTR | μ | Count | FFP | FP |
|---|---|---|---|---|---|---|
| $K_7$=31 | 2 | 1 | 0.25 | 3 | 2.25 | 9 |
| | 16 | 1 | 0.13 | 5 | 15.63 | 20 |
| $K_8$=31 | 4 | 2 | 0.00 | 2 | 8.00 | 6 |
| | 7 | 2 | 0.50 | 3 | 10.50 | 12 |
| | 13 | 2 | 0.40 | 6 | 21.60 | 24 |
| $K_9$=9 | 3 | 3 | 0.50 | 8 | 40.00 | 32 |
| | | | Total | 27 | 97.98 | 103 |

**Table 26: Calculation of FP and FFP for EI**

**Comparison of Function Points using conventional and Fuzzy Technique:**

| | FP | FFP |
|---|---|---|
| **ILF** | 0 | 0 |
| **EIF** | 30 | 28.00 |
| **EO** | 56 | 48.56 |
| **EQ** | 34 | 28.64 |
| **EI** | 103 | 97.98 |
| **Total UFP** | **211** | **203.17** |

**Table 27**

| | UFP | VAF | Total |
|---|---|---|---|
| **FP** | 211 | 1.13 | 238.43 |
| **FFP** | 203.17 | 1.13 | 229.58 |

**Table 28**

## 5. CONCLUSION AND FUTURE RESEARCH

The proposed study extends function point analysis to fuzzy function point analysis, using triangular fuzzy numbers. In FPA linguistic terms are used for some ranges of DET for which function points are considered to be the same. Of course they vary throughout these ranges. By using trapezoid shaped fuzzy numbers the problem is solved to some extent. We get better results then FPA by using Trapezoid shaped fuzzy numbers for linguistic terms for DETs in the border areas while for a considerable middle part of the range represented by linguistic term, the problem is not solved. In the proposed study triangular fuzzy numbers are used for linguistic terms, with the help of which we get variation of function points throughout the range represented by a linguistic term. Surely we must get better results. The methodology of fuzzy sets used for, in the proposed study, is sufficiently general and can be applied to other areas of quantitative software engineering.

## 6. REFERENCES


1. Alaa F. Sheta, “Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects”. Journal of Computer Science 2(2):118-123, 2006.
2. Bailey, J.W. and Basili, “A Meta model for software development resource expenditure”. Proc. Intl. Conf. Software Engineering, pp: 107-115, 1981.
3. Boehm, B., “Software Engineering Economics”, Englewood Cliffs, NJ. Prentice-Hall, (1981).
4. L.A. ZADEH., “From Computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions”, Int. J. Appl. Math. Computer Sci., Vol.12, No.3, 307-324., 2002.
5. L.A. ZADEH, “Fuzzy Sets, Information and Control”, 8, 338-353, 1965.
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, “COCOMO cost model using Fuzzy Logic”, $7^{th}$ International Conference on Fuzzy Theory & Technology Atlantic, New Jersy, 2000.
8. Emilia Mendes, Nile Mosley, “Web Cost Estimation: An Introduction, Web engineering: principles and techniques”, Ch 8, 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, 275-287, 1994.

10. Harish Mittal, Pradeep Bhatia, “Optimization Criterion for Effort Estimation using Fuzzy Technique”. CLEI Electronic Journal, Vol. 10 Num. 1 Pap. 2, 2007.
11. Musílek, P., Pedrycz, W., Succi, G., & Reformat, M., “Software Cost Estimation with Fuzzy Models”. ACM SIGAPP Applied Computing Review, 8(2), 24-29, 2000.
12. 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.
13. A. J. Albrecht, “Measuring application development productivity”, SHARE/GUIDE IBM Application development Symposium.
14. V. R. Basili, K. Freburger, “Programming Measurement and Estimation in the Software Engineering Laboratory”, Journal of System and Software, 2, 47-57, 1981.
15. B. W. Boehm et al., “Software Cost Estimation with COCOMO II”, Prentice Hall, (2000).
16. Lima O.S.J., Farias, P.P.M. Farias and Belchor, A.D., “A Fuzzy Model for Function Point Analysis to Development and Enhancement Project Assessments”, CLEI EJ 5 (2), 2002.
17. Jones, C., 1996, “Programming Languages Table”, Release 8.2, March
18. Chuk Yau, Raymond H.L. Tsoi, “Assessing the Fuzziness of General System Characteristics in Estimating Software Size”, IEEE, 189-193, 1994.