Particle Swarm Optimization in the fine-tuning of Fuzzy Software Cost Estimation Models

Prasad Reddy. P.V.G.D prasadreddy.vizag@gmail.com
Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, India,

Abstract:

Software cost estimation deals with the financial and strategic planning of software projects. Controlling the expensive investment of software development effectively is of paramount importance. The limitation of algorithmic effort prediction models is their inability to cope with uncertainties and imprecision surrounding software projects at the early development stage. More recently, attention has turned to a variety of machine learning methods, and soft computing in particular to predict software development effort. Fuzzy logic is one such technique which can cope with uncertainties. In the present paper, Particle Swarm Optimization Algorithm (PSOA) is presented to fine tune the fuzzy estimate for the development of software projects . The efficacy of the developed models is tested on 10 NASA software projects, 18 NASA projects and COCOMO 81 database on the basis of various criterion for assessment of software cost estimation models. Comparison of all the models is done and it is found that the developed models provide better estimation.

Keywords: Particle Swarm Optimization Algorithm (PSOA), Effort Estimation, Fuzzy Cost Estimation, software cost estimation

1. Introduction

Software cost estimation refers to the predictions of the likely amount of effort, time, and staffing levels required to build a software .Underestimating software costs can have detrimental effects on the quality of the delivered software and thus on a company's business reputation and competitiveness. Overestimation of software cost, on the other hand, can result in missed opportunities to use funds in other projects [4]. The need for reliable and accurate cost predictions in software engineering is an ongoing challenge [1]. Software cost estimation techniques can be broadly classified as algorithmic and non-algorithmic models. Algorithmic models are derived from the statistical analysis of historical project data [5], for example, Constructive Cost Model (COCOMO) [2] and Software Life Cycle Management (SLIM) [11]. Non-algorithmic techniques include Price-to-Win [2], Parkinson [2], expert judgment [5], and machine learning approaches [5]. Machine learning is used to group together a set of techniques that embody some of the facets of human mind [5], for example fuzzy systems, analogy, regression trees, rule induction neural networks and Evolutionary algorithms. Among the machine learning approaches, fuzzy systems and neural networks and Evolutionary algorithms are considered to belong to the soft computing group. The algorithmic as well as the non-algorithmic (based on expert judgment) cost estimation models, however, are not without errors. In the present paper a

fuzzy estimate is proposed. The parameters of the fuzzy estimate are tuned using the an optimization technique known as Particle swarm optimization Algorithm (PSOA) .

2. Fuzzy Logic:

One of the new methods, which have recently been used in many applications, is Fuzzy Logic Control. Fuzzy logic is one of the most useful approaches which deals with fuzziness. Fuzzy logic is a methodology, to solve problems which are too complex to be understood quantitatively, based on fuzzy set theory [13,14]. 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. A triangular fuzzy MF is described by a triplet (a, m, b), where m is the model value, a and b are the right and left boundary respectively. Handling the imprecision in input supplied for size requires that size of software project to be defined as a fuzzy number, instead of crisp number. The uncertainty at the input level of the model yields uncertainty at the output. This becomes obvious and, more importantly, bears a substantial significance in any practical endeavor. By changing the size using fuzzy set, we can model the effort that impacts the estimation accuracy. Therefore, the size is taken as an input MF and Effort is taken as output MF. The fuzzy estimate E can be computed as a weighted average Sugeno defuzzification of the input MF.

$$\mathrm{WA} = \frac{\mu(\omega_1)\times\omega_1 + \mu(\omega_2)\times\omega_2 + \mu(\omega_3)\times\omega_3}{\omega_1+\omega_2+\omega_3} \quad \text{----(1)}$$

Where $\mu(\omega_1)$, $\mu(\omega_2)$ and $\mu(\omega_3)$ represents the degree of fulfillment of each input. ω_1, ω_2 and ω_3 are the weights of the fuzzy estimate. The parameters or the weights of the Fuzzy Estimate are to be tuned properly. The parameters of the fuzzy estimate are tuned using the optimization technique known as Particle swarm optimization Algorithm (PSOA).

3. Overview of Particle Swarm Optimization Algorithm(PSOA):

PSOA is one of the optimization techniques and a kind of evolutionary computation technique[6,10]. The method has been found to be robust in solving problems featuring nonlinearity and non-differentiability, multiple optima, and high dimensionality through adaptation, which is derived from the social-psychological theory. The features of the method are as follows:

1. The method is developed from research on swarm such as fish schooling and bird flocking.
2. It is based on a simple concept. Therefore, the computation time is short and requires few memories
3. It was originally developed for nonlinear optimization problems with continuous variables. It is easily expanded to treat a problem with discrete variables.

According to the research results for birds flocking are finding food by flocking. PSO is basically developed through simulation of bird flocking in two-dimension space. The position of each agent is represented by XY axis position and also the velocity is expressed by vx (the velocity of X axis) and vy (the velocity of Y axis). Modification of the agent position is realized by the position and velocity information. Bird flocking optimizes a certain objective function. Each agent knows its best value so far (pbest) and its XY position. This information is analogy of personal experiences of each agent. Moreover, each agent knows the best value so far in the group (gbest) among pbest. This information is analogy of knowledge of how the other agents around them have performed. Namely, each agent tries to modify its position using the following information:

– The current positions (x,y),
– The current velocities (vx, vy),
– The distance between the current position and pbest
– The distance between the current position and gbest

This modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation:

$$v_i^{k+1} = wv_i^k + c_1 \mathrm{rand}_1 \times (\mathrm{pbest}_i - s_i^k) + c_2 \mathrm{rand}_2 \times (\mathrm{gbest} - s_i^k) \quad \text{----(2)}$$

Where

v_i^k - velocity of agent i at iteration k
w - weighting function
ci - weighting factor
rand - random number between 0 and 1
s_i^k - current position of agent i at iteration k
pbesti - pbest of agent i
gbest - gbest of the group

The following weighting function is usually utilized in (2).

$$w = \frac{w_{max} - w_{min}}{iter_{max}} \times iter \qquad \text{----(3)}$$

where
wmax - initial weight
wmin - final weight
itermax - maximum iteration number
iter - current iteration number

Using Eqs. (2) and (3) a certain velocity, which gradually gets close to pbest and gbest can be calculated. The current position can be modified by the following equation:

$$s_i^{k+1} = s_i^k + v_i^{k+1} \qquad \text{----(4)}$$

s_i^k current searching point
s_i^{k+1} modified searching point
v_i^k current velocity
v_i^{k+1} modified velocity

4. Proposed Models:

Case 1:

4.1 Model I based on Kilo Lines of Code (KLOC):

The COnstructive Cost Model (COCOMO) was provided by Boehm [2][3][11]. This model structure is classified based on the type of projects to be handled. They include the organic, semidetached and embedded projects. This model structure comes in the following form

Effort = α (KLOC) $^{\beta}$

This model considers the effect of lines of code only. Model I is proposed taking the fuzzified size of the software project to account for the impression in size, using triangular fuzzy sets. The estimated effort now is a fuzzy estimate obtained weighed average defuzzification in (1) as

Fuzzy Estimate E= $$\alpha\left(\frac{\omega_1 a^{\beta} + \omega_2 m^{\beta} + \omega_3 b^{\beta}}{\omega_1 + \omega_2 + \omega_3}\right) \qquad \text{----(5)}$$

where , α =3.2, ß =0.795, m represents size in KLOC, a=m and b=1.225m

4.2 Model II based on Kilo Lines of Code (KLOC) and Methodology (ME):

Model II is developed considering the effect of methodology (ME), as an element contributing to the computation of the software developed effort. It is further modified by adding a bias term ‘d’. The Model II thus takes the following form

Effort = α (KLOC) $^{\beta}$ + c (ME)+ d

The fuzzy estimated effort for the above model is

Fuzzy Estimate E= $\alpha\left(\frac{\omega_1 a^{\beta} + \omega_2 m^{\beta} + \omega_3 b^{\beta}}{\omega_1 + \omega_2 + \omega_3}\right)$ +c(ME)+d ----(6)

Where , α =3.2, ß =0.795, m=size in KLOC, ME is methodology of the project, a=m and b=1.225m ,c=-.895;d=19.9

Where ω_1, ω_2 and ω_3 are the weights of the fuzzy estimate to be tuned. These weights are tuned using the Particle Swarm optimization technique.

Case II:

The COCOMO81 database [14] consists of 63 projects data [15], out of which 28 are Embedded Mode Projects, 12 are Semi-Detached Mode Projects, and 23 are Organic Mode Projects. Thus, there is no uniformity in the selection of projects over the different modes. In carrying out our experiments, we have chosen 53 projects data out of the 63, which have their lines of code (size) to be less than 100KDSI.

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) .

The EAF term is the product of 15 Cost Drivers [5] that are listed in Table II .The multipliers of the cost drivers are Very Low, Low, Nominal, High, Very High and Extra High.

If the category values of all the 15 cost drivers are "Nominal", then EAF is equal to 1.

The 15 cost drivers are broadly classified into 4 categories [15,16].

1. Product : RELY - Required software reliability
   DATA - Data base size
   CPLX - Product complexity
2. Platform: TIME - Execution time
   STOR—main storage constraint
   VIRT—virtual machine volatility
   TURN—computer turnaround time
3. Personnel: ACAP—analyst capability
   AEXP—applications experience
   PCAP—programmer capability
   VEXP—virtual machine experience
   LEXP—language experience
4. Project : MODP—modern programming
   TOOL—use of software tools
   SCED—required development schedule

The cost drivers are as given in Table 3.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 [15].

The Effort is given by Effort= α (KLOC) $^{\beta}$ $\times \prod_{i=1}^{15} EM_i$

Table 1: 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	—

5. Experimental Study:

For this study we have taken data of 10 projects of NASA [12]. The experimental results for various models are as shown in Table 3

Table 2: Estimated Efforts in Man Months of Various Models

Size in KLOC	Measured Effort.	Alaa F. Sheta G.E [7] model Estimate	Alaa F. Sheta Model 2 Estimate	Mittal[12] Model I	Mittal Model II	Model I	Model II
2.1	5	8.44042	11.2712	6.357633	4.257633	6.15	4.1304
3.1	7	11.2208	14.45704	8.664902	7.664902	8.393	7.4914
4.2	9	14.01029	19.97637	11.03099	13.88099	10.6849	13.6602
12.5	23.9	31.09857	31.6863	26.25274	24.70274	25.4291	24.1772
46.5	79	81.25767	85.00703	74.60299	77.45299	72.2623	75.9596
54.4	90.8	91.25759	94.97778	84.63819	86.93819	81.8631	85.1229
67.5	98.4	106.7071	107.2547	100.3293	97.67926	97.1814	95.6709
78.6	98.7	119.2705	118.0305	113.238	107.288	109.6851	105.0212
90.2	115.8	131.8988	134.0114	126.334	123.134	122.3703	120.6051
100.8	138.3	143.0604	144.4488	138.001	132.601	132.5814	129.8385

Figure 1 and 2 show the comparison of estimated effort to measured effort for Model I and Model II. It is observed that by adding the effect of ME will improve the model prediction quality.

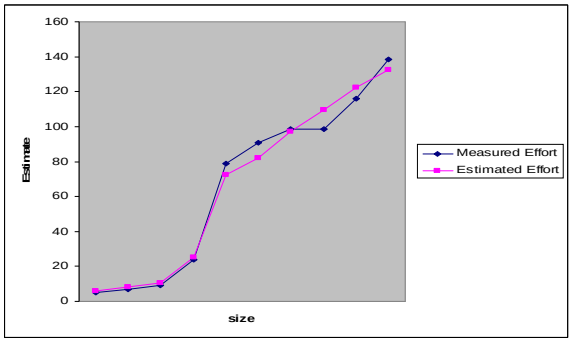

Fig 1: Effort from Model I versus measured effort for 10 NASA projects

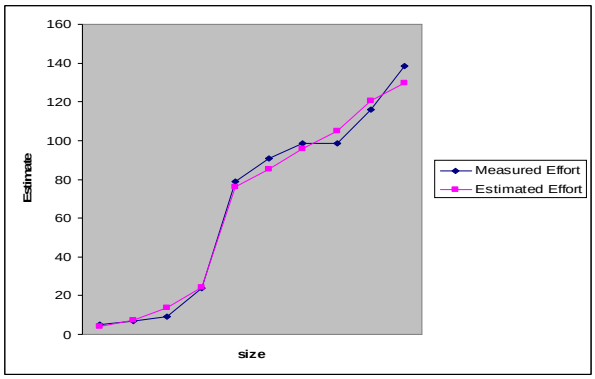

Fig 2: Effort from Model II versus measured effort for 10 NASA projects

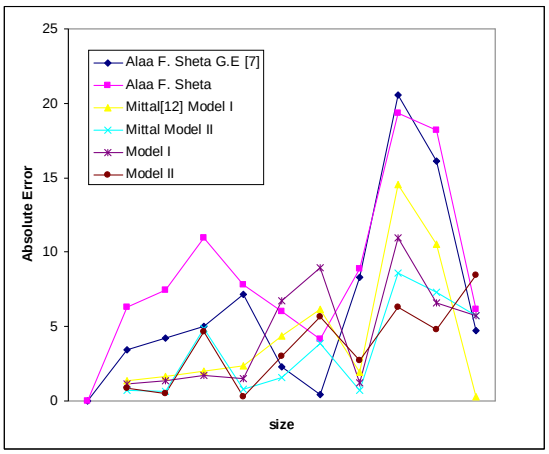

Fig 3 Comparison of Error for different Models for 10 NASA projects

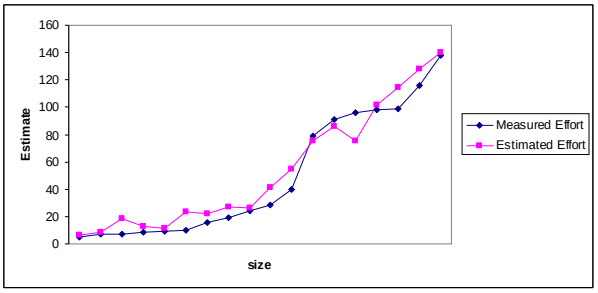

Fig 4: Effort from Model I versus measured effort for 18 NASA projects

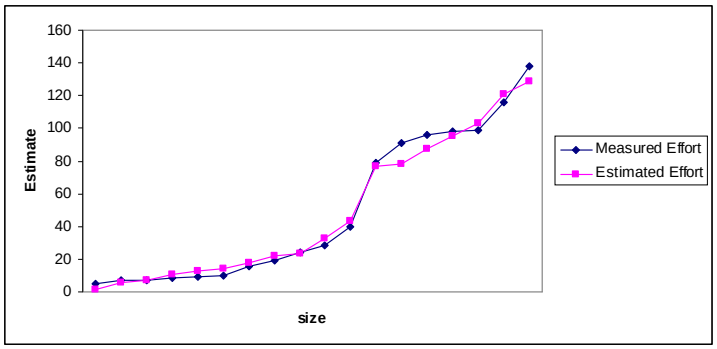

Fig 5: Effort from Model II versus measured effort for 18 NASA projects

It was also found that adding a bias term similar to the classes of regression models helps to stabilize the model by reducing the effect of noise in measurements. The efficacy of the models is tested on NASA projects . A case study based on the COCOMO81 database compares the proposed model with the Intermediate COCOMO Effort Prediction.

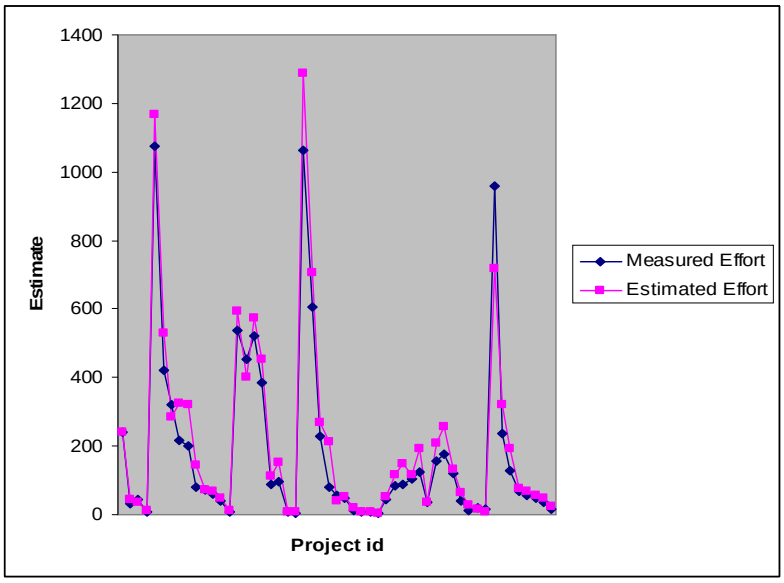

Fig 6 : COCOMO 81 model Project id versus measured effort

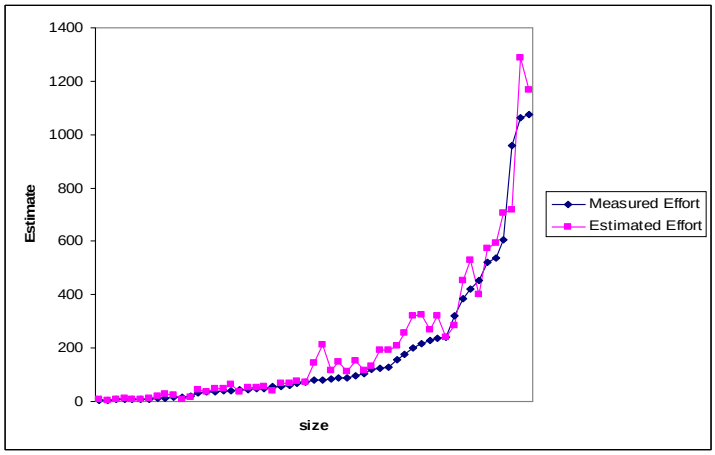

Fig 7 : COCOMO 81 model project size versus measured effort

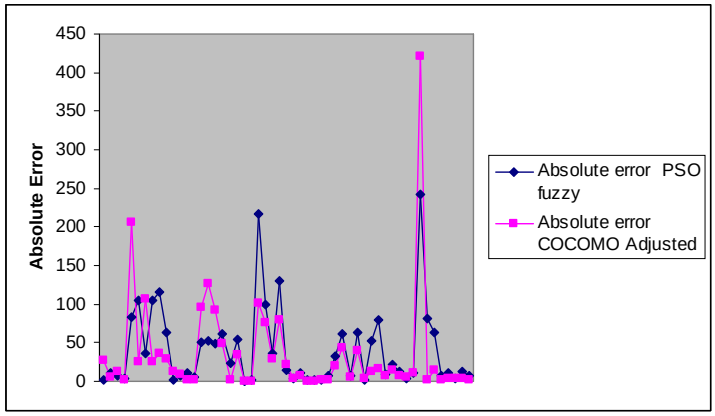

Fig 8 : COCOMO 81 model comparison of Absolute Error

Figure 3 shows a comparison of error in various models with respect to the estimated effort. Figure 4 to Figure 8 shows the comparison of estimated effort to measured effort for 18 NASA projects and COCOMO 81 dataset. Comparison of various models on the basis of various criterions is given in Figure 9 to Figure 16.

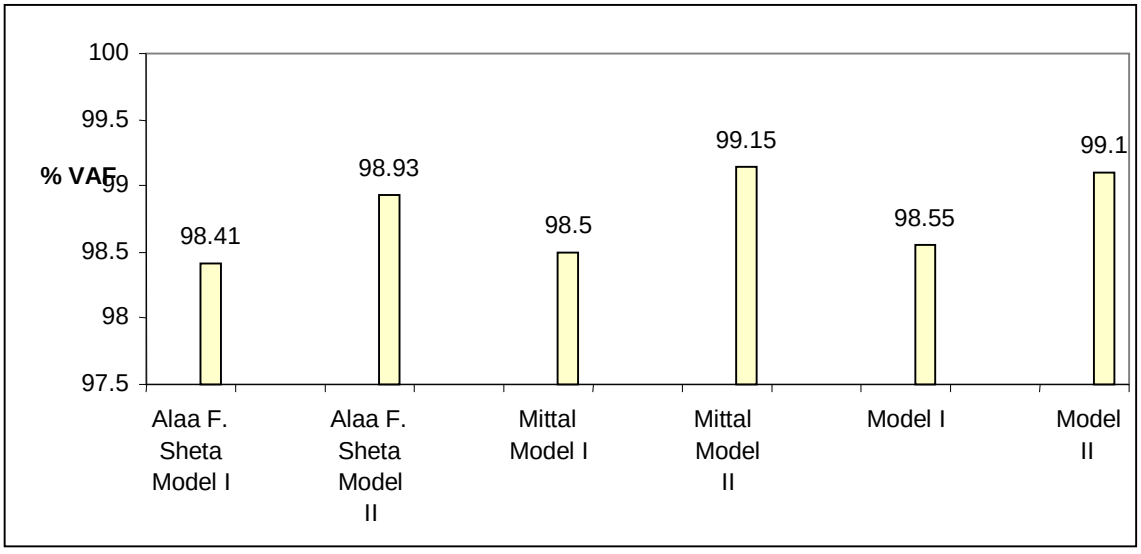

Fig 9 Comparison of % VAF for different Models for 10 NASA projects

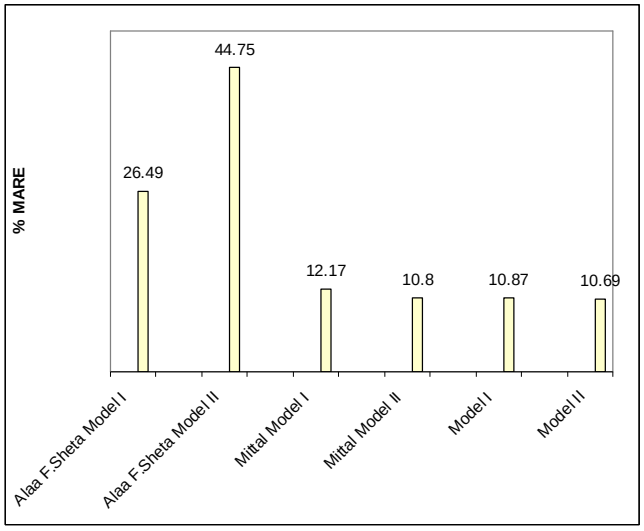

Fig 10 Comparison of % Mean Absolute Relative Error for different Models for 10 NASA projects

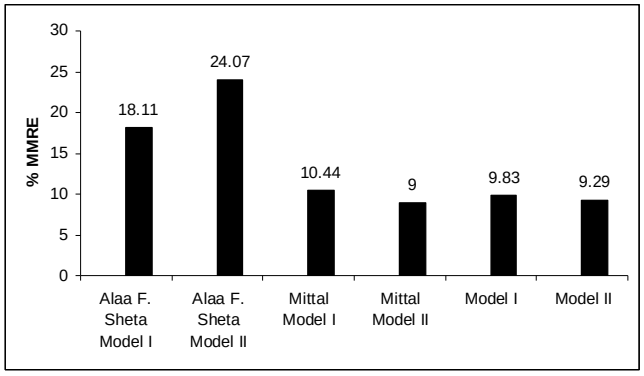

Fig 11 Comparison of % Mean Magnitude of Relative Error for different Models for 10 NASA projects

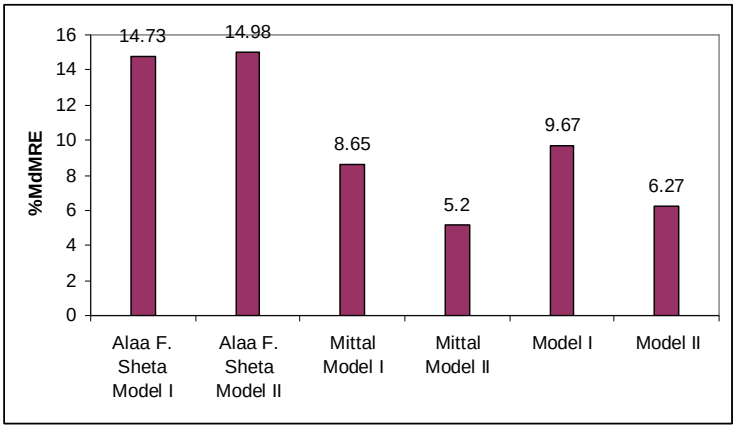

Fig 12 Comparison of % Median of Magnitude of Relative Error for different Models for 10 NASA projects

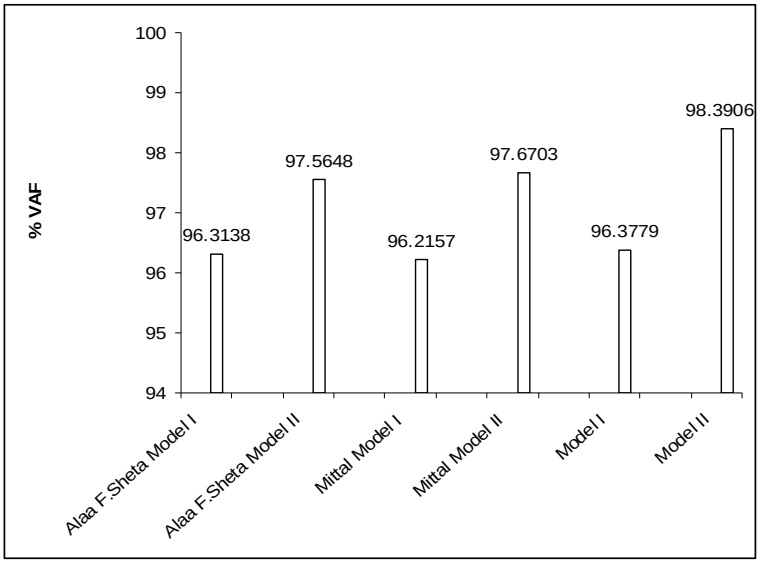

Fig 13 Comparison of % VAF for different Models for 18 NASA projects

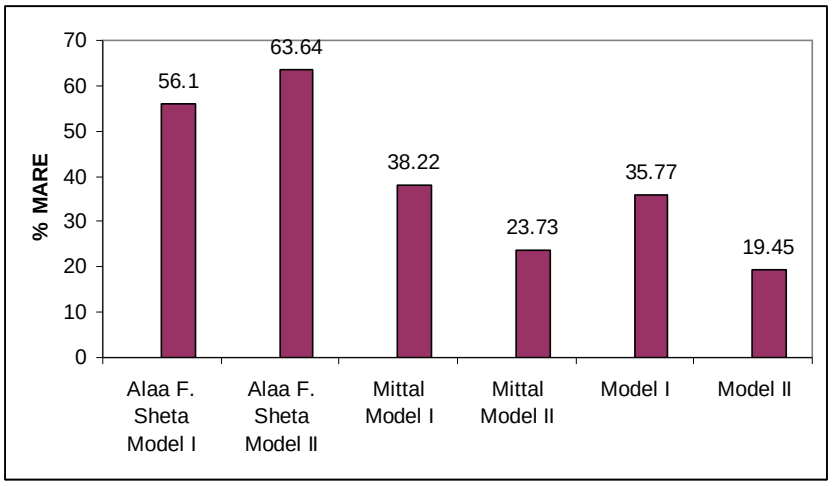

Fig 14 Comparison of % Mean Absolute Relative Error for different Models for 18 NASA projects

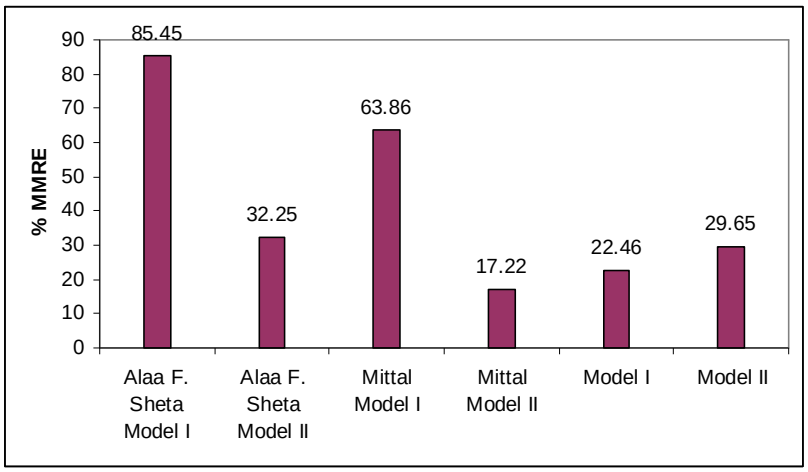

Fig 15 Comparison of % Mean Magnitude of Relative Error for different Models for 18 NASA projects

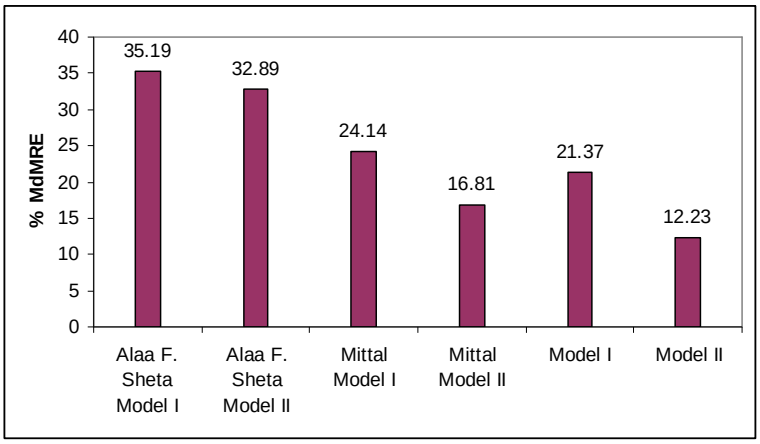

Fig 16 Comparison of % Median of Magnitude of Relative Error for different Models for 18 NASA projects

A first criterion for comparison is Variance-Accounted-For (VAF).The VAF is calculated as:
%VAF=[1 - var (Measured Effort –Estimated Effort)/ var (Measured Effort)] × 100 –(7)
The second criteria is Mean Absolute Relative error (MARE) is calculated as
%MARE=mean(abs(Measured Effort –Estimated Effort)/ (Measured Effort)) × 100 –(8)
%MMRE Mean Magnitude of Relative Error (MMRE) values. It should be less than 25% to be acceptable.

$$\%\mathrm{MMRE} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{MRE}_i \times 100$$

Where MRE(Magnitude of Relative Error)= abs(Measured Effort –Estimated Effort)/ (Measured Effort)] × 100
% MdMRE is Median of MRE values. It should be less than 25% to be acceptable.
% MdMRE for COCOMO 81 dataset is 17.02% and % MMRE for COCOMO 81 dataset is 21.15%
It is observed that the proposed models have higher % VAF, lower % MARE ,lower % MMRE and lower % MdMRE as compared to previous methods in literature. A model which gives higher VAF, lower Mean absolute Relative Error would be the best model. Hence it is obvious that the proposed models give better estimates.

6. Conclusions:

In the present paper two Fuzzy software cost estimation models based on weighed average defuzzification are considered. The weights of the models are fine tuned using Particle Swarm Optimization Algorithm. The analysis based on VAF, Mean Absolute Relative Error, Mean Magnitude of Relative Error and Median Magnitude of Relative Error show that PSOA always leads to a satisfactory result. The obtained results are superior as compared to previously reported work in the literature

7. References:

[1] Hodgkinson, A.C. and Garratt, P.W.,A Neuro-Fuzzy Cost Estimator, In (Eds.) Proc. of the 3rd International Conference on Software Engineering and Applications – SAE , 1999 pp.401-406.
[2] Boehm B. W., Software Engineering Economics, Englewood Cliffs, NJ, Prentice-Hall,1981.
[3]. B. W. Boehm et al., Software Cost Estimation with COCOMO II, Prentice Hall, (2000.)

[4] L. C. Briand, T. Langley, and I. Wieczorek, A replicated assessment and comparison of common software cost modeling techniques, In Proceedings of the 2000 International Conference on Software Engineering, Limerick, Ireland, 2000, pp.377-386.
[5] Schofield C. , Non-Algorithmic Effort Estimation Techniques, Technical Reports, Department of Computing, Bournemouth University, England, TR98-01 (1998)
[6] Suresh Chandra Satapathy, J.V.R. Murthy, P.V.G.D. Prasad Reddy, B.B. Misra, P.K. Dash and G. Panda, Particle swarm optimized multiple regression linear model for data classification Applied Soft Computing , 9, (2), (2009), Pages 470-476
[7] Alaa F. Sheta, Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects , Journal of Computer Science 2 (2)(2006) 118-123
[8]Bailey, J.W. and Basili, A Meta model for software development resource expenditure. In: Proc. Intl. Conf. Software Engineering, (1981)107-115
[9] Putnam, L. H.,A General Empirical Solution to the Macro Software Sizing and Estimating Problem, IEEE Transactions on Software Engineering, 4(4) (1978). 345 – 361
[10] E. C. Laskari, K. E. Parsopoulos and M.N. Vrahatis, Particle Swarm Optimization for Minimax Problems , Evolutionary Computation, In: (Eds.) CEC '02 Proceedings of the 2002 Congress On, 2, 2002, pp. 1576 -158.
[11] 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.
[12] Harish Mittal and Pradeep Bhatia Optimization Criteria for Effort Estimation using Fuzzy Technique CLEI ELECTRONIC JOURNAL, 10(1) (2007) pp1-11
[13] L.A. Zadeh, From Computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions, Int. J. Appl. Math. Comut.Sci, 12(3) (2002) 307-324.
[14] L.A. Zadeh, , Fuzzy Sets, Information and Control, 8, (1965) 338-353.
[15] Kirti Seth, Arun Sharma & Ashish Seth, Component Selection Efforts Estimation– a Fuzzy Logic Based Approach, IJCSS-83, Vol (3), Issue (3).
[16] Zhiwei Xu, Taghi M. Khoshgoftaar, Identification of fuzzy models of software cost estimation, Fuzzy Sets and Systems 145 (2004) 141–163