

# Fuzzy and Swarm Intelligence for Software Cost Estimation

Hari Ch.V.M.K<sup>1</sup>

<sup>1</sup> GITAM Institute of Technology

*Received: 30 October 2011 Accepted: 24 November 2011 Published: 9 December 2011*

---

## Abstract

Software cost estimation is the process of predicting the amount of time, effort and resources required to complete the project successfully. Software development is a collection of activities includes feasibility study, analysis, design, coding, testing, implementation and maintenance. Each phase requires resources people, time, software and hardware which should be predicted well before the software development. The prediction means lot of uncertainty. So far many models are proposed by using Fuzzy Logic, Neural Networks, Machine Learning, Regression analysis and Soft Computing techniques. In this paper we are proposed a new model structure basing on Alaa F. Sheta using Fuzzy logic for controlling prediction uncertainty and the parameters of the cost model tuned by using swarm intelligence-Particle Swarm Optimization. The proposed model results are verified with NASA software dataset and results are compared with the existing models. The Results show that the value of MARE (Mean Absolute Relative Error) applying fuzzy-swarm intelligence was substantially lower than MARE of other models exists in the literature.

---

**Index terms**— Software Cost Estimation, Swarm Intelligence, Fuzzy Logic, COCOMO, Particle Swarm Optimization.

## 1 INTRODUCTION

oftware project management is collection of two activities: Project Planning and Project Monitoring and control. Planning is predicting the activities that must be done before starting development work. Once project work is started it is the responsibility of project manager to monitor the work and see the goal high quality of software must be produced with low cost and within a time and budget. The input for the planning is SRS-Software Requirement Specification Document and output is project plan mainly includes Cost estimation and Schedule estimation.

Software cost estimation is the process of predicting the amount of time required to build a software system. The time is measured in terms of Person-Months (PM's) which is later on converted into dollar cost. The basic input for the cost model is size measured in terms of KDLOC (Kilo Delivered Lines Of Code) and set of Cost parameters. The advantage of cost estimation is Cost benefit analysis, proper resource utilization (software, hardware and people), staffing plans, functionality trade-offs, risks and modify budget.

The software cost estimation problem deserves a special attention because of development of product is unique under taking results in uncertainty, with increased size of software projects estimation mistakes could cost lot in terms of resources allocated to the project.

## 2 II.

## 3 BACKGROUND

In this section we briefly discuss the COCOMO (Constructive Cost Model), Fuzzy Logic and Swarm Intelligence-Particle Swarm Intelligence. a) COCOMO ??Boehm, 1981]

## 5 LITERATURE REVIEW

Where  $a$  and  $b$  are the set of values depending on the complexity of software (for organic projects  $a=2.4, b=1.05$ , for semi-detached  $a=3.0, b=1.1.2$  and for embedded  $a=3.6, b=1.2$ ).

An Intermediate COCOMO model effort  $E$  is function of program size and set of cost drivers or effort multipliers. The Effort calculated using the following equation  $\text{Effort} = a * (\text{size})^b * \text{EAF}$  (2)

where  $a$  and  $b$  are the set of values depending on the complexity of software (for organic projects  $a=3.2, b=1.05$ , for semi-detached  $a=3.0, b=1.1.2$  and for embedded  $a=2.8, b=1.2$ ) and EAF (Effort Adjustment Factor) which is calculated using 15 cost drivers. Each cost driver is rated from ordinal scale ranging from low to high.

In Detailed COCOMO the effort  $E$  is function of program size and a set of cost drivers given according to each phase of software life cycle. The phases used in detailed COCOMO are requirements planning and product design, detailed design, code and unit test, and integration testing. The weights defined accordingly. The Effort calculated using the following equation  $\text{Effort} = a * (\text{size})^b * \text{EAF} * \sum(W_i)$  (3)

Boehm and his colleagues have refined and updated COCOMO called as COCOMO II. It is a collection of three variants, Application composition model, early design model, and Post architecture model.

## 4 b) Fuzzy Logic

A fuzzy set is a set with a smooth boundary. Fuzzy set theory generalizes classical set theory to allow partial membership [5,6]. The best way to introduce fuzzy sets is to start with a limitation of classical sets. A set in classical set theory always has a sharp boundary because membership in a set is a black-and-white concept, i.e. an object either completely belongs to the set or does not belongs to the set at all. The degree of membership in a set is expressed by a number between 0 and 1; 0 means entirely not in the set, 1 means completely in the set, and a number in between means partially in the set. This way a smooth and gradual transition from the region outside the set to those in the set can be described. A fuzzy set is thus defined by a function that maps objects in a domain of concern to their membership value in the set. Such a function is called the Membership Function and usually denoted by the Greek symbol  $\mu$ . The membership function of a fuzzy set  $A$  is denoted by  $\mu_A$ , and the membership value of  $x$  in  $A$  is denoted by  $\mu_A(x)$ . The domain of membership function, which is the domain of concern from which elements of the set are drawn, is called the Universe Of Discourse. We may identify meaningful lower and upper bounds of the membership functions. Membership functions of this type are known as interval values fuzzy sets. The intervals of the membership functions are also fuzzy then it is known as interval Type-2 fuzzy sets. c) Swarm Intelligence-Particle Swarm Optimization Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates. Particle Swarm Optimization (PSO) incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the idea is emerged. PSO is a population-based optimization tool, which could be implemented and applied easily to solve various function optimization problems, or the problems that can be transformed to function optimization problems. Particle Swarm Optimization was first introduced by Dr. Russell C. Eberhart and Dr. James Kennedy in 1995. As described by Eberhart and Kennedy, the PSO algorithm is an adaptive algorithm based on a social-psychological metaphor; a population of individuals (referred to as particles) adapts by returning stochastically toward previously successful regions. The basic concept of PSO lies in accelerating each particle towards its Pbest and Gbest locations with a random weighted acceleration at each time. The modification mbns of the particles positions can be mathematically modeled according to the following equations: 
$$V_{ik+1} = w * V_{ik} + c_1 * \text{rand}() * (P_{best} - S_{ik}) + c_2 * \text{rand}() * (G_{best} - S_{ik})$$
 (4) 
$$S_{ik+1} = S_{ik} + V_{ik+1}$$
 (5)

Where,  $S_{ik}$  is current search point,  $S_{ik+1}$  is modified search point,  $V_{ik}$  is the current velocity,  $V_{k+1}$  is the modified velocity,  $P_{best}$  is the velocity based on Pbest,  $G_{best}$  = velocity based on Gbest,  $w$  is the inertia weight,  $c_1$  is the weighting factors,  $\text{rand}()$  are uniformly distributed random numbers between 0 and 1. In order to guide the particle effectively in the search space, the maximum moving distance during each iteration must be changed in between the maximum velocity  $[-V_{max}, V_{max}]$ .

III.

## 5 LITERATURE REVIEW

In this section we discuss the some previous models proposed using Genetic Algorithms [8], Fuzzy models [9],

Soft-Computing Techniques [10], Computational Intelligence Techniques [3], Heuristic Algorithms, Neural Networks [7], Radial Basis [11], and Regression [1, ??,4].

The Cost of product is a function of many parameters which are Size (coding size), Cost Drivers and Methodology used in the project. The Walston Felix uses 36 cost drivers, 16 by Boheam and 30 other factors considered by the Bailey-Basili for the cost estimation. The parameters are estimated by using regression analysis and the effort equation is [1]  $E = 5.5 + 0.73(KLOC)^{1.16}$  (6)

Where  $E$  is effort and KLOC is kilo lines of code-coding size

The Alaa F. )

Where ME is the methodology used in the project.

---

The Harish proposed two model structures based on triangular fuzzy sets [13]. Interval Type-2 fuzzy logic, Particle Swarm Optimization for is proposed by Prasad Reddy. [12].  
IV.

## 6 PROPOSED METHODOLOGY AND ALGORITHM a) Methodology

The uncertainty about cost estimation is usually quite high, because of prediction of basic element size, cost drivers and other parameters. By introducing some modifications in the interval type-2 fuzzy logic we can control the uncertainty. In the present work fuzzy sets are used for modeling uncertainty and imprecision in an efficient way. The inputs of the standard cost model include an estimation of project size and evaluation of the parameters, rather than a single number, the software size can be regarded as a fuzzy set yielding the cost estimate also in the form of a fuzzy set. We emphasize a way of propagation of uncertainty and ensuring violation of the resulting effort. Fuzzy sets create a more flexible, high versatile development environment. They generate a feedback also the resulting uncertainty of the results. The decision-maker is no longer left with a single variable estimate which could be highly misleading in many cases and lead to the belief as a to the relevance of the obtained results.

In the present work on the proposed models the parameter tuning is done by using Particle Swarm Optimization. For each particle position with values of tuning parameters, fitness function is evaluated with an objective to minimize the fitness function. The objective is to minimize or maximize fitness function. The particles moving towards optimal parameters by doing several iterations until particles exhaust or derivative of velocity becomes nearly zero then we get the optimal parameters which are later used for effort estimation.

## 7 b) Proposed Model

The Proposed model consists of three major components. First component is fuzzification process which identifies the suitable firing intervals for the input parameter size. Second component is parameter tuning using particle swam optimization. Finally effort estimation done through weighted average defuzzification method using the results obtained in first and second steps.

### 8 i. Fuzzification process

The input size is fuzzified by using two triangular fuzzy sets. The Triangular member function is shown below.

The parameters a, b, and c of the equation 10 are tuned by using particle swarm optimization with inertia weight and MARE as the fitness function (minimize).

### 9 iii. Defuzzification

The defuzzification is done through weighted average method is as shown below  $E = \frac{w_1 * [(a * ? b) + c * (ME) + d] + w_2 * [(a * m b) + c * (ME) + d] + w_3 * [(a * ? b) + c * (ME) + d]}{w_1 + w_2 + w_3}$  (11)

Where w i is the weighting factor and ?, m, and ? are the fuzzified sizes obtained from triangular member function.

V.

## 10 RESULTS AND DISCUSSIONS

NASA dataset is considered for experimentation. The firing intervals obtained after the fuzzification are [0.7362, 0.8998]. The parameters obtained after tuning PSO methodology a=3.131606, b=0.820175, c=0.045208 and d=-2.020790. While performing defuzzification w1=1, w2=10 and w3=10. The following table 1 shows the efforts the proposed model. The estimated efforts are very close to the measured efforts.

The proposed model results are compared with the existing models in the literature and the results are shown in the following table 2.

nThe performance measure considered here is Mean Absolute Relative Error (MARE)

### 11 % MARE = mean

### 12 Effort

### 13 Size

### 14 Measured Effort vs Estimated Effort

### 15 Measured Effort Estimated Effort

The MARE of various models is shown below.

The Results show that the value of MARE (Mean Absolute Relative Error) applying fuzzy-swarm intelligence was substantially lower than MARE of other models exists in the literature.

## 16 VI.

## 17 CONCLUSION

Software cost estimation is based on a probabilistic model and hence it does not generate exact values. However availability of good historical data coupled with a systematic technique can generate better results. In this paper we proposed new model structure to estimate the software cost (Effort) estimation. Fuzzy sets is used for modeling uncertainty and impression to better the effort estimation and particle swarm optimization for tuning parameters. It is observed from the results that Fuzzy-Swarm intelligence gives accurate results when juxtaposed with its other counterparts. On testing the performance of the model in terms of the MARE the results were found to be useful.

## 18 December



Figure 1: ( 9 )

The Basic COCOMO Model computes effort E as function of program size, and it is same as single variable method. The Effort calculated using the following equation  

$$\text{Effort} = a * (\text{size})^b$$

Figure 2:

$$\text{Effort} = 3.1938(\text{DLOC})^{0.8209} - 0.1918(\text{ME}) \text{ for model 1} \quad (7)$$

$$\text{Effort} = 3.3602 (\text{DLOC})^{0.8116} - 0.4524(\text{ME}) + 17.8025 \text{ for model 2} \quad (8)$$

Figure 3:

1

		Model		Mean Absolute Relative Error(MARE%)		
Bailey -Basili Estimate				17.325		
Alaa F. Sheta G.E.Model Estimate				26.488		
Alaa F. Sheta Model 2 Estimate				44.745		
Harish model1				12.17		
Harish model2				10.803		
Proposed Model				6.947316		
		Mean Absolute Relative Error(MARE%)				
50						
45	40 35	44.745 Mean Absolute Relative Error(MARE%)				
10	15 20 25 30 MARE	17.325	26.488	12.17	10.803 6.947316	
5						
0						
		Bailey	Alaa	Alaa	Harish Harish	Proposed
		-Basili	F.	F.		
			Sheta	Sheta		
		Estimate	G.E.Model	Model	model1 model2	Model
			2			
			Estimate	Estimate		
			Model			

Figure 4: Table 1 :

2

Figure 5: Table 2 :



- 
- [Bailey and Basili ()] 'A meta model for software development resource expenditures'. John W Bailey , Victor R Basili . 1627- 9/81/0000/0107500.75@IEEE. *proceedings of the Fifth International Conference on Software Engineering*, C 2 Chris, F Kemerer (ed.) (the Fifth International Conference on Software Engineering) 1981. 1981. 1987. May 1987. 30 p. . (An Empirical Validation of Software Cost Estimation Models)
- [Boehm et al. ()] Barry Boehm , Chris Abts , Sunita Chulani . *Software Development Cost Estimation Approaches -A Survey1,IBM*, 1998. 1998. p. .
- [Sheta et al. ()] *Development of Software Effort and Schedule Estimation Models Using Soft Computing Techniques*, Alaa Sheta , David Rine , Aladdin Ayesh . 978-1-4244-1823- 7/08. 2008. 2008. IEEE. p. .
- [Sheta ()] 'Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects'. Alaa F Sheta . *Journal of Computer Science* 2006. 2006. 2 (2) p. .
- [Jørgensen (2005)] 'Evidence-Based Guidelines for Assessment of Software Development Cost Uncertainty'. Magne Jørgensen . *IEEE Transactions On Software Engineering* 2005. November 2005. 31 (11) p. .
- [Reddy et al. ()] 'Fuzzy Based Approach for Predicting Software Development Effort'. Prasad Reddy P, Ramesh .S.N.S.V.S.C. , P V G D Sudha P, Ramesh .S.N.S.V.S.C. , K R Rama Sree P, Ramesh .S.N.S.V.S.C. . *International Journal of Software Engineering (IJSE)* 2010. 2010. 1 (1) p. .
- [Huang et al. ()] 'Improving the COCOMO model using a neuro-fuzzy approach'. Xishi Huang , Danny Ho , Jing Ren , F Luiz , Capretz . DOI:10.1016 /j.asoc. 2005.06.007. *Applied Soft Computing* 2005. 2007. 2005. Elsevier. 7 p. .
- [Krishnamurthy and Fisha (1995)] 'Machine Learning Approaches to estimating software development effort'. S Krishnamurthy , Douglas Fisha . *IEEE Transactions On Software Engineering* 1995. February 1995. 21 (2) p. .
- [Mittal and Bhatia (2007)] 'Optimization Criteria for Effort Estimation using Fuzzy Technique'. Harish Mittal , Pradeep Bhatia . *CLEI Electronic Journal* 2007. June 2007. 10 (1) p. .
- [Reddy ()] 'Particle Swarm Optimization In The Fine Tuning Of Fuzzy Software Cost Estimation Models'. Prasad Reddy , PV G . *International Journal of Software Engineering (IJSE)* 2010. 2010. 1 (2) p. .
- [Sheta (2006)] 'Software Effort Estimation and Stock Market Prediction Using Takagi-Sugeno Fuzzy Models'. Alaa Sheta . *Proceedings of 2006 IEEE International Conference on Fuzzy Systems*, (2006 IEEE International Conference on Fuzzy Systems) 2006. 0-7803-9489-5/06/IEEE 2006. July 16-21, 2006. p. .
- [Ying ()] 'The Takagi-Sugeno fuzzy controllers using the simplified linear control rules are nonlinear variable gain controllers'. Hao Ying . *Automatica* 1998. 1998. ELSEVIER. 34 (2) p. .