International Journal of Technical Research and Applications e-ISSN: 2320-8163, www.ijtra.com Volume 3, Issue5 (September-October 2015), PP. 136-141

Development of Effort Estimation Model using Simulated Annealing Technique

Sheenu Rizvi[1], Dr. S.Q. Abbas[2], Dr. Rizwan Beg[3]

[1]Department of Computer Science, Amity University, Lucknow, India
[2]Ambalika Institute of Management and Technology, Lucknow, India
[3]Integral University, Lucknow,India
[1]sheenu_r@yahoo.com
[2]qamar_abbas@yahoo.com
[3]rizwanbeg@gmail.com

Abstract-The present study deals with the optimization of the COCOMO Model parameters using Simulated Annealing algorithm so as to minimize the variance in software development effort. For this data is taken from the NASA projects. The data set that has been used consists of two independent variables, viz. Lines of Code (LOC) and Effort Adjustment Factor (EAF) and dependent variable as Development Effort (DE). The results so obtained have been compared with the earlier work done by the author on Artificial Neural Network (ANN) and Adaptive Neuro Fuzzy Inference System (ANFIS), along with COCOMO results. The developed SA based model was able to provide better estimation capabilities than COCOMO, ANN and ANFIS.

Index Terms:- COCOMO, NASA, ANN, ANFIS, SA

Nomenclature

COCOMO – Constructive Cost Model

NASA – National Aeronautical Space Agency

ANN - Artificial Neural Network

ANFIS-Adaptive Neuro Fuzzy Inference System

SA – Simulated Annealing

LOC – Lines of Code

EAF – Effort Adjustment Factor

DE - Development Effort

MMRE – Mean Magnitude of Relative Error

RMSE – Root Mean Square Error

1. INTRODUCTION

Quality software project development with reasonable economy is the prime objective of each software development company. Planned estimation has long been and continues to be a major complexity in managing software development projects. Inability to cater to the quality expectations of the client leads to the failure of the software project. Hence it is prime requisite for a project manager to know the efforts, agenda and functionalities of a project before hand. However, during the project life cycle there may be change in the project factors. One may not be able to calculate these values in advance and expect it to be correct. This however calls for a better estimation/prediction techniques, which will lead to good results and better plans. Software estimation is the art of forecasting the period and rate of a project. It is a intricate process with error built into its very structure, but it is very satisfying when done the correct way. The estimation process does not end until the project is over. This is the reply of the project supervisor to the changing environment of the project. Precise estimate is a decisive part of the base work of a well-organized software project.

Here a Simulated Annealing approach as an optimization algorithm has been used for tuning the COCOMO model parameters such that a better effort estimate can be predicted. The performance of the developed model has been tested on NASA software project dataset provided in [1] and compared to the models presented in [11][12]. The developed models were able to provide good estimation capabilities compared to other models provided in the literature [3][11][12].

A. Literature Review

Many software effort estimation models have been developed over the last decades. Jovan Zivadinovic, et. al. presented the most relevant methods and models for effort estimation used by software engineers in the past four decades. Classification of the methods has been also suggested as well as brief description of the estimation methods presented. [3] provides a general overview of software estimation models and techniques. Models can be categorized as Size-Based, Function-Based, Learning-Based and Expertise-Based. [4] in his study, developed an model of effort estimation for a bank to forecast the effort before the project's development cycle. The outcome authenticates the merits of using AI methods in actual life situations. [5] developed a hybrid model by combining C-Means clustering, neural network and analogy technique. As one knows that there are complex and non linear associations amongst software project types, one can develop such proposed estimations for obtaining better results. The inferences demonstrated that fuzzy clustering could reduce the negative effect of not so relevant projects on estimation accuracy. [6] designed an artificial neural networks based software effort estimation models. The models were intended to better the performance of the network that suit the COCOMO Model. Artificial Neural Network models were developed using Radial Basis and Generalized Regression. [7] presented an overview of the different techniques currently available for software effort estimation in the software industry. Software effort estimation is a incredibly essential task in the software engineering field because the future of the project depends on the estimation report. The techniques discussed about algorithmic model, nonalgorithmic model and some soft computing technique. [8].described an enhanced Fuzzy Logic model for the estimation of software development effort and proposed a new approach by applying Fuzzy Logic for software effort estimates.

Though many researchers contributed to the literature on effort estimation, still the difficulty of effort estimation is an open challenge. Many effort estimation techniques exist in the literature, but their utilization is very particular to the development environment. So, one cannot say a specific technique is best fit for all the situations to give an accurate estimation.

B. Data Used

The data used is NASA project data. The data used as input and output variables for optimum model development are given in the Table 1. In all two input variables have been used which include fifteen effort multipliers and the SIZE measured in thousand delivered source instructions (KLOC). The output of the model is the Development Effort (DE), which is measured in man-months. The data were collected from the analysis of sixty three (63) software projects, as published by Barry Boehm in 1981[1].

Table 1. Input and Output variables for ANFIS model

Input Variables	SIZE — in KLOC EAF – Effort Adjustment Factor
Output Variable	Development Effort (DE)

1) Simulated Annealing Modelling

Simulated annealing (SA) is a nontraditional optimization technique based on a random search process. SA resembles the cooling process of molten metal through annealing (slow cooling). At higher temperatures, the molten metal atoms can move freely but at reduced temperatures, the atoms form a crystalline structure having a minimum energy configuration. However, the cooling rate governs the formation of crystals. At very fast cooling rates, the material remains in non-crystalline or amorphous form. The SA algorithm simulates this process of slow cooling to achieve the minimization of a function value. The cooling phenomenon is simulated by controlling a temperature parameter introduced with the concept of Boltzmann probability distribution [10]. This algorithm is based on the Boltzmann probability distribution by which the probability of energy distribution is given by –

The complete details of the SA algorithm are given in [10].

The modeling study is based on the Intermediate COCOMO model, which is based on the relationship ,

Development Effort (DE) = EAF*a*(size)^b,

where size is measured in thousands delivered lines of code (KLOC). The constants 'a' and 'b' are dependent upon the mode of development of projects. DE is measured in man-months.

2) Effort Estimation Model Used

The following software effort model is used in the present study:

DE = f (KLOC, EAF),

where DE is effort, KLOC is the thousands lines of the developed code and EAF is Effort Adjustment Factor used in the software project. f is a nonlinear function in terms of KLOC and EAF.

The function f is expressed as follows;

DE = EAF*a*(size)^b, (1)

where 'a' and 'b' have values given by COCOMO model as follows.

Table 2 : COCOMO Model Parameter values

Development Mode	Value of 'a'	Value of 'b'
Organic	**3.2**	**1.05**
Semi-Detached	**3.0**	**1.12**
Embedded	**2.8**	**1.2**

In the present work an effort has been made to develop a new model on the lines given by Intermediate COCOMO using simulated annealing optimization technique, wherein the constant values, 'a' and 'b' are optimized so as to lead to a better solution method. Here separate models have been developed for all the three development modes viz. Organic, Semi-Detached and Embedded, along with a single model combining all the three modes. An effort is made to make the computed values of the development effort very close to the measured value, leading to a very low Magnitude of Relative Error (MRE). The degree to which a model's estimated effort matches the actual or target effort is estimated by a % relative error.

SA modeling for effort estimation has been carried out using the Matlab software (Version 2012a). Matlab codes were developed for solving multivariable minimization problem using optimization method "simulannealbnd – Simulated Annealing algorithm" solver.

3) Algorithm Implementation and results

For the present problem, of precise modeling of SD effort the objective function used is ***Minimize Abs. (∑(Emeasured – Ecomputed))*** for tuning of COCOMO model parameters. Where Emeas, is measured value of effort, Ecomp is computed value of effort as per the model used. In order to minimize the total squared error given above, simulated annealing algorithm is used changing the parameter values of the model. The code for the Objective function used is written as M-file in M-File editor and is recalled in the Matlab command window. The lower and upper bounds of the two variables 'a' and 'b' as specified in the effort estimation model are fixed based on the values used in the COCOMO empirical model as given in Table 2 above. The SA solver inputs used for modeling are given in Table 3 below.

Table 3:- SA Options used for Optimization

Parameter (X) related Options	**Parameter update Function**	**e^{-X} ,random X, 1/X { E , R ,I }**
	Initial Parameter Value	**100**
	Reanneal Function	**100**
Algorithm Settings	**Annealing Function**	**Random X, 1/X { R , I }**
	Acceptance Function	**M.S.E**
Stopping Criteria	**Tolerence Function**	**1.00E-06**
	Maximum Iteration	**Inf**
	Max. Function Evaluation	**3000*number of variables**

C. Results and Discussions

After multiple runs for the optimization of the objective function using MATLAB command, the optimized function value and the optimal parameter values are obtained. The different cases using various combinations of SA algorithm functions and Parameter update functions are analyzed and the corresponding solutions obtained are as follows:

Table 4:- 'a' & 'b' Optimized Values for different Development Mode using Parameter Update and Annealing Function combinations

Sl. No.	**SA Parameters**		**Notation**	**Development Mode**					
	Parameter Update . Fcn.	**Annealing Fcn**		**Semi-Detached (SD)**		**Embedded (EM)**		**Organic (OR)**	
				A	**b**	**a**	**b**	**a**	**b**
1	E	R	ER	2.2306	1.192	3.5581	1.1167	**3.399**	**0.9312**
2	E	I	EI	3.0946	1.1347	3.1561	1.1386	3.631	0.9143
3	R	R	RR	1	1.3324	3.4448	1.1227	3.9813	0.8757
4	R	I	RI	3.547	1.1103	3.2038	1.1362	3.4873	0.908
5	I	R	IR	2.484	1.1726	3.5548	1.1275	3.9063	0.8957
6	I	I	II	**3.5844**	**1.109**	**3.8633**	**1.1093**	3.4832	0.9069

Based on the parameter values "a" and "b" obtained from the optimized models, the various SA model equations for various combinations of Parameter update function and Annealing functions are given in Table 5 below.

Table 5:- DE Equations for different development modes

Development Mode	Model Type	Parameter update. fcn.	Annl.fcn.	Development Effort Equation using SA
Semi-Detached	SD_ER	E	R	DE=EAF*2.2306(size)^1.192
	SD_EI	E	I	DE=EAF*3.09466(size)^1.1347
	SD_RR	R	R	DE=EAF*1.0(size)^1.3324
	SD_RI	R	I	DE=EAF*3.547(size)^1.1103
	SD_II	**I**	**I**	**DE=EAF*3.5844(size)^1.109**
	SD_IR	I	R	DE=EAF*2.484(size)^1.1726
Embedded	EM_ER	E	R	DE=EAF*3.5581(size)^1.1167
	EM_EI	E	I	DE=EAF*3.1561(size)^1.1386
	EM_RR	R	R	DE=EAF*3.4448(size)^1.1227
	EM_RI	R	I	DE=EAF*3.2038(size)^1.1362
	EM_II	**I**	**I**	**DE=EAF*3.8633(size)^1.1093**
	EM_IR	I	R	DE=EAF*3.5548(size)^1.1275
Organic	OR_EI	E	I	DE=EAF*3.631(size)^0.9143
	OR_RR	R	R	DE=EAF*3.9813(size)^0.8757
	OR_RI	R	I	DE=EAF*3.4873(size)^0.908
	OR_IR	I	R	DE=EAF*3.9063(size)^0.8957
	OR_II	I	I	DE=EAF*3.4832(size)^0.9069
	OR_ER	**E**	**R**	**DE=EAF*3.399(size)^0.9312**

From the perusal of the above tables it is seen that the best developed optimized models for all the three development modes are

For semi-detached mode,

$DE = EAF*3.5844*(DLOC)^{1.109}$ (2)

For Embedded Mode,

$DE = EAF*3.8633*(DLOC)^{1.1093}$ (3)

For Organic Mode (OR_ER),

$DE = EAF*3.399*(DLOC)^{0.9312}$ (4)

Further, based on the above models RMSE and MMRE values for all the three development modes together with COCOMO model is given in Table 6 below.

Table 6:- RMSE and MMRE value for COCOMO and SA model for different development modes

Using Model	RMSE		MMRE	
	COCOMO	SA	COCOMO	SA
Semi-Detached	402.7236	81.58	30.018	22.483
Embedded	756.898	226.214	33.009	30.556
Organic	28.704	14.85	34.299	23.048
Combined mode	532.2147	154.9344	32.978	26.286
Single Optimized	***	177.9143	***	38.599

From the perusal of the above Table 6, it is seen that Semi-Detached SA model (SD_II), Embedded EM_II and Organic OR_ER was found to be the best optimized model, having an RMSE value of 81.58, 226.214 and 14.85 as against that of COCOMO which is 402.7236, 756.898 and 28.704 respectively. Further there is also an improvement in the MMRE value, which is 22.483, 30.556 and 23.048 as against 30.018, 33.009 and 34.299. Next, a single optimized model was also developed for all the 63 projects observed DE taken together using different MATLAB codes and were later compared with the COCOMO model. The results so obtained, as given above in Table 6 shows a far better RMSE value. The corresponding plots of all the models showing the plot of Observed DE versus Predicted DE using SA and corresponding COCOMO values are shown in the Figures 1, 2 & 3 below.

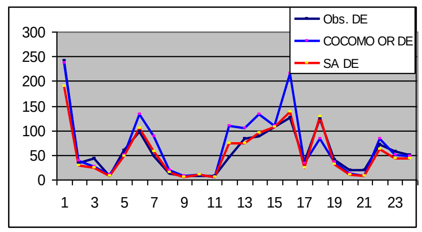

Fig 1.:- Plot of Observed DE Vs. COCOMO and SA model DE for Organic mode (OR_ER)

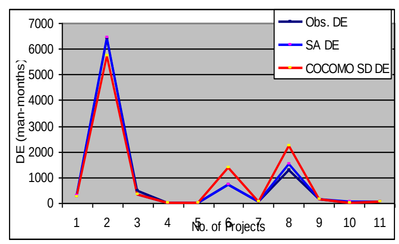

Fig 2.:- Plot of Observed DE Vs. COCOMO and SA model DE for Semi-detached mode (SD_II)

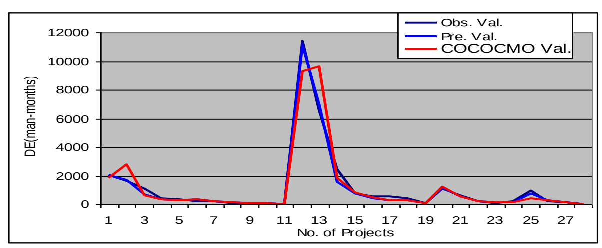

Fig 3.:- Plot of Observed DE Vs. COCOMO and SA model DE for Embedded mode (EM_II)

From the above two figures it can be inferred that the SA Development effort predicted value closely matched the observed DE value as compared to COCOMO DE value. The SA process although is random in nature but a majority of runs converge to the above solutions.

Further, the results so obtained above have been compared with the earlier work done by the author on ANN [11] and ANFIS [12]. The results are given in Table 7 below and are depicted in figure 4 & 5. From the perusal of the results given in Table 7 and Figure 4 above, it is seen that SA model has outperformed ANN, ANFIS and COCOMO models.

Table 7:- Comparative results of all the models

Model		RMSE	MMRE
COCOMO		532.2147	32.978
ANN		353.1977	83.0234
ANFIS	Training	0.00302	0.0000892
	Testing	2756.895	125.6756
	Complete	112.638	39.95
SA		***154.9344***	***26.286***

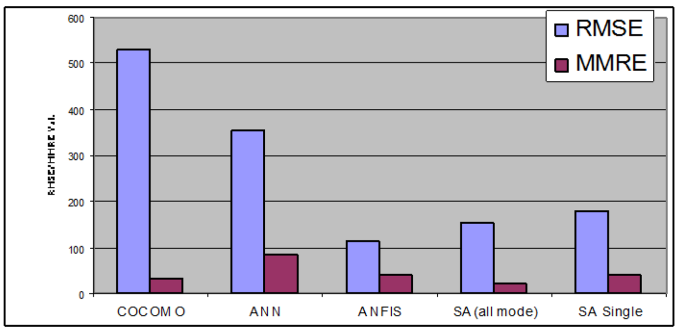

Fig 4. :- Comparative Plots of all the models in terms of RMSE and MMRE.

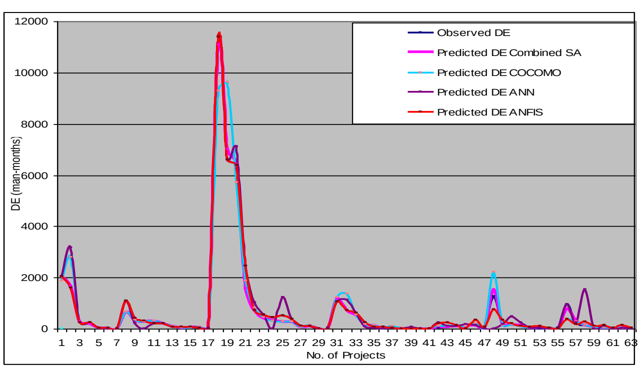

Fig 5. :- Plot of Observed Vs. Predicted DE for all the models

II. Conclusion

In this paper, it has been shown that Simulated Annealing algorithm can be made use of for estimating the best possible parameters of the effort components of software projects. The upper and lower limits of the parameters so defined should be given due weightage as the optimized model output depends upon the values of the model parameters to be optimized. For this multiple runs have to be carried out ignored to arrive at an accurate result. For this Matlab code was developed and run on Matlab platform. The results so obtained using Simulated Annealing Algorithm are better as compared to COCOMO, ANN and ANFIS. However, the effectiveness of SA's tending to depend on implementation details and how the problem is encoded.

For further study on SA modeling for software effort estimation one can increase the number of input variables and observe the results. Modeling on other datasets can also be attempted and the results validated with the SA approach.

References

[1]. Boehm, B.W., "Software Engineering Economics," Prentice-Hall, Englewood Cliffs, NJ, USA, 1994.
[2]. Jovan Zivadinovic, et. al., "Methods of Effort Estimation in Software Engineering", I International Symposium Engineering Management And Competitiveness ,2011.
[3]. S.K. Mohanty, et. al., "Software Effort Estimation Approaches – A Review", International Journal of Internet Computing, 2012, ISSN No: 2231 – 6965, VOL- 1.
[4]. Ekrem Kocaguneli, et. al., "AI-Based Models for Software Effort Estimation**"**, 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 2010, pp.323-326.
[5]. Vahid Khatibi, et. al., "A New Fuzzy Clustering Based Method to Increase the Accuracy of Software Development Effort Estimation", World Applied Sciences Journal, (2011),1265-1275
[6]. Prasad Reddy P.V.G.D, et.al.,"Software Effort Estimation using Radial Basis and Generalized Regression Neural Networks", JOURNAL OF COMPUTING, 2010, VOLUME 2, ISSUE 5, pp. 87-92.
[7]. Shiyna Kumar, et.al., "A Review of Surveys on Estimating Software Development Effort", IJAIR , 2013,Vol. 2 Issue 5, pp.448-451.
[8]. Iman Attarzadeh, et.al., "Software Development Effort Estimation Based on a New Fuzzy Logic Model", International Journal of Computer Theory and Engineering, 2009,Vol. 1, No. 4, pp. 473-476.
[9]. P. J. M. Laarhoven and E. Aarts, Simulated Annealing: Theory and Applications, Reidel, Ordrecht, 1987
[10] Darrall Henderson, et.al. , "THE THEORY AND PRACTICE OF SIMULATED ANNEALING", pp. 287-319. .
[11] Sheenu Rizvi et. al.,"Effort Estimation Using a Soft Computing Technique", IJCET,2013, Vol 4, Issue 6 ,pp.290-298 .
[12] Sheenu Rizvi et. al., "A Hybrid Fuzzy-ANN Approach for Software Effort Estimation", IJFCST, 2014, Vol 4, No. 5.

Author's Profile:

Sheenu Rizvi, is a Research Scholar in Integral University, U.P, India. Currently he is working as Assistant Professor in Amity School of Engineering and Technology, Amity University, Lucknow, India. His area of interest includes Software Engineering and Soft Computing.

Dr Syed Qamar Abbas got His M.S.(Computer Science) from BITS PILANI and PhD in Computer-Oriented study of Queueing models. Currently he is working as Director General of Ambalika Institute of Management and Technology, Lucknow, India. He has more than 26 years of teaching and research experience in the field of Computer Science and Information Technology. He has published more than 130 Research papers in International Journals & Conferences. 11 Ph.D have been awarded under his guidance.

Dr. Rizwan Beg got his Ph.D in Computer Science. & Engineering. from Integral University, Lucknow, U.P, India. Currently he is associated with Computer Science and Engineering department in Integral University. He has more than 17 years of experience which includes around 15 years of teaching experience. His area of expertise is Software Engineering., Requirement Engineering, Software Quality, and Software Project Management. He has published more than 80 Research papers in International Journals & Conferences.