CGANN-Clustered Genetic Algorithm with Neural Network for Software Cost Estimation

Pushpendra K Rajput, Geeta Sikka, and Aarti

***Abstract*—**Main challenges in the software development are predicting the accurate and credible cost of software projects to be developed in early stages. A serious consequence can occur using some miscalculated effort in the early phases of software project development life cycle. Several models have been developed to produce the accurate estimation, but no model has efficient and consistent with the uncertainty of the project development. This paper proposes a hybrid model that exploits the uncertainty using clustering the data. In this proposed model we use Genetic Algorithm (GA) combined with COCOMO model on clustered data. Model carries the desirable features of neural network, including learning ability to classify the new project for using the COCOMO model with best fit parameters. The best parameters of COCOMO model can be found for each cluster. Our proposed scheme made comparison of estimated effort with original COCOMO model which can be applied on larger data sets. This scheme also avoids the problem of different estimated cost of similar projects.

***Keywords*—** COCOMO model, Fuzzy c-Means (FCM) clustering, Genetic Algorithm (GA), Neural Network, Optimization.

I. Introduction

To develop a project successfully, it is necessary for any organization that the project should be completed within budget, on time and the project should have required quality. In order to produce a successful project, cost estimation is essential in managing software projects because of the uncertainty and diversity nature inherent in project development. Software development cost estimation is all about the future prediction of a work so that managers can make decision to how long and how many resources are required to complete the project. The use of inaccurate estimation makes the manager's decision as a recipe of disaster and looses the control and execute plan in a wrong direction.

For these reasons, we study software effort estimation model and techniques that can incorporate the accuracy of estimation. The cost estimation of the software project represented in terms of Person-Month (pm), depends on the many cost drivers used to develop a project. The main cost driver is the size of the project and usually given as Kilo Lines of delivered Code (KDLOC). In addition, the other cost drivers are used to define the complexity of the software project, given by the Effort Adjustment Factor (EAF). A number of software estimation models have been developed over the last decades. COCOMO [1] is one of the most popular models. Many computational Intelligence Techniques and soft computing provides some promising techniques such as fuzzy logic, Artificial Neural Networks (ANNs) and evolutionary computation for cost estimation modelling. ANNs have the ability to model complex nonlinear relationship and are capable of approximating any measurable function through learning. The evaluation algorithm such as simulated annealing and Particle Swarm Optimization (PSO) have been used to tune the best parameter of a regression model and resulted that the estimation has become sufficient with a significant level of improvement [2].

To handle the unpredictability that is generated by the inherent uncertainty contributed by the software cost drivers [3], soft computing techniques such as GA and Neural Network have been developed. GA is an optimization technique used in researching the optimal solution randomly [4][5][6][7]. A COCOMO based model is proposed here combined with GA [14]. Any soft computing technique results more accurately if the data values used have some sort of relationship. For the purpose of efficient use, this scheme uses cluster based GA. In the clustered approach the test data are classified by the Neural Network [8], which is an efficient technique for classification. Neural Network learn from the past experience and give the result for new test case, following which the best fit parameters can be applied to obtain the effort.

Rest of the paper is organized as follows: Section II describes the related background used in the proposed model. Section III Described the propose work followed by the section IV that demonstrate the data sets used and experimental results. In the end the conclusion is made in section V.

II. Related Research

A. COCOMO

In this proposed work we choose (Constructive Cost Model) COCOMO [9], which is the most famous model for software cost estimation proposed by Boehm. It is used for estimating software cost, effort and schedule that help the software developers regarding their decisions about investment, setting software budgets and schedule. COCOMO model define the

Pushpendra K Rajput is with the Sharda University, Greater Noida, India (phone:+917503502805; e-mail:pushpendrarajputs@gmail.com).

Geeta Sikka is with the Department of Computer Science and Engineering, National Institute of Technology, Jalandhar, India (e-mail: sikkag@nitj.ac.in).

Aarti is with the Department of Computer Science and Engineering, National Institute of Technology, Jalandhar, India (e-mail: aarti.1208@gmail.com).

relationship between development cost (pm), the size of the project and the EAF. It can be defined by the mathematical formulation as shown in eq. (1).

$$Effort\ (pm) = c * (KDLOC)^k * EAF \qquad (1)$$

Where,

$$EAF = \prod_j EM_j$$

Here, c and k are the statistical parameters that are previously estimated by using Boehm classification. Effort Multipliers (EMs) are values of the cost drivers associated with development complexity. Intermediate version of COCOMO model is used in this study that has fifteen cost drivers that are categorized into three groups shown in Table I. Each cost driver or effort multiplier has some value from the categorical set of values (very low, low, nominal, high, very high, extra high). The categorical values can be converted into the numerical form. using Table II. These values were taught by Barry Boehm after a regression analysis of the projects in COCOMO I dataset.

TABLE I
EFFORT MULTIPLIER OF COCOMO MODEL

Increase these to decrease effort	acap	analysts capability
	Pcap	programmers capability
	aexp	application experience
	modp	modern programing practices
	tool	use of software tools
	vexp	virtual machine experience
	lexp	language experience
	sced	schedule constraint
Decrease these to increase effort	stor	main memory constraint
	data	data base size
	time	time constraint for cpu
	turn	turnaround time
	virt	machine volatility
	cplx	process complexity
	rely	required software reliability

TABLE II
NUMERIC VALUES OF EFFORT MULTIPLIER

EM	Very low	low	nominal	high	Very high	Extra high
acap	1.46	1.19	1	0.86	0.71	--
Pcap	1.42	1.17	1	0.86	0.70	--
aexp	1.29	1.13	1	0.91	0.82	--
modp	1.24	1.10	1	0.91	0.82	--
tool	1.24	1.10	1	0.91	0.83	--
vexp	1.21	1.10	1	0.90	--	--
lexp	1.14	1.07	1	0.95	--	--
sced	1.23	1.08	1	1.04	1.10	--
stor	--	--	1	1.06	1.21	1.56
Data		0.94	1	1.08	1.16	--
Time	--	--	1	1.11	1.30	1.66
Turn	--	0.87	1	1.07	1.15	--
Virt	--	0.87	1	1.15	1.30	--
Cplx	0.75	0.88	1	1.15	1.40	--
Rely	0.70	0.85	1	1.15	1.30	1.65

B. Clustered Structure

Due to the inherent nonlinearity in the data available for software cost estimation to gain the accuracy in prediction is more difficult. This nonlinearity can be reduced by defining the certain relationship between the data values that can be attain by classifying the data into different cluster. Soft computing technique such as GA results better on such data [10]. The cluster structure has the following two steps.

1. Unsupervised learning for training data: To cluster the data a soft clustering algorithm Fuzzy C- Mean (FCM) algorithm is used [11]. FCM is a data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade. A data having N observations is clustered around K centeroids. Each value in the data set belongs to a cluster having the maximum value of membership.
2. Supervised learning for test data: The classification of the test data values is done by using the neural network. Back propagation model of neural network is used. It is a multilayer feed forward neural network. A neural network can be divided into two phases. The first phase is called the training phase in which some training data is passed into the network and the weights are adjusted through the back propagation algorithm to reduce the generated error. In the second phase the new data point or test data is fed to find the result based on past experience.

C. Genetic Algorithm

Holland first introduced GA as adaptive search algorithms that are based on concepts of natural selection and natural genetics. GA simulates a process of natural evolution that operates on chromosome. It works with parameter coding, rather than the parameter values. The working principle of GA depends on the representation of individuals, objective function, and genetic operators. Every individual is represented with the binary string. An objective function is used that decide the fitness of an individual. GA works with the principle of survival of the fittest, the individual having large fitness value is supposed to be more stable and close to the solution. GA uses three basic genetic operators: reproduction, crossover, and mutation. Reproduction operator uses the fitness of an individual solution and decides whether a string should be select for the new population, also called selection operator. There are a number of 0.83methods used to select an individual in the new population, Roulette wheel is one of the famous among them. Crossover requires a mating of two randomly selected st1rings. A part of the string is exchanged between two. The main characteristics of the parents are transferred to their child used for the new generation. Mutation operator works on the bits of the individuals. It adds information in a random way that introduces diversity in the population. A number of bits are randomly changed when the string is copied into the new population. Basic GA operations are illustrated in figure 1.

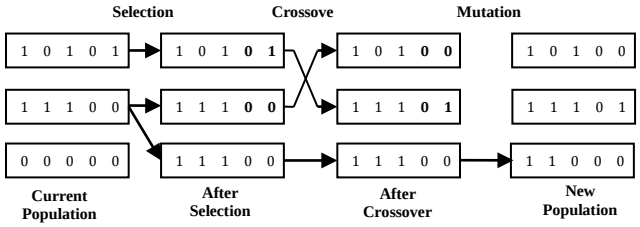

Fig.1 The basic GA operation

III. PROPOSED METHODOLOGY

The proposed methodology is based on the cluster structure. Here the proposed methodology employed for the CGANN model is described. The proposed model uses the FCM algorithm for clustering the input data sets. Then GA algorithm is used to find the best parameter of the COCOMO model for every cluster. A feed forward neural network is trained using the clustered values obtained for each project, by employing back propagation for finding the cluster values of testing data values. The flowchart of the proposed model is shown in figure 2. The detailed description is as follows:

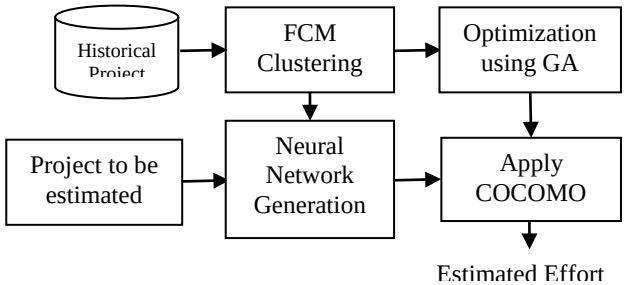

Fig. 2 The framework of proposed hybrid model

A. Clustering Data

The For the purpose of clustering the input data values we use the FCM algorithm the steps below describe the implementation overview of FCM.

Input: Data sets consisting of *n* values with *m* dimensions and number of clusters.

Output: *k* clusters of data values.

Step 1: Randomly choose the k values out of n values as initial centeroids.

Step 2: Define the membership function for each attribute.

$$\mu_{ij} = {}^{1}\Big/{\sum_{p=1}^{k}\left(d_{ij}/d_{ip}\right)^{2/m-1}}$$

Step 3: Calculate the degree of membership for each data value corresponding to every cluster. Assign each data value to the cluster for which the Euclidean distance is minimum.

$$\boldsymbol{d_{ij}} = \sqrt{\sum_{\boldsymbol{p=1}}^{\boldsymbol{m}}\left(\boldsymbol{x_{ip}} - \boldsymbol{x_{jp}}\right)^{\boldsymbol{2}}}$$

Step 4: Calculate the new centeroid of the cluster based on the mean value of membership degree of corresponding cluster as follows:

$$v_j = \left(\sum_{i=1}^{n}\left(\mu_{ij}\right)^m x_i\right)\Big/\left(\sum_{i=1}^{n}\left(\mu_{ij}\right)^m\right), \forall_j = 1,2,\ldots k$$

Step 5: Repeat steps 3-5 until the values obtained denote stable cluster values.

Step 6: Stop.

B. Apply Genetic Algorithm

Here we use genetic algorithm toolbox [12] to optimize the parameter of the COCOMO model for the clusters obtained from the above FCM algorithm. The process of GA is shown in figure 3.

Input: Genetic algorithm parameters and fitness function.

Output: Optimized COCOMO parameter values for each cluster.

Step 1: Write the fitness function that defines the fitness of the solution. Here we write the fitness function to minimize the accuracy parameter MMRE.

Step 2: Set the parameters for genetic algorithm used in the genetic operations.

(a) *Crossover:* we use the single point crossover. In single point crossover two individuals are selected randomly. Then a single crossover point is set between 0 and the length of the individuals. The part of the individuals is exchanged behind the crossover point, and the two new individuals for the new population are generated.

(b) *Mutation:* we use the uniform single bit mutation function, in which a bit has a probability to change it states from 0 to 1 or from 1 to 0.

(c) *Selection:* In the proposed work the most frequently used roulette wheel selection method is used. The individual having larger fitness value has more space on roulette wheel.

Step 3: Repeat step 2 until the stop criteria fulfilled either by number of generation or by the minimum change in fitness value.

Step 4: The final values of the parameters are the optimized values for the corresponding cluster.

Step 5: Repeat step 1-5 for each cluster.

Step 6: Stop.

C. CGANN Cost Estimation

Clustered GA with Neural Network (CGANN) model for estimation can be implemented using the following steps using the above described algorithms.

Training phase

Input: Data set of N values having some attribute that define the project with measured effort, the number of cluster *K.*

Output: A hybrid model for estimation of software projects.

Step 1: Cluster the given data set by using the algorithm described in section III.A.

Step 2: Using the clustered data train the neural network. The input layers receive the project attribute and the output layer gives the corresponding cluster value.

Step 3: Find the optimized parameter of every cluster using the algorithm explain in section III.B.

Testing Phase

1. The new project is fed into the neural network obtained from the clustered data
2. Use the corresponding parameters of the COCOMO model to which cluster the new project belongs.

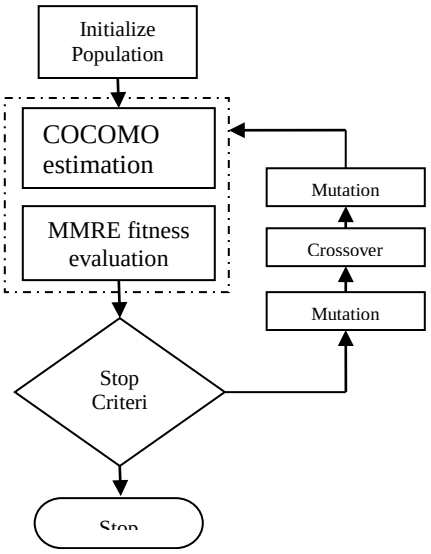

Fig. 3 The optimization process using GA

IV. Evaluation Criteria

A number of evaluation criteria to evaluate the performance of the proposed model that how accurate the estimation is defined in the field of software engineering [13]. Here we used most widely accepted criteria. First criterion is Mean Magnitude Relative error (MMRE), based on the Magnitude Relative Error (MRE). MRE for a particular project, defined with eq. (2), is a ratio of absolute error to the actual effort value i. e. percentage of error between actual and estimated effort. The MMRE value can be calculated as shown in eq. (3).

$$MRE = \frac{|Actual_{Effort} - Estimated_{Effort}|}{Actual_{Effort}} \quad (2)$$

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \quad (3)$$

Second widely used criterion is $PRED(l)$ presents the percentage of estimates that fall within l percent of the actual value. The definition of the $PRED(l)$ is shown in eq. (4).

$$PRED(l) = \frac{k}{n} \times 100 \quad (4)$$

Where k is the number of projects where $MRE_i \leq l\%$, and n is the number of all predictions.

To decrease the sensitivity of the extreme values, we also use the median of the MREs of n projects, represented as MdMRE. Median is less sensitive to extreme values, while MMRE is sensitive to the outliers.

$$MdMRE = median_i(MRE_i) \quad (5)$$

The most usable criteria MMRE also have the drawback in many validation circumstances that it leads to overestimation. To alleviate the problem of overestimation and underestimation we also used another accuracy indicator MMER. MMER is the mean value of MER that is the percentage of error between estimated effort and actual effort, defined in eq.(6).

$$MER = \frac{|Actual_{Effort} - Estimated_{Effort}|}{Estimated_{Effort}} \quad (6)$$

$$MMER = \frac{1}{n}\sum_{i=1}^{n} MER_i \quad (7)$$

The Box plots for residuals are also used that illustrate the distribution of residuals.

V. Experiment Results

A. Dataets

The two datasets are used in the validation of this study

The data used in this research taken from the COCOMO database. This data set has 63 projects with 16 effort drivers and actual effort for each project. For the purpose of this study the dataset is divided into two parts, one for training purpose and second for testing the validity of the model. The training and testing data is divided in the ratio of 80-20%. Hence 50 projects are used as the training data and 13 projects are used as the testing data.

The training data having 50 projects is clustered into three clusters using the FCM algorithm. Data belongs to a cluster has a degree of similarity shown in figure 4. After clustering the data GA algorithm is applied to each cluster to optimize the parameters of the COCOMO model. Before applying the GA we have to set the parameters of the algorithm shown in Table I. Optimized values of parameter for each cluster is listed in Table II.

TABLE I
GA Parameters Used

S. No.	Parameters	
	Name	Value
1	Generation Count	200
2	Population size	40
3	Crossover rate	0.8
4	Mutation rate	0.01

TABLE II
Optimized Parameters Of COCOMO Model for Each Cluster

Cluster No.	Parameters	
	c	K
1	2.50249445801975	1.14216161067868
2	1.5854974003043	1.28754127061868
3	4.07361843196589	0.98297625215604

For the classifying the testing data into the clusters we used a feed forward neural network with 25 neurons and 15 cost drivers as the input values and corresponding cluster as the targets of the system. The training is applied to the neural network and determines the cluster for testing data to which the values belong to. From the parameter values obtained for each cluster, listed in Table II are used to calculate the estimated effort using eq. (1). Figure 5 and figure 6 depict the comparison between the values obtained from COCOMO, CGANN, and the Actual Effort for training and testing data respectively.

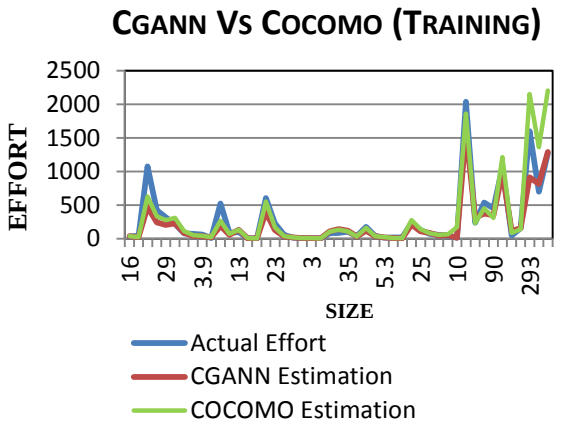

Fig. 5 Comparision of COCOMO, CGANN, and Actual Effort Values (Training data)

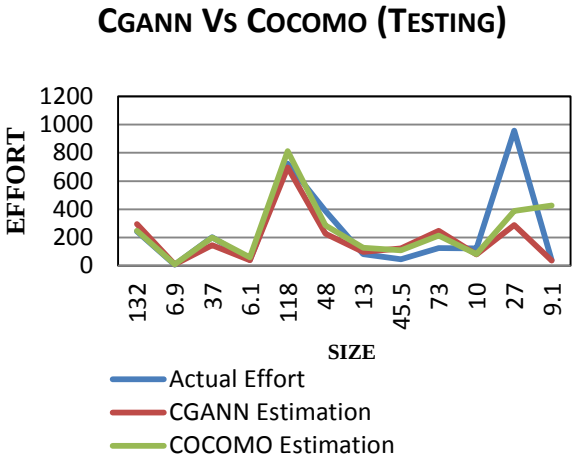

Fig. 6 Comparision of COCOMO, CGANN, and Actual Effort Values (Testing data)

Table III shows the estimated results at both the training and testing stages of COCOMO and Proposed CGANN model based on three evaluation parameters; MMRE, MdMRE, and PRED(l). For better estimation accuracy MMRE and MdMRE values should be lower and the PRED should be larger. Table III illustrate that the new proposed model provides much smaller MMRE than COCOMO. CGANN also provides the lager value of prediction.

TABLE III
ESTIMATION RESULTS OF CGANN AND COCOMO

Models	CGANN		COCOMO	
	Training	Testing	Training	Testing
MMRE	0.33	0.42	0.51	1.23
MdMRE	0.30	0.24	0.33	0.38
PRED(0.25)	0.54	0.50	0.44	0.33

VI. CONCLUSIONS

In this paper, we have proposed a clustered based GA model that estimates the cost of software project using COCOMO model. The Historical data is used to generate the model. The previous developed projects are classified into cluster and the optimized parameters are obtained using GA algorithm. Neural network is used to classify the new project that is supposed to be estimated. The hybrid approach has a significant improvement over the COCOMO model.

REFERENCES

[1] B. Boehm, Software Engineering Economics, Englewood Cliffs, NJ. Prentice-Hall, 1981.
[2] P.V.G.D. Reddy, C.V.M.K. Hari, and T.S. Rao, "Multi Objective Particle Swarm Optimization for Software Cost Estimation", International Journal of Computer Applications (0975-8887), vol32. No.3, 2011.
[3] T. gruschke, "Empirical Studies of Software Cost Estimation: Training of Effort estimation Uncertainity Assesment skills", 11th IEEE International Software Matrics Symposium, IEEE, 2005.
[4] D. E. Goldberg, *Genetic Algorithm in Search Optimization and Machine Learning*. Addision-Wesley, New York, 1989.
[5] L.Davis, *Handbook of Genetic Algorithm*. Van Nostrand Reinhold, New York, 1991.
[6] C. Kavita, "GA Based Optimization of Software Effort Estimation",IJCSI, vol.1,p.p 38-40,2010.
[7] A. F. Sheta,"Estimation of the OCCOMO model parameters using genetic algorithm for NASA software projects," Journal of Computer Science 2(2):118-123,ISSN-1549-3636.2006.
[8] S. Haykin, "Neural Networks: A Comprehensive Foundation, Prentice Hall", ISBN 0-13-273350-1, 1999.
[9] B. Boehm, Cost Models for Future Software Life Cycle Process: COCOMO2 Annals of Software Engineering. 1995. http://dx.doi.org/10.1007/BF02249046
[10] C. J. Burgess and, M. Lefley, "Can Genetic programming improve software effort estimation? A Comparative Evaluation. International and Software Technology 43, 863-873.2001.
[11] S. Adaekalavan, and C. Chandrasekar,A Comparative Analysis of Fuzzy C-Means Clustering and Harmonic K Means Clustering Algorithms. European Journal of Scientific Research ISSN 1450-216X Vol.65 No.1 pp. 45-49, 2011.
[12] Chipperfield, A. J., P. J. Fleming, and H. P. Pohlheim, "AGenetic Algorithm Toolbox for Matlab", Proceeding of International Conference on System engineering, Coventry, 6-8 september 1994.
[13] E. Mendes, N. Mosley, and s Counsell, "A Replicated Asessment of the use of adaption rules to improve web cost estimation", International symposium on empirical software engineering, pp.-100-109.2003.
[14] S. J. Huang, N. H. Chiu, L. W. Chen, "Integration of the grey relational analysis with genetic algorithm for software effort estimation", European Journal of Operational Research 188, 898-909, 2008. http://dx.doi.org/10.1016/j.ejor.2007.07.002