Journal of Scientific Research and Development 2 (4): 87-94, 2015
Available online at www.jsrad.org
ISSN 1115-7569
© 2015 JSRAD

A new approach based on continuous genetic algorithm in software cost estimation

Farhad Soleimanian Gharehchopogh [1,*], Amir Pourali [2 *]

[1]*Department of Computer Engineering, Hacettepe University, Beytepe, Ankara, Turkey*
[2]*Sama technical and Vocational training college, Islamic Azad University, Urmia Branch, Urmia, Iran*

Abstract: Software Cost Estimation is a considerable issue in the development and production of software projects. Delivery time of project and completing it in a timely fashion is a problem that software companies must be overcome them. Recently, the usages of Meta-heuristic techniques which are presented for software cost estimation are increasingly growing. Prerequisite of relative accurate estimation is work experience. Therefore, the risk associated with construction of projects is based on preliminary estimates. Increasing risk causes increasing uncertainty about the initial program with increasing complexity and size of projects level of uncertainty become higher. In this paper, we aim to develop an evolutionary model for software cost estimation by using continuous genetic algorithm. We scrutinized adjusted COCOMO II model by using data set of NASA projects to examine the effect of the developed model and showed the effectiveness of the proposed algorithm in correction of parameters of COCOMO II. Experiment results show that this model offers very good estimate for software cost estimation.

Key words: Software cost estimation; COCOMO II; Genetic algorithm

1. Introduction

Software Cost Estimation (SCE) is the estimation of scheduled cost and personnel needed in one month (person-month) for a software project. Increase in complexity of software causes increase of costs. Therefore, the estimation accuracy in the early stages of software projects would be highly desirable. The SCE development is sensitive, complex and inevitable. In the last three decades, a significant growth has been seen different models in cost estimation.

Effort Estimation is a realistic forecast of the required effort to construct and maintain software. Inaccurate estimation of software effort will lower the efficiency of the project; will be a waste of company funds and will make failing results during the project. Impact of the failure or success of software projects depends on software effort estimation. Software engineering community has provided many models to find effort of software development as some of them are COCOMO, SLIM, and function point (FP) which are classified in the category of algorithmic models (Tim Menzies, 2006).

SCE known as a science and many variables like personal, techniques, environment and politics are effective in approximate software cost and efforts for its development. However, SCE can become series of systematic steps from an unknown art to provide estimates with acceptable risk. In this paper, the basic model is used COCOMO model, that we adjusted some important parameters of COCOMO II model using genetic algorithm to increase its efficiency.

The structure of the paper as follows: after a brief overview of previous work in this area, firstly the concept of SCE and its importance in the process of software engineering is presented. Then, the proposed algorithm will be scrutinized in section 3. The evaluation criteria are presented in section 4. In Section 5, we are shown the effort estimation using COCOMO II and genetic algorithm and the results of paper are shown in section 6.

2. Related works

The basic COCOMO model was developed by Barry W. Boehm in 1981 (Boehm, B.W., 1981). In 1987, Kemmerer (Kemerer, 1987) did a closer look at software project cost estimation models and analyzed and examined estimates, SLIM, COCOMO and FP patterns. According to the results of these models, it was observed that the error rate of the various models have a high value of error. So, the estimation models have a significant impact on the accuracy of software effort. COCOMO model is the most authentic and transparent available model in cost estimation. In this method, the value of needed code is estimated for software production, based on it the value of required work for software projects have been estimated. Since the cost estimation methods which are based on pattern are less accurate. In recent years, methods which are based on Meta-heuristic algorithms for estimating parameters have been earned more attention.

Hakkarainen, J. et.al in 1993 used the Meta-heuristic algorithm called artificial neural networks

[*] Corresponding Author.

(ANNs) to estimate the size of the software projects. This observation showed that ANNs operate very well to obtain results. In this method the value of error tolerance depends on the tested particular application and its training algorithm depends on limited space, which was limited by its parameters. The main advantage of using ANNs is its compatibility, non-parametric and its predictive model that would be suitable for the particular data.

In 2000, Boehm provided COCOMO II model to estimate the cost and schedule software projects (B. Boehm et.al, 2000). This model can be used to determine the various software projects. Since the success of most organizations are increasingly dependent on software development solutions. Over 50% of software projects failed in schedule and budget. While some of these failures are unpredictable but they can be partly avoided by more accurate models like COCOMO II. COCOMO II has been created by using modern software methods and structural engineering by done studies in the initial COCOMO field and its correction. COCOMO II had been completed with using statistical techniques for compatibility expert judgment method and using 161 software projects that are carefully collected.

Sheta A.F in 2006 used the genetic algorithm to estimate the parameters of COCOMO model. In this study, data set of NASA software projects are used for scrutinizing and analysis of the proposed model. Also, having two important and influential factors; Developed Lines of Software Code (DLOC) and Measured Effort (ME), this work were analyzed and examined on 18 projects. The results showed that the genetic algorithm provides a better cost estimation. The structure of this model is based on value of software efforts as a function of DLOC. Establishing such a function helps project managers to allocate resources to their projects accurately. In this study, two methods were used. First, the famous COCOMO model is corrected, then parameters of the COCOMO model are estimated by using genetic algorithm and its efficiency is evaluated by using 18 projects of NASA. The Value Adjustment Factor (VAF) is the fitness function that has been used in this model i.e. the ratio of the variance of the difference between actual effort and estimated effort on the value of actual effort. The proposed model has estimated the parameters of intermediate COCOMO model and evaluates its efficiency using 60 projects from data set of NASA. The main difference between this model and the proposed one is that the chromosomes which are made to estimate parameters of COCOMO model are not binary. In other words, genetic algorithms genes are coded as continuous numbers. Intersection operator and genetic operator are defined differently. The Intersection operator is defined by formula (3-3) and mutation operator is defined by formulas (3-4), (3-5), and (3-6). The proposed model used evaluation function such as Magnitude of Relative Error (MRE), PRED (x), Mean MRE (MMRE), and VAF.

Other researchers have used fuzzy logic to estimate the cost of software projects (Mittal, a. et.al. 2010). In this study the amount of KLOC which indicate size of program lines is considered as a fuzzy number. They used 14 projects from a set of KEMERE projects for this purpose, and also they used MATLAB software to tune parameters of famous COCOMO model. This study showed that the percentage error of MMREand PRED (N) of the proposed method is better than using an algorithmic method such as intermediate COCOMO.

Another researcher used genetic algorithm to evaluate software effort estimation (Singh, B.K., Misra, 2012) and considered parameters value of COCOMO model are optimized by genetic algorithm to obtain the results of observations which utilize NASA data set and the value of MMRE get to 0.2298% in comparison with COCOMO model.

In the prior work, have been used soft computing techniques to the SCE (Ziauddin et.al., 2012) and they are combined fuzzy logic and particle swarm optimization algorithm for estimation. The results of COCOMO model and Sheta model was compared on the basis of MMRE, PRED (N), and VAF measures. The proposed model is able to provide a better estimation in comparison with other methods and the MMRE value got to 7/512 % in this model.

Data mining techniques and algorithmic models are used in order to examine and estimate software costs in the conducted research (Khalifelu and Gharehchopogh, 2012). They have also studied COCOMO model by data mining techniques. Cost estimation is stimulated by the application of ANNs, Liner Regression (LR), Support Vector Regression (SVR) and K-Nearest Neighbors (KNN) techniques. The consequence of this research shows that SVR model has less MRE in comparison with other models. Researchers have used regression model for clarifying available data in NASA and calibrating the parameters of COCOMO model (Khalifelu and Gharehchopogh, 2012). The results show that regression model has decreased the error value of MRE and have a good impression on calibrating parameters of COCOMO model. Also, data mining techniques are used in SCE in (Khalifelu and Gharehchopogh, 2012). They have used neuro fuzzy (NF), ANNs, LR, fuzzy decision trees (FDTs), Bayesian network (BN), multilayer liner regression (MLR) and fuzzy logic (FL) techniques. Each inserted factor is given one weight by using ANNs and then their values are calculated by the layers in ANNs.

3. Software cost estimation

In recent years, many researchers have carried out in order to find the main reasons of the failure of software projects. They demonstrate that incorrect estimations are salient reasons of them. Since, weak estimation and not exact software projects are among the most important and radical reasons, the increase of unsuccessful software projects and the projects, which their budgets and scheduling are furthered from the predictions, are resulted in considerable importance of effort estimation in the management of software projects. Weak estimation leads to furthering the real budget and time of a project them estimations or increasing estimations than the real budget and scheduling of the project. It is noticeable that total failure of project can be occurred in the worst situations. The feasibility of a project before beginning and the effective management of development process are occurred by the help of software effort estimation especially in the primary steps of software development. One of the most challenging among the researchers of software engineering is increasing the accuracy of SCE especially in primary steps (Attarzadeh, 2011). The significant role of software in today's business market is the exact estimation of software costs. Its reasons are organized follow as:

- ❖ It is used to determine the sources of project and how these sources are used.
- ❖ It is helped to classify the projects which are underdeveloped.
- ❖ When the sources harmonize with real needs, control and management of projects are easy.

Cost estimation, as a software project includes some techniques and processes that are used in an organization so as to estimate cost, scheduling, human force and the sources of software projects (Verma and Sharma, 2010). SCE contains effort determination in which effort is measured in a person-months way. Generally, SCE is a process to predict the amount of needed effort for creating a software project. SCE is a continuous activity which the life cycle of the software is started in the primary steps and is continued in the whole lifetime. Cost estimation is not just a financial cost. On the contrary, it includes finding the direction of successful performance of project and humans which are active in the project.

It is necessary to note that our aim is to estimate its cost. There are different ways to estimate the cost of software which are generally divided into two main groups named algorithmic and non-algorithmic methods. Algorithmic method is based on mathematical equations. It needs enough information in primary steps so as to estimate and has low flexibility. COCOMO method is the most famous algorithmic method. Because the accuracy of model is low, doing extensive researchers for combining the model with different techniques and increasing its accuracy should be done. However, non-algorithmic methods work on the basis of inference and present a good estimation about previous projects with having some information. Software cost can be estimated in any of the following ways:

- ❖ Algorithmic model (Heiat A., 2002 and Shin M., Goel A.L, 2000): cost models such as COCOMO 81, COCOMO II and limited model like SLIM.
- ❖ Analogy (Huang S. J., Chiu N, H, 2006 and Auer M., et.al, 2006): like machine learning.
- ❖ Expert judgment (Jorgensen M., 2004) and models such as Parkinson's low, pricing-to-win, low-to-high estimation and high-to-low estimation.

3.1. Continuous genetic algorithm

The variables are not encrypted in a binary way in continuous genetic algorithm. Continuous chromosome is created after the variables are put, nevertheless, by each other as they are naturally. In fact, each chromosome is a collection of numbers and also it can be expressed as a vector. It is possible to normalize the amount of variables in a way that the numbers of all variables are put in interval [0, 1]. The total operation of continuous genetic algorithm is similar to its binary fellow. Mutation and crossover operates are the marked difference between two types of algorithms. For this algorithm, crossover operator can be defined in different ways. The most popular definition for this operator follows as:

$X=x_1, x2,\dots,x_n$(3-1)

$Y=y_1, y2,\dots,y_n$

$X\boxtimes Y= \alpha_1x_1+1-\alpha_1y_1,\ \alpha_2x_2+1-\alpha_2y_2,\dots,\ \alpha_nx_n+1-\alpha_ny_n$

$X\boxtimes Y= Y \boxtimes X$

$\alpha=\{\alpha1, \alpha2, \dots, \alpha n\}$

In order to show mutation on a chromosome, one component of chromosome should be selected to make random changes on it. This change can be explained with every distribution of suitable probable. It is noticeable that the new amounts of variables remain in the permissible interval.

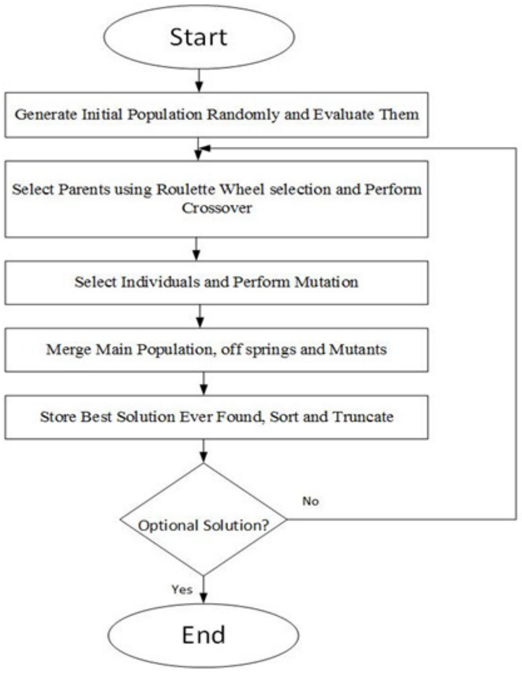
Start
Generate Initial Population Randomly and Evaluate Them
Select Parents using Roulette Wheel selection and Perform Crossover
Select Individuals and Perform Mutation
Merge Main Population, off springs and Mutants
Store Best Solution Ever Found, Sort and Truncate
Optional Solution?
No
Yes
End

Fig. 1: Flowchart of the proposed model

3.2. Proposed model by using continuous genetic algorithm

In this model, we want to improve the parameters of COCOMO II which is derived from formula (3-2) by using genetic algorithm in order to intensify the efficiency of the proposed model.

$PM=a\times (size)\, b\times\prod_{i=1}^{15} EMi$ (3-2)

The formula (3-2) consists of two parameters named a, b and 15 factors of effort multipliers (EM). 5 factors out of 15 have the greatest importance as well as influence in estimating the needed efforts of software projects (Menzies et al., 2005). These multipliers are as following: required software reliability (RELY), process complexity (CPLEX), main memory constraints (STOR), use of software tools (TOOL), and schedule constraint (SCED)

The multipliers with enormous importance have been chosen so that a, b parameters and five effort multipliers are estimated by the employment of genetic algorithm. Seven parameters of continuous genetic algorithm are used because of SCE. First, seven parameters are considered as the genes of genetic algorithm chromosome. Then, primary amounts are allocated to them in considered intervals randomly and monotonously. Table 1 determines those intervals which each parameter gains its amount of them.

Table 1: Genes in genetic algorithm

Parameters	Lower Bound	Upper Bound
Rely	0.75	1.40
CPLX	0.70	1.65
STOR	1	1.56
TOOL	0.83	1.24
SCED	1.10	1.23
a	2.8	3.2
b	1.05	1.2

The amounts of genetic algorithm parameters are determined in Table 2.

Table 2: Parameters of genetic algorithm evolutionary process

Parameter	Value
Size of Population	10
Number of Generation	10
Number of Runs	10
Mutation Percentage	0.4
Crossover Percentage	0.6
Mutation Rate	0.02
Fitness Function	MMRE
Selection Method	Roulette Wheel

3.2.1. Crossover operator in genetic algorithm

Arithmetic crossover is used for generating children. Generated children of these parents will be y_1 and y_2. If alpha is considered as a random number with monotonous distribution between (-gamma, 1+gamma) and if x_1 and x_2 are observed as the parents as shown in formula (3-3).

y_1=alpha×x_1+ (1-alpha) ×x_2
y_2=alpha×x_2+ (1-alpha) ×x_1 (3-3)

3.2.2. Mutation operator in genetic algorithm

At first, sigma is determined by using formula (3-4).Where Min is the low amount of gene and Max is the high amount of it.

Sigma=0.1× (Max-Min) (3-4)

Then, a chromosome is considered in which mutation will be operated on it. It is noticeable that some genes are selected with respect to the rate of the operations of mutation until the operation is operated on it. It is the numbers of components, which will impressed by mutation operator. The sigma is multiplied by one random number with a normal distribution. After that, it is pulsed with the considered gene in order to create new amount of gene as shown in formula (3-5).

y_2=y_1+ Sigma× NormalRandom (3-5)

At last, the gained number is sent to the considered and determined interval by using formula (3-6).

y_2=maximum (y_2, Min), y_2=minimum (y_2, Max) (3-6)

4. Performance validation

Performance of the proposed model is assessed by the following performance validations: MRE, MMRE, VAF and PRED(X)

Next, the amount of errors is compared with COCOMO model in the following diagrams and tables. Finally, it is demonstrated that the proposed algorithm has higher efficiency and lower error magnitude in comparison with COCOMO model.

1- MRE: It equals absolute value of the magnitude of calculated error as formula (4-1).

$$\text{MRE}=\frac{|\,estimated-actual\,|}{\text{actual}}\times 100 \quad (4\text{-}1)$$

2. MMRE: It equals the average percent of the calculated absolute value in dataset as calculated by formula (4-2).

$$\text{MMRE}=100/N\sum_{i=1}^{N} MRE\ i \quad (4\text{-}2)$$

3. Predictions: It shows that the accurate amount of defined model. It is noticeable that the relationship between the amount of estimation is returnable to the amount of real cost like formula (4-3).

$$\text{PRED (x)} = \frac{1}{n}\times\sum_{1}^{n}\begin{Bmatrix}1, if\ MRE \le x\\ 0, otherwise\end{Bmatrix} \quad (4\text{-}3)$$

4. Correlation: propriety ratio is the best answer in every population of genetic algorithm in comparison with the average of fitness functions in the whole of that population.

5. VAF: it is calculated that by formula (4-5) in which v is considered as variance.

$$[1-\frac{\text{var (Actual Effort}-\text{Estimated Effort)}}{\text{var (Actual Effort)}})]\times 100 \quad (4\text{-}5)$$

5. Experiment results

60 projects of NASA data set have been used to estimate the amount of accuracy of the proposed algorithm. The prior 48 projects have been used for training data and the next 12 ones are for testing data.

Because the amount of calculated MRE for COCOMO 81 with the amount of real efforts and estimated effort by COCOMO model have been shown, these amounts are compared with the gained amounts of genetic algorithm. We have shown that to what extent genetic algorithm can improve the value of effort estimations.

Table 3: Estimated effort and mre results using cocomo and proposed models

Project. No	Actual Effort	Estimated Effort using COCOMO	Estimated Effort using Proposed	MRE using COCOMO	MRE using Proposed
1	8.4	6.4	7.68	23.4112	8.57
2	10.8	10.5	12.50	3.0052	15.79
3	18	16.8	20.10	6.4575	11.66
4	24	9.9	14.68	58.7526	38.85
5	25.2	30.5	36.47	21.2301	44.72
6	31.2	24	28.62	23.1646	8.28
7	36	25.6	30.57	28.8618	15.08
8	36	25.2	37.37	29.9932	3.79
9	42	30.2	38.94	28.0791	7.28
10	42	32.7	48.43	22.2245	15.31
11	48	28.1	47.91	41.4606	0.19
12	48	25.9	38.41	46.0237	19.97
13	48	35	51.96	26.9889	8.25
14	50	32.9	39.24	34.2621	21.53
15	60	64.3	76.74	7.1369	27.89
16	60	44.9	53.61	25.1514	10.65
17	60	54.4	80.63	9.3564	34.39
18	60	117.5	111.63	95.7620	86.05
19	60	62.7	92.90	4.4362	54.84
20	62	45.1	61.66	27.2475	0.55
21	70	54.8	74.87	21.7529	6.96
22	72	28.5	42.29	60.3816	41.26
23	72	42.1	38.02	41.5027	47.20
24	82	58.1	79.47	29.0990	3.08
25	90	54.9	81.40	38.9952	9.55
26	98.8	57.3	74.22	42.0341	24.88
27	107	146.6	112.20	37.0412	4.86
28	114	75.1	111.31	34.1445	2.36
29	117.6	85.7	102.28	27.1452	13.03
30	117.6	81.2	96.89	30.9799	17.61
31	120	98.2	117.25	18.1474	2.29
32	155	99.5	129.01	35.7782	16.77
33	170	120.4	164.55	29.1893	3.21
34	192	131.6	179.95	31.4342	6.28
35	210	151	194.62	28.1149	7.32
36	215	475.9	545.20	121.3408	153.58
37	239	182.7	249.71	23.5636	4.48
38	252	158.1	212.67	37.2491	15.61
39	278	220.2	267.42	20.7866	3.80
40	300	231	275.71	23.0112	8.10
41	324	491.4	350.81	51.6775	8.27
42	352.8	231	275.71	34.5333	21.85
43	360	418.2	479.12	16.1679	33.09
44	360	200	229.18	44.4330	36.34
45	370	234.2	343.13	36.7142	7.26
46	400	219.8	325.92	45.0431	18.52
47	420	436.9	550.37	4.0311	31.04
48	450	280.6	457.19	37.6416	1.60
49	480	539.8	492.83	12.4565	2.67
50	571.4	431.4	559.09	24.5015	2.15
51	750	602.7	580.09	19.6397	22.65
52	815	862.2	880.76	5.7864	8.07
53	973	1353.8	1695.78	39.1323	74.28
54	1181	1236.7	803.53	4.7126	31.96
55	1248	1202.7	1156.07	3.6269	7.37
56	1368	1139.6	1805.10	16.6991	31.95
57	2120	1509.6	1618.08	28.7930	23.68
58	2300	1731.7	1353.08	24.7095	41.17
59	2400	2419.3	1220.64	0.8042	49.14
60	3240	4068.6	2806.38	25.5726	13.38

Fig.2 shows a diagram about calculated error called MRE of COCOMO model 81 with 60 NASA projects.

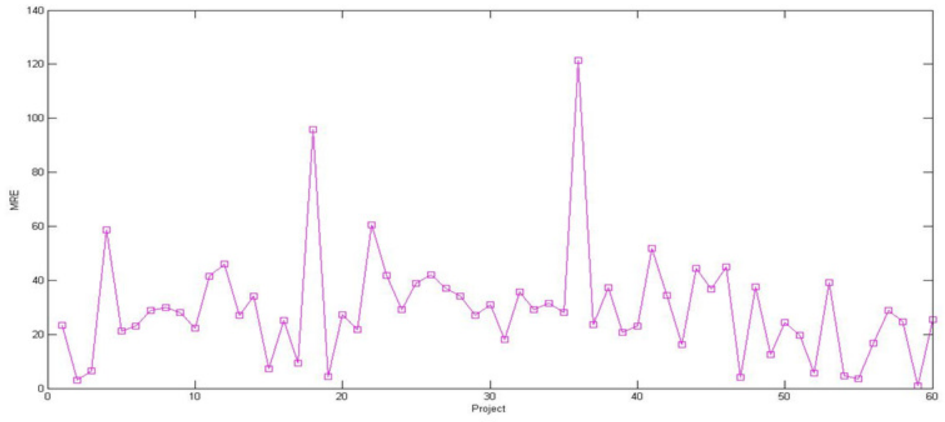

Fig.2: MRE results in COCOMO model

6. Results and discussion

60 projects from the data set of NASA projects were selected to study the amount of efficiency of this model. The prior 48 of 60 projects were related to training data and the next 12 ones were considered as testing data. Gained values of genetic algorithm were recorded and listed below after 10 times of successful performance of genetic algorithm.

Fig.3 shows the comparison of MRE value in 60 projects of NASA's data set, using genetic algorithm and COCOMO 81 model.

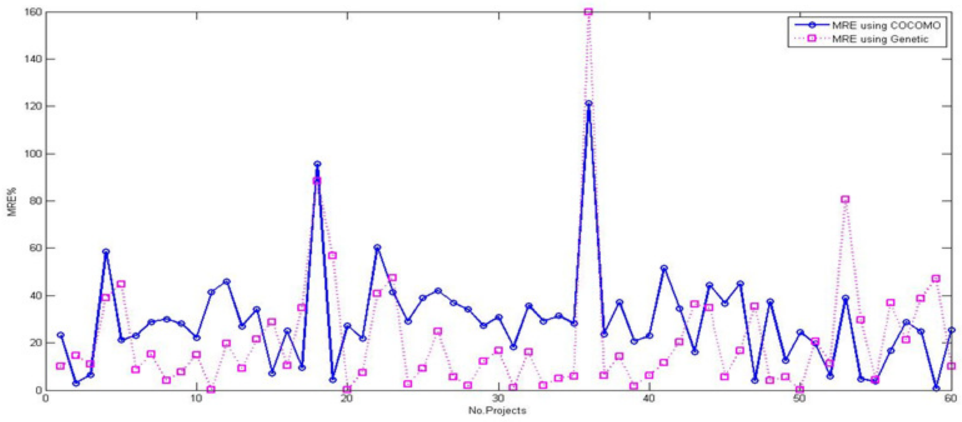

Fig.3: MRE results using COCOMO and proposed models

Table 4 shows the value of MMRE in comparison with the real effort of projects for training data. This value is 29/52 for the COCOMO model which is reduced to 21/53 by using the proposed model.

Table 4: Performance Comparison of COCOMO and Proposed Models based on MMRE

Models	Proposed Model	COCOMO Model
MMRE	21.53	29.52

Figure 4 shows the reduction of MMRE value over 10 generations after execution of genetic algorithm to the training data.

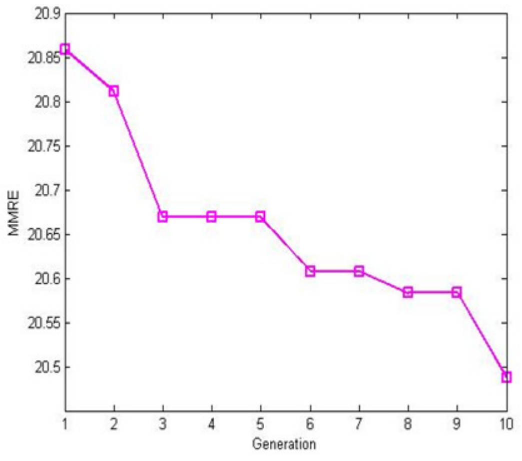

Fig.4: The best MMRE for 10 generation

Table 5: Value of gens using genetic algorithm

a	b	REY	CPLEX	STR	TOL	SCED
2.9131	.0500	8413	1.6489	2708	.8999	.1806

The values after training with 48 projects and the values for the parameters in the formula PM=a×(size)b×$\prod_{i=1}^{15}$ EMi are obtained by genetic algorithm which are summarized in Table 5.

PM=2.9131× (size) 1.05×$\prod_{i=1}^{15}$ EMi

Two last columns in Table 3 shows the results of MRE values obtained after applying genetic algorithms as shown in Table 5 for all data.

Fig.5 shows the comparison of the actual efforts and the estimated efforts using the proposed model based on the number of program lines.

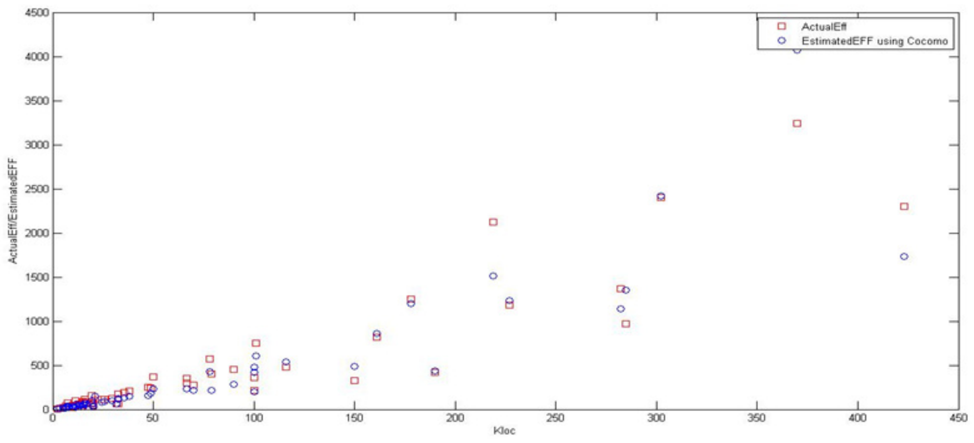

Fig.5: Actual and estimated effort using genetic algorithm

Comparison of the PRED value and variance of COCOMO II and proposed models are listed in Table 6.

Table 6: Values of PRED and VAF measures in the COCOMO and proposed models

Projecs	PRED(2)	PRED(25)	PRED(3)	VAF
COCOMO Model	25	41.66	61.66	92.67
Proposed Model	63.33	71.66	75	85.4007

Convergence factor: MMRE Ratio fitness of the best solution in each population in genetic algorithm to the mean fitness function of entire population as defined in Figure 6.

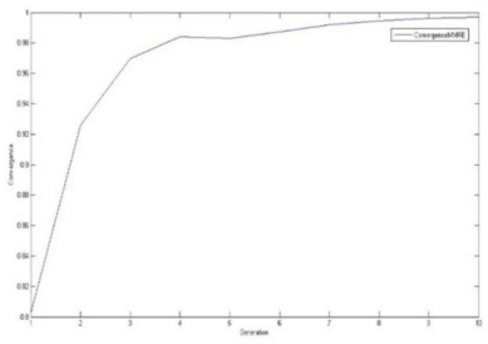

Fig.6: Convergence of MMRE in genetic algorithm

Fig.7 shows the convergence of genetic algorithm parameters (a, b) in the corresponding intervals.

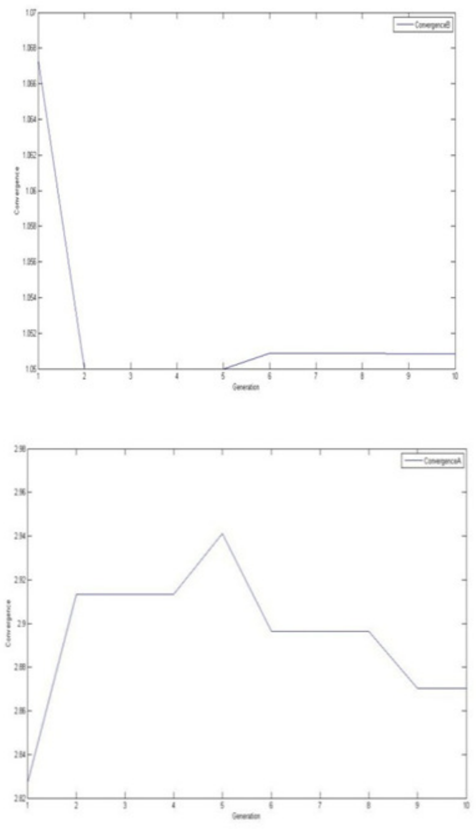

Fig.7: Convergence of parameters a, b in genetic algorithm

7. Conclusions

Using the results of COCOMO II model and the proposed model with genetic algorithm and comparison of the error value in COCOMO II model and proposed algorithm can conclude that, the proposed algorithm is able to moderate required parameters of COCOMO II model and create an effective model in SCE. In the future, we can combine the Meta-heuristic algorithms with ANNs and other data mining techniques to obtain better results in SCE field.

References

Attarzadeh, I. Ow, S. H. (2011), Improving Estimation Accuracy of the COCOMO II Using an Adaptive Fuzzy Logic Model, International Conference on Fuzzy Systems (FUZZ), IEEE, pp. 2458-2464

Auer M., Trendowicz A., Graser B., Haunschmid E., Biffl S.(2006). Optimal project Feature weights in analogy-based cost estimation: improvement and limitations. IEEE Transactions on Software Engineering, 32(2), 83-92.

B. Boehm, E. Horowitz, R. Madachy, D. Reifer, B.K. Clark, B. Steece, A.W. Brown, S. Chulani, and C. Abts, "Software Cost Estimation with COCOMO II". Prentice Hall, 2000

Boehm, B.W.(1981). Sofware Engineering Economics ,Prentice-Hall, Englewood Cliffs, New Jersy

Hakkarainen, J., Laamanen, P., Rask, R. (1993). Neural Network in Specification Level Software Size Estimation, Proceeding of the 26th Hawaii International Conference on, Vol.4, pp. 626-634

Heiat A. (2002). Comparison of artificial neural network and regression models for estimating software development effort. Information and Software Technology, 44, 911-922.

Huang S. J., Chiu N, H. (2006). Optimization of analogy weights by genetic algorithm for software effort estimation.Information and Software Technology 48, 1034-1045

Jorgensen M. (2004). A review of studies on expert estimation of software development effort. Journal of Systems and Software, 70, 37-60.

Kemerer, C.F. (1987). An Empirical Validation of Software Cost Estimation Models. Communication of the ACM, vol. 30, No.5 , pp. 416-429

Khalifelu, Z.A., Gharehchopogh, F.S.(2012a). Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation. Elsevier, Procedia-Technology Journal, ISSN: 2212-0173, Vol. 1, pp. 65-71

Khalifelu, Z.A., Gharehchopogh, F.S.(2012b). A New Approach in Software Cost Estimation Using Regression Base Classifier. AWERProcedia Information Technology&Computer Science Journal, Vol. 2, pp. 252-256.

Khalifelu, Z.A., Gharehchopogh, F.S.(2012c). A Survey of Data Minng Techniques in Software Cost Estimation. AWERProcedia Information Technology Computer Science Journal, Vol. 1, pp. 331-342.

Menzies T, Port D, Chen Zh, Hihn J., 2005, Validation Methods for Calibrating Software Effort Models.

Mittal, a., Parkash, K., Mittal H. (2010). Software Cost Estimation Using Fuzzy Logic. ACM SIGSOFT Software Engineering, Vol. 35, No. 1, pp. 1-7.

Sheta, A.F.(2006). Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects, Journal of computer science, vol. 2, No.2, pp. 118-123.

Shin M., Goel A.L. (2000). Empirical data modeling in software engineering using radial basis functions. IEEE Transactions on Software Engineering, 26(6), 567-576.

Singh, B.K., Misra, A.K. (2012). Software Effort Estimation by Genetic Algorithm Tuned Parameters of Modified Constructive Cost Model for NASA Software Projects. International Journal of Computer Applications, Vol. 59,No .9, pp . 22-26

Tim Menzies, Member ,IEEE , zhihacchen , JairusHihn and Karen Lum Selecting Best Practices for Effort Estimation, IEEE Transaction on Software Engineering , Vol. 32, No. 11, November 2006

Verma, H.K. Sharma,V. (2010), Handling imprecision in inputs using Fuzzy logic to predict effort in software development, Advance Computing Conference (IACC), IEEE, pp. 436-442

Ziauddin Khan Zia, Shahid Kamal Tipu, KhiruzZaman Khan, Shahrukh Khan Zia.(2012). Software Cost Estimation Using Soft Computing Techniques. Advances in Information Technology and Management (AITM), Vol. 2, No. 1, pp. 233-238.