An Effective Precision Enhancement Approach to Estimate Software Development Cost: Nature Inspired Way

Ankita, Somya Jain and Chetna Gupta
Department of Computer Science and Information Technology
Jaypee Institute of Information Technology, A-10, Sector-62, Noida, Uttar Pradesh, India.
chetnagupta04@gmail.com

***Abstract*— In recent years, many researchers and practitioners have explored the possibility of estimating effort and cost using nature inspired algorithms. The purpose of this paper is to investigate the relevance of bacterial foraging optimization algorithm (BFOA) for optimizing the COCOMO model coefficients to estimate the software development time. The goal of this research is to minimize the fitness function value which is the measure of the deflection of estimated time from the real time taken in the software development. Results of the experimental study conducted shows that the proposed approach produces promising results in comparison to COCOMO model and other existing approaches listed in literature. Results show that COCOMO model and other existing approaches are less accurate in comparison to BFOA with MMRE as 0.16 and PRED(25) as 0.9. Thus BFOA can help software industry in predicting accurate and reliable values for planning and maintenance of software project.**

***Index Terms*— Bacterial Foraging Optimization Algorithm; COCOMO; Fitness Function; Nature Inspired Algorithm; Software Cost Estimation.**

I. Introduction

Software effort estimation is the process of predicting effort required to develop and maintain a software system once the requirements are finalized. Accurate effort estimation is essential for the success of any software system development. Inaccurate and unreliable results can results in customer dissatisfaction and risk of inflation in cost of project development [1]. Therefore accurate effort estimation has to be conducted at early stage of software development as development costs tend to increase with complexity of the project. [2]. Over the years the main objective of researchers has been to develop appropriate models and prediction techniques to compute development effort (cost) for the project. This effort is actually an estimate which is carried out in access the amount of work required and schedule to carry out the project within specified resources, budget and time frame [3].

The best known cost model so far was developed by Boehm in 1981 called COCOMO (COnstructive COst MOdel). This model has three levels, namely basic COCOMO, intermediate COCOMO and detailed COCOMO [4, 5]. This model takes line of code (LOC) as an input and was based on a study of 63 projects ranging from 2K to 100K LOC. Results of using old coefficients of COCOMO I [4] and its modified version namely, COCOMO II [6] for developing software projects in this era of time to market environment may not be accurate in assessing software effort required to build the project. As a consequence, there is a need design optimization algorithm for accurate, precise and reliable effort estimation. Over the years many cost estimation models have been proposed with the aim of assessing accuracy of different approaches using neural networks, genetic algorithm, particle swarm optimization, bat algorithm and fire fly algorithm [7, 8, 9, 10, 11, 12, 13, 14] to estimate project cost, effort, development time, and productivity. These studies/approaches/models suggests that there is no "best solution or approach" for effort estimation as each algorithm or approach predicts difference in accuracy estimation in comparison to one another. Rather it strongly depends on the context of the given project [15] and thus different organizations can be benefited with different estimation approaches. These prediction techniques/algorithms can thus be helpful in predicting realistic values, expressed in terms of person-hours or money. They can also be used in preparing project plans, iteration plans, budgets, investment analyses, pricing processes etc. by software industry for project development.

In this paper we propose a new approach of effort estimation using bacterial foraging optimization algorithm (BFOA). Results of experimental study conducted shows that the proposed approach produces promising results in comparison to COCOMO model. These results are based on three evaluation criteria's namely, magnitude of relative error (MRE), mean of magnitude relative error (MMRE) and prediction(X) as performance measures. We have also conducted a study to compare our results with other existing approaches listed in literature on the same dataset for better comparison. Results show that COCOMO model and other existing approaches are less accurate in comparison to BFOA with MMRE as 0.16 and PRED(25) as 0.9. Thus BFOA can help software industry in predicting accurate and reliable values for planning and maintenance of software project.

The rest of the paper is organized as follows: Section II presents related work relevant to the field of effort estimation followed by a detailed discussion of proposed optimization algorithm, process model and a flow graph of the presented work in section III. An experimental study is conducted in section IV to compare goodness of our approach with COCOMO and other existing techniques. Finally, the presented approach is concluded in section V.

II. RELATED WORK

Since the ever first evolution of cost estimation model namely COCOMO [2] almost three decades ago, many researchers have proposed various cost estimation models to deal with several optimization problems. The wide spread can be seen in the area of neural networks [8, 9, 10, 16], fuzzy logic [17], image analysis [18, 19, 20, 21] and nature inspired algorithms [12, 13, 14, 22, 23, 24, 25, 26, 27] etc to provide good results in terms of performance. The proposed work is motivated by the work of several antecedent researches who have explored the possibility of using machine learning and nature inspired algorithms for optimizing COCOMO coefficients for better accuracy [7, 28, 29, 30, 31, 32, 33, 34, 34, 35,36, 37].

Y. Shan et.al. [28] uses grammar guided genetic programming with a data set of 423 software, to generate two grammar languages for effort gauge. They later compared their results with linear and log regression which shows genetic programming fits for complex functions. Koch and Mitlohner [4] estimated accuracy and weights for three datasets namely COCOMO, Albrecht and ERP using extended Genetic algorithm by deriving weights for effort computation. Researchers also applied genetic programming to depict evolution effort. Lin and Tzeng [29] uses COCOMO database for testing and hybrid model composed of one way analyze, K-means clustering and particles swarm optimization to estimate effort and compared the results using MMRE and prediction(X) as a measure. Khalifelu and Gharehchopogh [30] used NASA projects dataset to train and test the data mining techniques including LR, artificial neural networks (ANN), support vector regression (SVR) and k-nearest neighbor KNN. They showed that SVR was best model with less MMRE. MRE was further reduced to 0.1619 by Dizaji et.al. [7] when they applied bee colony optimization and compared their experiment results with intermediate COCOMO. Maleki et.al. [31] developed a hybrid approach of firefly and genetic algorithm for software cost estimation which as a result increased the accuracy by 2.88% as compared to COCOMO model. In addition to these, researchers have also explored three particular genetic algorithms namely, Differential Evolution (DE) [32], Particle Swarm Optimization (PSO) [33] and Artificial Bee Colony Optimization (ABC) [34] to solve difficult optimization problems.

We have used MRE, MMRE and prediction(X) as performance measures for comparison with other approaches listed above. Experimental study conducted claim that they are better than the approaches listed above with MMRE value as 0.16 and Pred(25) as 0.9.

III. PROPOSED WORK

The proposed approach targets on acquiring optimal values of organic COCOMO model coefficients and is motivated by the hunting characteristics of microorganisms called E.Coli bacteria. In 2002, K. Passino got inspiration from the food searching nature of swarm of E.Coli bacterium and proposed a global optimization algorithm by mimicking their behavior [38, 39]. Global optimization aims at finding maximum or minimum values in the input range as compared to regular optimization which focuses on finding local minima or maxima. This behaviour of global optimization of finding maximum or minimum values in the input range is of our interest as for any software development we can maintain optimization on the content we find relevant, like products, services, articles, and other forms of information.

Following section provides a brief introduction of Bacterial foraging optimization algorithm followed by discussion on framework and process model adopted for our approach. A detailed discussion on computation of effort estimation is provided in the latter half of the section.

A. *Brief introduction to Bacterial foraging optimization algorithm*

The population of bacteria has to undergo four phases namely chemotaxis, swarming, reproduction and elimination -dispersal in its lifetime to achieve an optimum nutrients level [40].

- Chemotaxis: Bacterial population selects a direction for food by analyzing their surroundings for nutrients gradient. They either maintain their previous direction (called swimming) or choose another random direction (called tumbling). Their decision to either tumble or swim is totally directed towards optimizing their energy as fast as possible. Mathematically it can be represented by (1).

$$pos_e(ch+1,r,el) = pos_e(ch,r,el) + step(e)\frac{RV(e)}{\sqrt{RV^T(e)RV(e)}} \tag{1}$$

where pose(ch+1, r, el) is position of eth bacterium in landscape, step(e) is the bacterium *e's* step size, RV is a random vector and *ch*, *r*, *el* are chemotactic, reproduction and elimination-dispersal indices respectively.

- Swarming behavior: Several bacterial species including Coli depict a fascinating swarm behavior by forming some special patterns (concentric figures) in their nutrients pool. When these bacteria come in contact with high energy nutrients, they secrete attractants and when they find some noxious substances, they release repellents. Hence, they move together towards the area with high nutrient density by forming coextensive patterns. This signaling behavior is called cell to cell attraction and mathematically calculated using (3) [41].

$$fitness_{c2c}(pos,Pop(ch,r,el)) = \sum_{e=1}^{B}\left[-d_a exp\left(-w_a\sum_{m=1}^{dim}(pos_m - pos_m^e)^2\right)\right] + \sum_{e=1}^{B}\left[h_r exp\left(-w_r\sum_{m=1}^{dim}(pos_m - pos_m^e)^2\right)\right] \tag{2}$$

In (2), fitnessc2c(pos,Pop(ch,r,el)) is the total cell to cell attraction effect, Pop(ch,r,el) denotes the entire bacteria population, pos is a position vector in the landscape, B is the swarm size and dim is the number of dimensions of search space.

- Reproduce: In line with the Darwin's theory, bacteria are sorted according to their fitness. Half of the bacterial population which is healthy participates in reproduction and each of them splits into two halves. On the other hand, less fit bacteria (half population) get extinct.
- Eliminate and disperse: After certain number of generations it may happen that a certain colony of bacteria

die due to natural environmental conditions or some group may move to a new location. To implement this phase mathematically, some bacteria are removed and some are assigned new positions in the landscape.

B. Proposed Framework and Process Model

The process starts with selection and interaction with the database. Once the parametric values are finalized, the next step is to apply the optimization algorithm for computation of fitness values for chosen parameters. For our approach we are using Bacterial Foraging Optimization algorithm. Figure 1 below depicts the various stages of the process model followed to compute effort for a given project. The whole process is discussed step by step in following section:

Step 1: The process starts with interaction of software engineer with project database. The dataset developed by Martin et.al. [42] is adopted to train and then test the proposed model. The chosen dataset consists of 41 projects which were written in Pascal language and went through all the phases of software development life cycle. To the best of our knowledge researcher have used Martin's dataset [42] to test their approaches and compare their results with others for accuracy and performance. We have also used the same dataset so that we can compare our results on the similar platform for better understanding.

With respect to each project from the dataset [42], researchers have calculated four parametric values namely, LOC, dhama coupling, McCab complexity and time taken to develop the project (TDev) to compute effort for a given project. The presented approach takes only two parameters namely, LOC and TDev (mentioned in Table 1 below) for cost estimation as compared to other earlier approaches mentioned in literature. This makes BFOA to achieve more speed convergence over other nature inspired algorithms listed in literature. The results of the presented approach depicts that it produces more accurate results with less computation time as our approach takes only half parameters into consideration for effort computation. Out of 41 modules in the dataset, we have used 30 modules for training and rest 11 for testing the model.

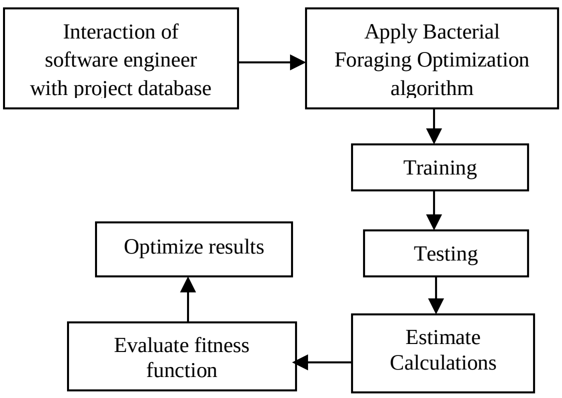

Figure1: Process Model

Table 1
Martin et.al. [39] dataset

Module number	LOC	TDEV
1	4	13
2	10	13
3	4	9
4	10	15
5	23	15
6	9	15
7	9	16
8	14	16
9	7	16
10	8	18
11	10	15
12	10	15
13	10	18
14	10	13
15	10	14
16	10	15
17	15	13
18	10	12
19	10	12
20	17	22
21	11	19
22	15	18
23	15	19
24	13	21
25	14	20
26	14	21
27	15	19
28	15	20
29	13	15
30	14	13
31	18	19
32	9	13
33	12	12
34	17	12
35	16	21
36	31	21
37	16	19
38	24	18
39	22	24
40	22	25
41	22	18

Step 2: The next step is to compute the fitness function. It is simply a measure defined as a function to identify how "fit" our how "good" the solution is with respect to the problem in consideration. To calculate its fitness, we use these coefficient values and evaluate the fitness function [40] for each project as stated in (3).

$$fitness_{ep} = \frac{(TDev_a^p - TDev_{bfoa}^{ep})}{(TDev_a^p)} \tag{3}$$

where, $fitness_{ep}$ denotes the fitness of eth bacteria specific to *pth* project, $TDev_a^p$ is the actual time taken to develop project p, $TDev_{bfoa}^{ep}$ is the time estimated by proposed model for project p by using bacteria e. The overall fitness of each bacterium is calculated by taking the average of fitness calculated for each project by that bacterium. This can be modelled mathematically by (4).

$$fitness_e = \frac{\sum_{p=1}^{P} fitness_{ep}}{P} \tag{4}$$

where, fitness is the fitness of bacterium *e* and P is the total number of projects. The position of each E.Coli bacterium in the swarm renders one possible combination of COCOMO coefficients.

Step 3: After computing the fitness function we apply bacterial foraging based optimization scheme for cost optimization foraging based optimization scheme. Following steps are followed for cost computation:

A. Parameter setting

The process starts with determining the parameters required by bacterial foraging algorithm namely, swarm size B, dimension of landscape dim, number of chemotactic steps *Nc*, step size of bacterium step, number of reproduction steps *Nr* and count of elimination & dispersal events Ned. The values chosen in this scheme are given in table 2 below. We are using the same values as given by [40] to compute cost of a given project.

Table 2
Parameter tuning values for BFOA [37]

Parameter	Value
B	50
Dim	4
N_c	10
Step	2
N_r	5
N_{ed}	2

B. Bacterium structure and formation

Each bacterium in the population has 4 dimensions, where each dimension corresponds to one of the four COCOMO model coefficients as shown in figure 2 below. All E.Coli are placed randomly in the search landscape so that solutions are picked from different areas of search space with equal probability.

C. Chemotaxis

For each bacterium fitness is calculated at their current position using equation (4) & (5). After evaluating their fitness, bacteria perform swim action and are moved to new positions by using (2). If the fitness at new position is better than previous fitness, it continues to swim, otherwise it tumbles.

D. Swarming

All solutions in the population secrete repellents or attractants based on their fitness values. This cell to cell attraction in swarming is achieved using (3). Parameters used in equation are set to values shown in table 3 [38]. Step C and D are repeated N_c times.

E. Reproduction

Less fit solutions in the landscape are deleted and more fit are used to reproduce. If the number of reproduction steps completed reach N_r; go to step F, otherwise goto step C.

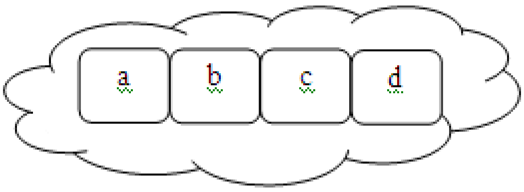

Figure 2: Bacterium Structure

Table 3
Parameter Setting

Parameter	Value
Depthattractant(d_a)	0.1
Widthattractant(w_a)	0.2
Heightrepellant(h_r)	0.1
Widthrepellant(w_r)	10

F. Eliminate and disperse

The entire group of E.Coli may face N_{ed} elimination dispersal events in their lifetime. If number of such events occurred till now has reached N_{ed}, terminate the algorithm, else go to step C. The whole process is summarized in the form of the flow graph shown in Figure 3 below. The algorithm stops on completion on elimination dispersal events, denoted by stop oval in the figure. The flowchart depicting fitness function calculations is given in Figure 4. It returns the fitness value calculated as explained in step 2.

IV. EVALUATION AND RESULT ANALYSIS

To analyze the results of the proposed approach, we have used three evaluation criteria's namely, Magnitude of relative error (MRE), Mean Magnitude of relative error (MMRE) and Prediction(X) for efficiency measurement. To conduct this experiment, the population size is initialized to 50. We have used the same values of tuning parameter as given by [40] to compute cost of a given project. The process starts with interacting with dataset, as stated above we have used Martin's [42] so that we can compare our results with other approaches. It follows all steps given in section 3 above. For each run, the algorithm generates different COCOMO coefficients values. Only those results are selected and used which generate best fitness. In this experimental study we have calculated the various computations for COCOMO model as well for step by step result comparison. Result comparison with other approaches on three evaluation criteria's is presented in section 4.2 below. Table 4 below presents the coefficient values obtained using our approach and Table 5 presents the comparison of development time estimated by proposed algorithm ($TDev_{bfoa}$) for test data with actual development time ($TDev_a$) and development time estimated by COCOMO model ($TDev_{Cocomo}$).

A. Result Observation and comparison with COCOMO model on chosen efficiency parameter:

We have used three evaluation criteria's namely, MRE, MMRE and Prediction(X) for efficiency measurement. A brief discussion on each of these along with result computation for proposed and COCOMO model is presented in following section.

- *Magnitude of relative error (MRE)*: MRE is a measure of deviation i.e. difference between and estimated effort relative to the actual effort for a given project. The mean takes into account the numerical value of every observation in the data distribution, and is sensitive to individual predictions with large MREs [27]. It is computed using the formula given in equation (5).

$$MRE = \frac{estimatedvalue - actualvalue}{actualvalue} \quad (5)$$

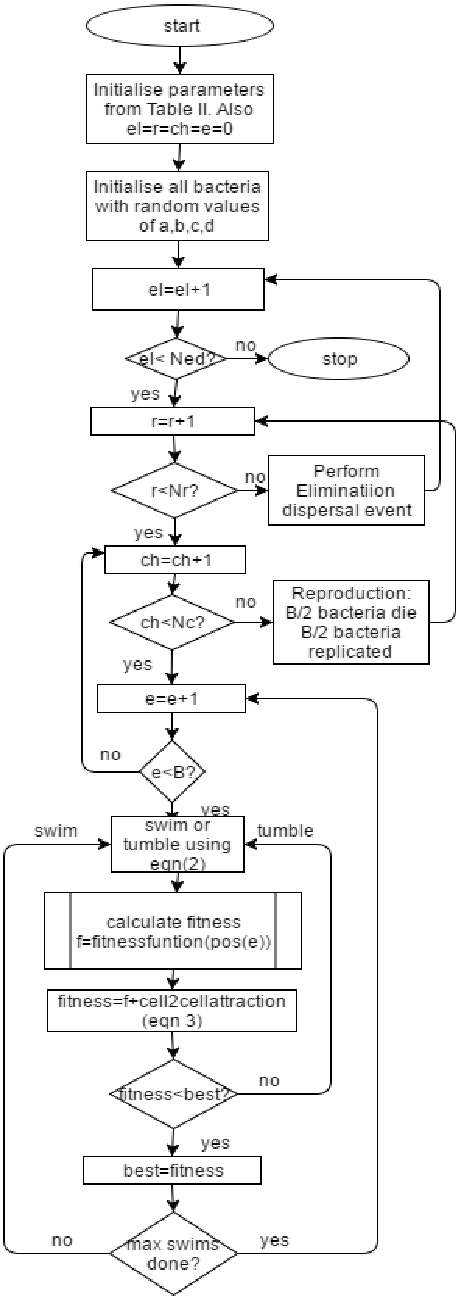

Figure 3: Flow graph of proposed approach

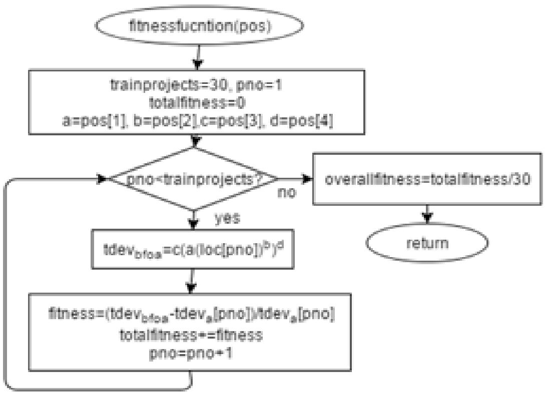

Figure 4: Flow graph of fitness function

Table 4
COCOMO and BFOA optimal coefficient values

Coefficient	COCOMO model	BFOA model
a	2.4	2.5703
b	1.105	0.9570
c	2.5	2.6036
d	0.38	0.5245

Table 5
Comparison of development time

S.No.	$TDev_a$	$TDev_{Cocomo}$	$TDev_{bfoa}$
1	15	10.24	15.48
2	13	10.56	16.07
3	19	11.74	18.23
4	13	8.77	12.87
5	12	9.9	14.87
6	12	11.46	17.71
7	21	14.74	23.94
8	19	11.17	17.18
9	18	13.24	21.06
10	25	12.77	20.16
11	12	9.17	13.57

Table 6 presents value of MRE calculated for each project for test data [14], for proposed approach and COCOMO model.

Table 6
Comparison of MRE values for COCOMO and BFOA

S. No	LOC	$TDev_a$	MRE_{COCOMO}	MRE_{BFOA}
1	13	15	0.32	0.03
2	14	13	0.19	0.24
3	18	19	0.38	0.04
4	9	13	0.33	0.01
5	12	12	0.18	0.24
6	17	12	0.05	0.48
7	31	21	0.3	0.14
8	16	19	0.41	0.1
9	24	18	0.26	0.17
10	22	25	0.49	0.19
11	10	12	0.24	0.13

- *Mean Magnitude of relative error (MMRE):* It is the average of MRE values of all tuples in the test set [44] given in (6). The corresponding values are given in Table 7 below.

$$MMRE = \sum_{n=1}^{N} MRE_n \tag{6}$$

- *Prediction(X)*: This analysis measure gives an idea of how many predictions lie within X% of the actual value [41]. We have considered 10, 20 and 25 as values of X to capture the response value of predication. Table 7 presents the values of MMRE and PRED(10), PRED(20) and PRED(25) (where PRED(10), PRED(20) and PRED(25) represents Prediction at 10, 20 and 25 respectively).

Table 7
MMRE & Prediction Values

Criteria	COCOMO	BFOA
MMRE	0.28	0.16
PRED(10%)	0.09	0.27
PRED(20%)	0.27	0.72
PRED(25%)	0.36	0.9

Hence it can be seen clearly that BFOA works better than COCOMO model and gives much less value of MMRE than COCOMO.

B. Result comparison with other approaches:

Result observation of BFOA and other algorithms given in Table 8 below, clearly shows that BFOA works better than other approaches with MMRE value of 0.16 which is much lower than other approaches and PRED(25) as 0.9. Thus, it can be concluded that the proposed model is very beneficiary for effort estimation and could estimate the effort better in comparison to the various models.

Table 8
Results of comparison with other approaches

Criteria	COCOMO	BFOA	ANN	FNN	FGRA	BAT
MMRE	0.28	0.16	0.37	0.22	0.232	0.2337
PRED(25%)	0.36	0.9	0.40	0.75	0.667	

V. CONCLUSION

This study successfully combines the stochastic and regression analysis. The exploitation of BFOA for optimizing COCOMO coefficients fairly works well in estimation accuracy for software development time taken. Overall, we think that this new leaf of nature inspired algorithm offers advantage as the proposed methodology uses the concept of bacterium generating new optimized coefficients which can meet the expectation of IT companies for unerring predicting the project feasibility in terms of time and cost constraints.

REFERENCES

[1] Sharma, D.S. Kushwaha, “Estimation of Software Development Effort from Requirements Based Complexity”, Procedia Technology, Vol. 4, pp. 716-722, (2012).

[2] M. Jørgensen, “Contrasting ideal and realistic conditions as a means to improve judgment-based software development effort estimation”, Information and Software Technology, Vol. 53, Issue 12, pp. 1382-1390, Elsevier B.V, (December 2011).

[3] P.C. Pendharkar, “Probabilistic Estimation of Software Size and Effort”, Expert Systems with Applications, Vol. 37, Issue 6, pp. 4435-4440, Elsevier Ltd, June (2010).

[4] B.W. Boehm, Software Engineering Economics, Prentice-Hall, EnglewoodCli4s,NJ, 1981

[5] Zhiwei Xu, TaghiM. Khoshgoftaar, Identification of fuzzymodels of softwarecost estimation,FuzzySets and Systems 145 (2004) 141–163

[6] Zhiwei Xu, TaghiM. Khoshgoftaar, Identification of fuzzymodels of softwarecost estimation, FuzzySets and Systems 145 (2004) 141–163

[7] Zahra Ashegi Dizaji, Reza Ahmadi, Hojjat Gholizadeh and Farhad Soleimanian Gharehchopogh, “Article: A Bee Colony Optimization Algorithm Approach for Software Cost Estimation”, *International Journal of Computer Applications* 104(12):41-44, October 2014.

[8] Idri, A. Khoshgoftaar, T.M. Abran, A., “Can neural networks be easily interpreted in software cost estimation?,” Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE’02, Vol.: 2, 1162- 1167, 2002.

[9] Heiat, A., “Comparison of artificial neural network and regression models for estimating software development effort”, Information and Software Technology 44 (15), 911–922, 2002.

[10] Srinivasan, K., Fisher, D., “Machine learning approaches to estimating software development effort”, IEEE Transactions on Software Engineering, vol. 21 (2), 126–137, 1995.

[11] Wittig, G., Finnie, G., “Estimating software development effort with connectionist models”, Information and Software Technology, 39 (7), 469– 476, 1997.

[12] Yahya Rahmat-Samii “Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in Engineering Eelectromagnetics” IEEE Applied Electromagnetics and Communications, 2003. ICECom 2003,2003

[13] Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proc. IEEE Int’l. Conf. on Neural Networks, IV, 1942–1948. Piscataway, NJ: IEEE Service Center ,1995

[14] Shi, Y. Eberhart, R. “A modified particle swarm optimizer” IEEE Evolutionary Computation Proceedings 1998 ,1998

[15] Shepperd, M. Kadoda, G. "Comparing software prediction techniques using simulation" IEEE Transactions on Software Engineering - Special section on the seventh international software metrics symposium, Volume 27(11), pp.- 1014-1022, November 2001

[16] F.S. Gharehchopogh, “Neural Networks Application in Software Cost Estimation: A Case Study”, 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), pp. 69-73, IEEE, Istanbul, Turkey, (15-18 June 2011).

[17] A. Mittal, K. Parkash, H. Mittal, “Software Cost Estimation Using Fuzzy Logic”, ACM SIGSOFT Software Engineering, Vol. 35, No. 1, pp. 1-7, (2010).

[18] Chen Wei; Fang Kangling (2008). Multilevel thresholding algorithm based on particle swarm optimization for image segmentation.27th Chinese Control Conference (CCC 2008), pp.348-351.

[19] E. Cuevas, D. Zaldívar and M. Pérez-Cisneros (2010).A novel multi-threshold segmentation approach based on differential evolution optimization.Expert Syst. Appl. 37(7):5265-5271.

[20] D. Karaboga (2005). An idea based on honey bee swarm for numerical optimization, Technical report,-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.

[21] L. Chih-Chih (2006) A Novel Image Segmentation Approach Based on Particle Swarm Optimization, IEICE Trans. Fundamentals, 89(1), 324-327.

[22] Maleki I., Ghaffari A., Masdari M., “A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimiza-tion”, International Journal of Innova-tion and Applied Studies, 5(1), 72-81, 2014.

[23] Gharehchopogh F.S., “Neural Networks Application in Software Cost Estimation: A Case Study”, In-ternational Symposium on Innovations in Intelligent Systems and Applica-tions (INISTA 2011), pp. 69-73, IEEE, Istanbul, Turkey, 15-18 June 2011.

[24] Khalifelu Z.A., Gharehchopogh F.S., “Comparison and Evaluation Data Mining Techniques with Algo-rithmic Models in Software Cost Es-timation”, Elsevier, Proce-dia-Technology Journal, Vol. 1, pp. 65-71, 2012.

[25] de A. Araújo Ricardo, Soares S., Oliveira A.L.I., “Hybrid Morphologi-cal Methodology for Software Devel-opment Cost Estimation”, Expert Sys-tems with Applications, Vol. 39, pp. 6129-6139, 2012.

[26] Oliveira A.L.I., Braga P.L., Li-ma R.M.F., Cornélio M.L., “GA-based Method for Feature Selection and Pa-rameters Optimization for Machine Learning Regression Applied to Soft-ware Effort Estimation”, Information and Software Technology, Vol. 52, pp. 1155-1166, 2010.

[27] Sheta A.F., Ayesh A., Rine D., “Evaluating Software Cost Estimation Models using Particle Swarm Optimi-zation and Fuzzy Logic for NASA Projects: a Comparative Study”, In-ternational Journal Bio-Inspired Computation, 2(6), 365-373, 2010.

[28] Shan, Y. ,McKay, R.I. ,Lokan, C.J. ,Essam, D.L. , “Software Project Effort Estimation Using Genetic Programming”, IEEE Communications, Circuits and Systems and West Sino Expositions vol. 2 ,pp.1108-1112 , 2002

[29] Lin, J.-C, " Applying Particle Swarm Optimization to Estimate Software Effort by Multiple Factors Software Project Clustering.", IEEE, 2010.
[30] Khalifelu Z.A., Gharehchopogh F.S., "Comparison and Evaluation Data Mining Techniques with Algorithmic Models in Software Cost Estimation",Elsevier,Procedia-Technology Journal, Vol. 1, pp. 65-71, 2012.
[31] Isa Maleki, Laya Ebrahimi, Farhad Soleimanian Gharehchopogh, "A Hybrid Approach of Firefly and Genetic Algorithms in Software Cost Estimation", Vol. 2(6), PP. 372-388, 2016.
[32] R. Storn and K. Price (1995).Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report.
[33] J. Kennedy, R. Eberhart (1995), Particle Swarm Optimization, From Proc. IEEE Int'l. Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV:1942-1948.
[34] D. Karaboga (2005). An idea based on honey bee swarm for numerical optimization, Technical report,-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department.
[35] Suharjito, S. Nanda, B Soewito, "Modeling Software Effort Estimation Using Hybrid PSO-ANFIS", International Seminar on Intelligent Technology and Its Application, IEEE, pp. 219-224, 2016.
[36] R Sachan, A Nigam, A Singh, S Singh, M Choudhary, A Tiwari and D. S. Kushwaha, "Optimizing Basic COCOMO Model using Simplified Genetic Algorithm", Procedia Computer Science 89 , Elsevier, pp. 492 – 498, 2016.
[37] Kumari S., Pushkar S., " Software Cost Estimation Using Cuckoo Search", Advances in Computational Intelligence, vol 509, Springer, pp. 167-175, 2017.
[38] Passino M.Kevin, "Biomimicry of Bacterial foraging for distributed optimization and control", IEEE Control System Magazine (S0272-1708), june 2002, pp. 52-67.
[39] Passino M.Kevin, "Bacterial foraging Optimization", International journal of swarm intelligence Research, 1(1),Jan-March 2010,pp.1-16.
[40] Liu XiaoLong, Li RongJun, YangPing, "A Bacterial Foraging Global Optimization Algorithm Based On the Particle Swarm Optimization", IEEE, (978-1-4244-6585-9), 2010, pp. 22-27.
[41] Sotirios P. Chatzis, Spyros Koukas, "Numerical optimization using synergetic swarms of foraging bacterial populations", Elsevier Expert Systems with Applications, vol. 38, Issue 12, pp. 15332-15343, Nov. Dec.,2011.
[42] C.L. Martin, J.L. Pasquier, M.C. Yanez, T.A. Gutierrez, "Software Development Effort Estimation Using Fuzzy Logic: A Case Study", IEEE Proceedings of the Sixth Mexican International Conference on Computer Science (ENC'05), 2005, pp. 113-120.
[43] A.Galilina, O. Burceva, S. Parshutin, "The optimization of COCOMO Model coefficients using Genetic Algorithms",Information Technology and management science, doi: 10.2478/v10313-012-0006-7,2012/15.
[44] M. Shin and A. L. Goel, "Emprirical data modeling in software engineering using radial basis functions," IEEE Trans. on Software Engineering, pp. 567-576, 2000.