A New Approach in Software Cost Estimation with Hybrid of Bee Colony and Chaos Optimizations Algorithms

Farhad Soleimanian Gharehchopogh[1] and Zahra Asheghi Dizaji[2]

[1]Department of Computer Engineering, Hacettepe University, Beytepe, Ankara, Turkey
[2]Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

Abstract: *One of the most important aspects of managing software projects is estimating the cost and time required for developing information systems. In this paper, we have used Bee Colony Optimization, Artificial Bee Colony, and hybrid of bee colony and Chaos Optimization Algorithm (COA) to Software Cost Estimation (SCE), and tenet mapping as the COA and NASA dataset for test and training data. The results obtained by this method have been compared with the COCOMO II model. The results of this comparison indicate a reduction of absolute relative error of the estimated cost up to 0.07.*

Key words: Bee Colony Optimization, Artificial Bee Colony, Chaos Optimization Algorithm, Software Cost Estimation

1. INTRODUCTION

The purpose of SCE is to estimate the cost and time required for software development before starting the project which continues until the end of production and system development [1, 2]. SCE is one of the key concerns of project management for the production of information systems. SCE patterns which estimate the cost of system in the early stages of construction with little information available about the project are useful and necessary [3, 4]. Appropriate method of cost estimation provides the possibility to effectively control the time and cost of system construction [5].

A software project evolves gradually; therefore a definite and clear estimate of software cannot be reached [6, 7]. Currently software and estimating models do interval estimation instead of point estimation and therefore correlate to the new software development methods. However, software project estimation can be converted into a series of systematic steps that provide estimates with acceptable risk. Proposed algorithm methods for SCE use mathematical models to estimate project costs. Each algorithm model is defined as a function of cost factors. Algorithm models differ in two ways: a: selection of cost factors, b: definition of function for cost calculation. COCOMO is the most famous and most documented algorithm model which was proposed by Barry W. Boehm in 1981 [8]. COCOMO model is used for SCE and required time for system developments [20]. However, due to the increasing volume and complexity of software projects using model-based methods are less accurate for estimation. Therefore in recent years several studies have been carried out to use non-algorithm models such as machine learning models as a replacement for model-based methods. Therefore in this paper we have used bee colony (BCO, ABC) and hybrid of bee colony and COA to estimate the cost of software project.

At the end of 1970s model based techniques have been proposed like SLIM model by Putnam and Myers in1992 [9], checkpoint model by Jones in 1997 [10], PRICE-S model by Park in 1988[9] and COCOMO model by Boehmin1981 [8]. COCOMO is the most documented and transparent model for cost estimation. Basically in this method, lines of

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)

code required for software production is estimated on the concept of Function point and the size of required works are estimated based on that Gharehchopogh and Khalifehlou have used regression model to classify the NASA data set and calibrating the COCOMO model parameters. They have evaluated the performance of the proposed model by comparing it with the standard COCOMO model and the results indicate improved performance of the proposed model [12]. I.Maleki et.al have used a hybrid of ant colony algorithm and genetic algorithm for SCE [13]. These researchers have used genetic algorithm for testing and ant colony algorithm for training and Mean Absolute Relative Error (MARE) to evaluate the proposed method. This hybrid algorithm has been analyzed and evaluated on NASA data sets and the results shows that the model has a better performance than the COCOMO model. Artificial Neural Networks (ANNs) have been used for SCE in [14]. In this work, 11 projects from 60 projects on NASA data sets were analyzed using ANNs [15] and the results were compared to COCOMO models. Evaluating the performance of the proposed method determined that the COCOMO model had more errors than the ANN in many cases. Results indicate that in over 90% of the cases ANNs had better estimations than the COCOMO model. Researchers [16] have used FL model for estimating the software projects. They have announced the cost estimation of software projects as one of the most challenging and important activities in software development. Their proposed method shows that FL model can be used in software development. They have used 14 projects from the KEMERE set. Based on the obtained results the MARE and PRED (N) are better in proposed methods compared to the algorithm methods. Cost function has too many parameters in software projects. Some of the factors that directly affect the cost estimates are: Line of Code (LOC) and Kilo LOC (KLOC).

2. Fundamental Concepts

In this section, we will discuss the COCOMO II, BCO, ABC and COA respectively.

2.1. COCOMO Model

COCOMO is the most documented and transparent model for SCE. Fifteen Cost Drivers have been emphasized in COCOMO for better results. To measure the work and time which is the relationship between the size, Cost Drivers and work and also between the work and time COCOMO uses a series of formulae which are obtained from the historical data of the completed projects, and then the effect of cost drivers on the work is achieved [17].

In COCOMO II cost estimation is done by formula (1) [17, 18].

Formula (1):$PM = a * (size)^b * \prod_{i=1}^{15} EM_i$

In formula 1, parameters (a) & (b) are constant, and their quantity depend on the data available in the data set. The parameter size is the project size in KLOC. Parameter Effort Multipliers(EM) is the coefficient which causes increase or decrease in the rates of effort in person/month[15].In COCOMO II a, b, & c parameters are initialized according to Table(1)[6].

Table. 1. Values of Constant Parameters in the COCOMO II Model

Class of Projects	a	b	c
Organic	2.4	1.05	2.5
Semidetached	3.0	1.12	2.5
Embedded	3.6	1.20	2.5

The organic class contains relatively small projects that are conducted by highly experienced teams. Usually if the project size is 100 KLOC they are placed in the organic

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)

class. Semidetached class includes average projects which are neither complex nor simple, and have a size of 100 to 300 KLOC. Embedded class includes projects with a size of more than 300 KLOC. This class is used when the hardware and operations are previously defined and do not require any changes.

2.2. ABC Algorithm

ABC algorithm was introduced by Karaboga to optimize the mathematical functions. In the ABC, bees are divided into three groups: Employed Bees (EB), Onlooker Bees (OB) and Scouts Bees (SB). Bees which remain in the dance area for making the decision to choose a food source are called OB. Bees which move to a specified food source are called EB .Bees which are doing a random search are called SB[16].

In ABC algorithm, each cycle of search consists of three stages:

1. Sending EB to food sources and then measuring the amount of their nectar.
2. Selecting food sources by OB and sharing the information by EB and determining the amount of nectar in food sources.
3. Specifying the SB and sending them to new food sources.

Onlooker bee selects a food source according to the associated amount of probability to that food source, which is calculated by formula (2) [19].

Formula (2):$\frac{fit_i}{\sum_{j=1}^{n} fit_j}$

Where fit_i(fitness of solution i) is evaluated in accordance to the amount of nectar in food source by its employed bee. And N is the number of food sources which is equal to the number of EB. In this method, in order to produce a pre-selected food source EB share their information with OB. And for this exchange formula (3) can be used: [19]

Formula (3):$V_{i,j} = X_{i,j} + \varnothing_{i,j} * (X_{i,j} - X_{k,j})$

Where the k, i ∈ {1, 2, . ., EB} and j ∈ {1, 2, . .,D} index are selected randomly. $\varphi_{i,j}$ is a random number between [-1, 1], which controls the situation of neighbor food source around $x_{i,j}$ and the comparing changes of neighboring food source are presented visually by the bee.

2.3. BCO Algorithm

The BCO algorithm have been introduced by Teodorovich. When moving in space, our artificial bees are doing one these movements: "moving forward" or "moving backward". When "moving forward", bees find new ways and methods to solve the problem. They do this by the help of some personal searches and data obtained earlier. After that, bees do the "moving backward" act which is getting back to the main colony. In the colony all the bees participate in a process of "decision making". Based on the new information obtained about the quality of the solution, bee can decide to: [21]

1. Abandon its own source and look up in the dance hall for someone who has a source with better quality
2. Without attracting anyone, go back to its own solution source
3. Gather other bees around him by performing specific movements (dances) in the dance hall.

The probability of selecting a solution by bees is calculated by formula (4) [21]

Formula (4)$V_j = \frac{Max\,(F) - F_j}{Max\,(F) - Min\,(F)} \quad ,J = 1,2, \dots, M$

Where M specifies the number of solutions, F specifies all the solutions and F_j specifies the current solution. To specify the type of the bee in this algorithm a random number and a number produced by formula (5) and formula (6) is used. If the produced number by this formula is bigger than the random number, the bee doesn't look for a new solution. But if it is smaller than the random number it randomly selects one of the solutions from the available solutions.

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)

Formula (5): $O_b = \frac{C_{max} - C_b}{C_{max} - C_{min}} \quad , b = 1,2, \dots , B$

B parameter specifies the number of the bee and *C* indicates the popularity of the solution.

Formula (6): $P_b^{u+1} = e^{\frac{O_{max} - O_b}{U}} \quad , b = 1,2, \dots \dots , B$

B parameter specifies the number of the bee and *U* indicates the moving forward steps.

2.4. COA

Chaos theory was first proposed by Henri Poincare in 1890. Chaos is a long-term non-periodic behavior in a nonlinear system which is highly dependent on initial conditions [22]. The mapping used, in this paper, is Tenet mapping. Formula (7) is used to implement this mapping.

Formula (7): $x_{n+1} = \begin{cases} rx_n , & 0 \leq x_n \leq \frac{1}{2} \\ r - rx_n, & 1/2 \leq x_n \leq 1 \end{cases}$

3. Proposed Method

Estimating the cost of software projects plays an important role in software development and considering the fact that in algorithm models no values are defined for constant variables and mean values are determined, so it is not easy to find a reliable solution. Therefore in this paper we have tried to find these values according to Bee Colony (BCO, ABC) and a hybrid of Bee Colony and COA. Various factors affect the cost estimation for software projects. One of the effective factors is the type of project. In this thesis the project type factor is used for initial classification. After initial classification, the training data are provided to proposed intelligent algorithms for predicting the values of the constant variables. And after the completion of training and obtaining the values of the variables, obtained values are applied to testing data and the costs of the software projects have been estimated. It should be noted that the initial values for these variables are according to Table (1).

3.1. ABC Algorithm

ABC Algorithm is the first proposed algorithm method. The operation of this algorithm is shown in Table (2):

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)

Inputs: NASA datasets (including affective factors in estimation, project type and number of available projects) **Outputs**: values for constant parameters in the COCOMO model and classified data. **Step 1**: reading data on data set. **Step 2**: separating training and testing data. **Step 3**: classifying training and testing data. **Step 4**: calling ABC for each class. **Step 5**: determining the number of food sources and the number of bees. **Step 6**: initializing the bees, evaluate the performance of each bee and initialize the constant parameters in COCOMO model. **Step 7**: generating new solutions for each employed bee with formula (3). **step 8**: evaluate the performance of Fitness function, assess the performance of Fitness function, Fitness function is the MARE ,the goal is to minimize MARE by selecting appropriate values from the specified range. **Step 9**: apply greedy selection. **Step 10**: For each onlooker bee a bee is selected randomly. **Step 11**: based on selected bee and current onlooker bee a new solution is presented. **Step 12**: evaluating the performance of Fitness function, assessing the performance of Fitness function, Fitness function is the MARE, the goal is to minimize MARE by selecting appropriate values from the specified range. **Step 13**: applying greedy selection. **Step 14**: abandoned solutions are replaced by a new solution which is generated randomly by scout bee. **Step 15**: saving best solution and repeating step seven to fifteen. **Step 16**: getting the parameter values from best bee. **Step 17**: finishing the ABC algorithm.

Table. 2. ABC algorithm operation

3.2. ABC and Chaos

The next algorithm is the hybrid of ABC and COA. For hybrid chaos and bee colony algorithms formula (8) is used.

Formula (8): $V_{i,j} = X_{i,j} + CM_1 * \left(X_{i,j} - X_{k,j}\right) * CM_2$

∅ parameters values of ABC in formula(3) are put in formula(7) by chaos optimization algorithm(tenet mapping) that compound formula is shown in formula(8) and CM_1 and CM_2 are added as variables.

3.3. BCO Algorithm

The next algorithm is from the proposed BCO algorithm. The procedure of the algorithm is shown in Table (3):

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)

Table. 3. BCO Algorithm operation

Inputs: NASA data sets (including affective factors in estimation, project type and the number of available projects).
Outputs: values for the constant parameters in the COCOMO model and classified data.
Step 1: reading data on the data set.
Step 2: separation of training and testing data.
Step 3: classification of training and testing data.
Step 4: calling BCO for each class.
Step 5: initializing the bees, evaluating the performance of each bee and initializing the constant parameters in COCOMO model.
Step 6: selecting best bee.
Step 7: repeat the following operations for each bee.
Step 8: repeat the following operations according to the number of steps forward.
Step 9: edit the solution for each bee according to the number of changes in each step.
Step 10: evaluate all the solutions produced for each bee and evaluate the performance of Fitness function, assessing the performance of fitness function, Fitness function here is the MARE, the goal is to minimize MARE by selecting appropriate values from the specified range.
Step 11: select solution for each bee from available basic solutions using roulette wheel.
Step 12: assign the selected solution to the bee and determine the loyalty i.e.
Fitness function for each bee and select best bee.
Step 13: specify type of bee (loyal or disloyal) according to formula (4).
Step 14: if a bee was disloyal replace its solution with a loyal bee's solution according to formula (5) and formula (6).
Step 15: repeat step seven to fifteen.
Step 16: get parameter values from best bee.
Step seventeen: finish the BCO algorithm procedure.

3.4. BCO and Chaos

The next algorithm from the proposed method is the hybrid of the bee colony and COAs. To hybrid the chaos and bee colony formula (9) is used.

Formula (9): $V_{i,j} = X_{i,j} + \varnothing_{i,j} * (X_{i,j} - X_{k,j}) * CM_1$

$\varnothing$Parameters values bee colony algorithm in formula (3) are put in formula (7) by COA (tenet mapping) the hybrid formula is shown in formula (9) and the variables CM_1.

3.5. Proposed Method

The proposed method in this paper is applied by getting the optimal solutions produced by the algorithms discussed earlier and selecting the best solution. The process is shown in Figure (1).

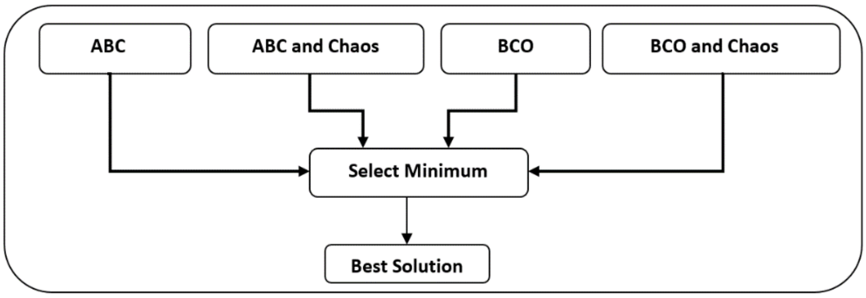

Figure 1: proposed method procedure

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)

4. Evaluating Results

In this paper, SCE are suggested by Hybrid of Bee Colony and Chaos Optimizations Algorithms. Data set used consists of 60 NASA data sets and from these 60 data set 80% are considered as training data and 20% are considered as test data. It should be mentioned that comparison is on the basis of MARE. This error is shown by formula (10) and formula (11).

Formula (10):$\mathrm{MARE_i} = \frac{|\mathrm{Actual}_{\,i} - \mathrm{Estimate}_{\,i}|}{\mathrm{Actual}_{\,i}}$

Formula (11):$MMARE = \frac{1}{N}\sum_{i=1}^{N} MRE_I$

Based on the fact that the proposed method selects the most optimal solution, therefore according to the figure (2), figure (3) and table (4). This model is more optimal than all the models presented in this paper.

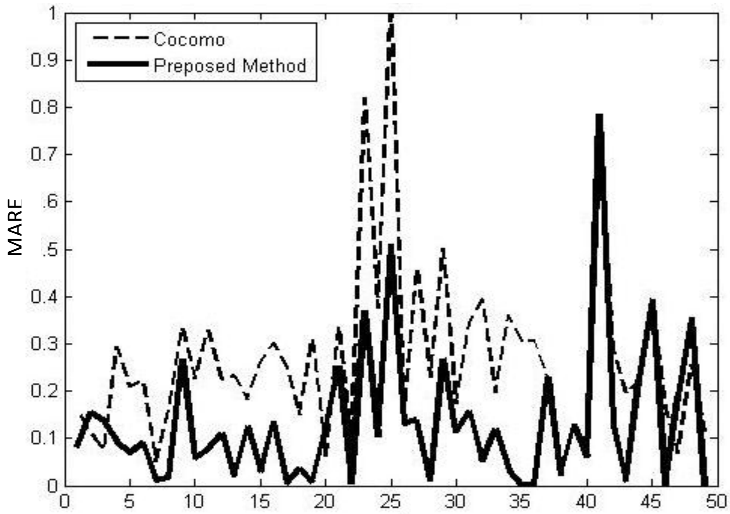

Figure 2: Comparison of the proposed method with COCOMO model is based on MARE on training data

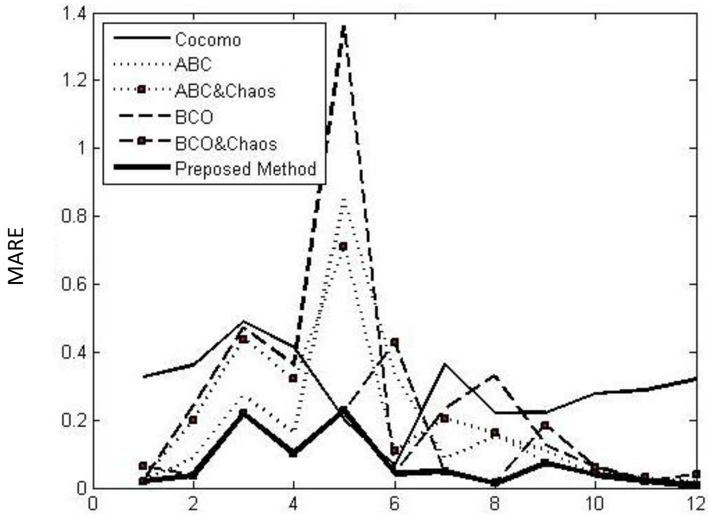

Figure. 3: Comparison of the proposed method with COCOMO model is based on MARE on test data

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)

Table. 4. MARE on test data

Model Name	MARE
COCOMO	0.2952
ABC	0.1925
Chaos_ABC	0.1800
BCO	0.2538
Chaos_BCO	0.1201
Proposed Method	0.07

5- Conclusion

In this paper, we use bee colony and hybrid algorithm causes an improvement in cost estimation for software projects in comparison to the COCOMO model based on MARE. The MARE produced by the COCOMO model for test data set is 0.2952% and the error for the same data set produced by the proposed method is lowered to 0.07%. Hence the proposed method is more optimal than the COCOMO model.

REFERENCES

1. Gharehchopogh, F.S. Talebi, A. Maleki, I. Analysis of Use Case Points Models for Software Cost Estimation, International Journal of Academic Research, Part A, 6(3): 118-124, 2014.
2. Gharehchopogh, F.S. Ebrahimi, L. Maleki, I. Gourabi, S.J. A Novel PSO based Approach with Hybrid of Fuzzy C-Means and Learning Automata in Software Cost Estimation, Indian Journal of Science and Technology, 7(6): 795-803, 2014.
3. Gharehchopogh, F.S. Maleki, I. Khaze, S.R. A Novel Particle Swarm Optimization Approach for Software Effort Estimation, International Journal of Academic Research, Part A, 6(2): 69-76, 2014.
4. Maleki, I. Ebrahimi, L. Jodati, S. Ramesh, I. Analysis of Software Cost Estimation Using Fuzzy Logic, International Journal in Foundations of Computer Science & Technology (IJFCST), 4(3): 27-41, 2014.
5. Gwang-Hee Kim, Sung-Hoon An , Kyung-In Kang, Comparison Of Construction Cost Estimating Models Based On Regression Analysis, Neural Networks, And Case-Based Reasoning, Building and Environment, Vol. 39, pp. 1235-1242, 2004.
6. F.S. Gharehchopogh, I. Maleki, S. Sadouni, Artificial Neural Networks Based Analysis of Software Cost Estimation Models, MAGNT Research Report, 2 (6): 597-505, 2014.
7. Maleki, I. Ebrahimi, L. Gharehchopogh, F.S. A Hybrid Approach of Firefly and Genetic Algorithms in Software Cost Estimation, MAGNT Research Report, 2(6): 372-388, 2014.
8. Boehm, B.W. Software Engineering Economics, Prentice-Hall, Englewood Cliffs, New Jersy, 1981.
9. Putnam, L. and Myers, Measures for Excellence, Yourdon Press Computing Series, 1992
10. Jones, Applied Software Measurement, McGraw Hil, 1997
11. Park, The Central Equations of the PRICE Software Cost Model, COCOMO Users' Group Meeting, November,1988
12. Khalifehlou, Z.A. Gharehchopogh, FS. A New Approach in Software Cost Estimation Using Regression Based Classifier, World Conference on Innovation and Computer Sciences, pp:352-356, 2012
13. Maleki, I. Ghaffari, A. Masdari, M. A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization,

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)

International Journal of Innovation and Applied Studies, 5(1): 72-81, 2014

14. Gharehchopogh, F.S. Neural Networks Application in Software Cost Estimation: A Case Study, International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011), pp:69-73, June 2011
15. Menzies, T. Port, D. Chen, Zh. Hihn, J. Validation Methods for Calibrating Software Effort Models, ICSE ACM USA, 2005
16. Mittal, A. Parkash, K. Mittal, H. Software Cost Estimation Using Fuzzy Logic, ACM SIGSOFT Software Engineering, 35(1): 1-7, 2010
17. Khalifelu, Z.A. Gharehchopogh, FS. A Survey of Data Mining Techniques in Software Cost Estimation", AWER Procedia Information Technology & Computer Science Journal, Vol: 1, pp. 331-342, 2012.
18. Karaboga, D. Akay, B. A Survey: Algorithms Simulating Bee Swarm Intelligence; Artificial Intelligence Review; 31 (1): 68-85, 2009.
19. Gao, W. Liu, S. Improved artificial bee colony algorithm for global optimization, Information Processing Letters 111, pp: 871–882, 2011.
20. Khalifelu, ZA. Gharehchopogh, F.S. Comparison and Evaluation Data Mining Techniques with Algorithmic Models in Software Cost Estimation, Elsevier Press, Procedia-Technology Journal, ISSN: 2212-0173, Vol. 1, pp. 65-71, 2012.
21. Nikolic, M. Teodorovic, D. Empirical study of the Bee Colony Optimization(BCO) algorithm, Expert Systems with Applications, Vol. 40, pp. 4609-4620, 2013.
22. Gharehchopogh, F.S. Dizaji, Z.A. A New Chaos Agent Based Approach in Prediction of the Road Accidents with Hybrid of PSO Optimization and Chaos Optimization Algorithms: A Case Study, International Journal of Academic Research Part A, 6(2): 108-115, 2014.

(DOI: dx.doi.org/14.9831/1444-8939.2014/2-6/MAGNT.150)