International Journal of Academic Research in Computer Engineering (2476-7638)
Vol: 1, No: 1, Pages: 53-60, September 2016

International Journal of Academic Research in Computer Engineering
Volume 1, Number 1, Pages: 53-60, September 2016
www.ijarce.org
IJARCE

A New Model for Software Cost Estimation Using Bat Algorithm

Muhammad Ibrahim[1*],

[1]Department of Computer Science, COMSATS Institute of Information Technology, Wah Campus, Pakistan.

Receive Date: 19 June 2016; Accepted Date: 27 August 2016, Published Date: 15 September 2016

*Corresponding Author: ibrahim@ciitwah.edu.pk

Abstract
Several software projects are developed and produced in software companies annually. Rapid environmental and technological changes, cost constraints, mismanagement, lack of skills of the project managers and inaccurate estimation cause many of these projects to fail in practice. In large software projects, software cost estimation has always been one of the main problems for the project manager and project-development team. The main SCE criterion is to determine the effort required to complete the project. The effort needed is usually determined according to people / month. There is a direct relationship between the amount of effort and the time required to complete a project. And the time can be used to estimate costs. Therefore, the main step in SCE is determining the amount of effort required to complete projects. COCOMO model is the algorithmic model for SCE. In COCOMO model, the emphasis is on the effort coefficients for better performance. In order to measure the relationship between the effort and time, i.e., the relation between the number of code lines and effort coefficients, COCOMO uses some linear formula that utilizes data from NASA software projects. In this paper, we have used bat algorithm to accurately estimate the effort coefficients and to reduce MMRE. Evaluation has been conducted on NASA60 dataset; the results show that BA has much less error compared with COCOMO model.

Keywords: software cost estimation, COCOMO model, bat algorithm, NASA60 dataset

1-Introduction

SCE is one of the key fields in the management of software projects [1]. Estimating the cost of software projects, especially in status quo where projects are facing crisis at any time are more obvious. The project environment is affected by the sophisticated management requirements and these conditions are worse for large projects. On the other hand, in the project life cycle, the time and expense are among important and necessary conditions to start, stop and exploit the projects. Managing and controlling software projects cost include planning, cost estimating, budgeting and controlling the cost of software projects in the project life cycle [2]. In the case that the project can be completed with the approved budget. In planning the cost management, procedures, project management methodology and controlling project costs are specified. In costs estimation, the financial resources needed to implement the project are predicted. In budgeting, gathering estimated costs is specified to determine the cost base. Also, in controlling the costs, control and management of the changes in project costs are analyzed and the required predictions for the future costs in various practical scenarios are determined and the cost of all projects are identified [3].

Software world is rapidly changing; its market expands exponentially and any failure in software projects eliminates software development and production companies from the competition scene[4]. In the emergence of software, users of the new industry were its suppliers and producers. At that time, the software was mainly used for calculations and solving math problems. Low level languages and hardware limitations (lack of memory and low processing speed) are some other features of the software emergence [5]. In those early days, software was not separate from hardware and was embedded in it for free. But with the increasing application of computer and the growth of software in a variety of field gradually brought about some conditions in which the users of software were separated from its designers and manufacturers. The companies were created which their activities were exclusively the production and development of software projects. The emergence of high-level languages and overcoming hardware limitations are some other features of the new software age. Thus, the needs of software users and its capabilities have made software designing and production very complex. Other factors such as intense competition, absence of professionals and experts, lack of access to knowledge and successful experience of others, need to produce quickly and, most importantly, the lack of proper use of the principles of engineering in designing software production have made the industry face many challenges [6]. There are some methods and techniques for SCE which should be used whenever required.

IF possible, the project manager can use two or more methods and techniques for the estimation. If the estimates obtained from different methods are very close, this represents an accurate estimate, but if the numbers obtained using different estimation methods are different it shows that the predictions are wrong and should be re-evaluated [7-8]. In this paper, BA [9], which is one of the meta-heuristic algorithms based on collective intelligence, is used for SCE and its results were compared with the COCOMO model [10-11].

We have constructed the overall structure of the paper as follows: we explain the related works in the second section. We describe bat algorithm at the third section. At the fourth section, we describe the proposed model. In the fifth part, we evaluate the results of the proposed model, and finally in the section six, we explain the conclusion and future works.

2. Related Works

Project success depends on a balance among three major factors: time, resources and the results consistent with offering a suitable level of services to customers. So three important factors already mentioned should be considered and a good balance should be created among them. The ratio between these three factors is not linear; sometimes the reduction of the half of performance time can increase the cost of using the resources. Before the project-planning phase, project managers must make decisions about the ratio of the three factors mentioned and a good balance must be made between them. So far, much research has been done on estimating the cost of software projects which we will refer to them in this section.

Ant colony optimization model and a hybrid of ant colony optimization and chaos models were implemented on NASA60 dataset [12]. Chaos model is used to optimize the parameters of the ant colony. The results show that the amount of MARE in COCOMO model is 0.29 and in ACO, COA and hybrid models are 0.22, 0.14 and 0.078, respectively. Comparisons show that the hybrid model has less error compared with ACO and COA models. The hybrid model of artificial bee colony and genetic algorithm is proposed to estimate software costs [13]. The amount of MMRE error is 7.69 in the hybrid model and 29.64 in COCOMO model. The hybrid model reduces MMRE error to 3.38 in comparison with COCOMO model. The hybrid model increases the accuracy of PRED (15) criterion to 5.46 compared to COCOMO model.

Moreover, in comparison with Genetic algorithm and artificial ant colony, the hybrid model increases the percentage of standard error PRED (15) to 1.65 and 1.12 times. Genetic algorithm and artificial ant colony increases the percentage of standard error PRED (15) to 3.30 and 4.84 times in comparison with COCOMO model. Based on the evaluations done, it can be said that MMRE and PRED (15) criteria have better efficiency in the hybrid model; also, artificial ant colony is more efficient in comparison with COCOMO and genetic models.

PSO-Chaos hybrid model is proposed for the effort and SCE cost [14]. PSO algorithm has mixed with chaos model and was able to reduce the amount of MARE error compared with COCOMO model. Assessment was conducted on NASA60 dataset. The results show that PSO model compared to COCOMO model has reduced MARE error rate to 0.1506 which its MARE error rate is equal to 0.2952. Furthermore, the hybrid model of PSO with chaos shows that MARE amount is reduced to 0.1153. It can be concluded that hybrid model is more accurate than the COCOMO model.

Using PSO algorithm, a new effort estimation model is proposed for software projects [15].

Parameters affecting effort estimation has been studied using PSO algorithm. Evaluation has been conducted on KEMERER dataset with 15 projects. Experimental results show that the amount of MMRE error is 56.57 in the proposed model and 245.39 in COCOMO model.
With the combination of PSO algorithm, PSO-FCM and PSO-LA hybrid models have been suggested for SCE [16]. Evaluation was conducted on NASA 60 dataset. In the PSO-FCM hybrid model, minimum inter-cluster distances and the total of intra-cluster distances plus the number of clusters have been used as fitness parameters and PSO algorithm's performance. The FCM (Fuzzy C-Means) use makes the particles gather in the best cluster and fitness function have local optimum points. In order to improve the performance of PSO algorithm LA (Learning Automata) is used to adjust the particle behavior. In the hybrid PSO-LA model, all particles simultaneously look for a place in the searching space. LA model strategy in the PSO-LA hybrid model paves the way for the particles to reach several optimum places according to reward criterion. Their test results show that PSO-FCM model has less MRE error rate than the PSO-LA hybrid model. MMRE error in the PSO-FCM model is 25.36, 24.56, and 24.22 and in PSO-LA model is 23.86. The accuracy of PRED (25) in the COCOMO model is equals to 40 and in PSO_FCM model is 61.6, 58.3, 65 and 68.3. Also in the PSO-LA, it is equal to 63.3.
The hybrid of GA and Ant Colony Optimization has been suggested for SCE [17]. In the hybrid model, affective factors in estimation have been tested using GA and trained using ACO and the better results have been obtained compared with COCOMO model. Review and evaluation has been done on 60 software projects of NASA dataset. Their results show that the hybrid of GA and ACO is more efficient in estimating the cost of software projects compared with COCOMO model and has less MRE (Mean Magnitude of Relative Error) than COCOMO model.
The hybrid of PSO algorithm and FLANN network has been used for SCE [18]. PSO-FLANN is a kind of Feed Forward Neural Network which has three layers. PSO algorithm is used to train the weight of FLANN vectors. Evaluation took place on three COCOMO 81, NASA63 and Maxwell datasets. Experimental results show that the PSO-FLANN model is more efficient in testing and training data in MMRE and PRED criteria (25).
The hybrid model Function Link Artificial Neural Network (FLANN) and GA is used for SCE [19]. PSO-FLANN is a kind of Feed Forward Neural Network which has three layers. GA algorithm is used for the two OFWFLANN and OCFWFLANN hybrid models. In OFWFLANN model, GA algorithm is used to optimize the parameters influencing the cost and in OCFWFLANN model to train the weight of training FLANN vectors. Evaluation has been done on NASA93 Dataset. Experimental results show that hybrid models have good performance in testing and training data.
Recently, the application of data-mining techniques has become very popular in SCE [20]. Software cost estimation has been estimated using Liner Regression (LR), ANNs, Support Vector Regression (SVR) and K-Nearest-Neighbors (KNN) techniques. LR model can be used to determine the dependency traits in SCE. LR model finds the relationship between independent and dependent factors among the data. ANNs try to do SCE more accurately by training and testing data. SVR model was used to optimize the factors affective in SCE. KNN is a technique in data-mining which is utilized for data classification in the collection of data that has been previously classified and identified. KNN is used to determine the weight of features affective in SCE. Their test results show that the SVR model has less MRE than other models.
ANN-MLP is one of the common methods in SCE [21]. In order to show the efficacy of ANNs, from among the 60 projects in software NASA dataset, 11 projects have been tested and trained using ANNs and compared with COCOMO model and have shown that COCOMO model has more error than ANNs model. The results show that in more than 90% of the cases, ANNs model has much better estimation than COCOMO model. Thus, it can be concluded that AI- based methods are a good substitute for algorithmic methods as a complement.

3. Bat Algorithm

Collective intelligence is one of the strongest optimization techniques which are based on group behavior. Bat algorithm [9] is inspired from the collective behavior of bats in the environment which was introduced by Yung in 2010. This algorithm is on the basis of the use of sound reflection of the bats. Bats find the exact direction and place of their bait by sending sound waves and receiving its reflection. When the sound waves are reflected towards the wave transmitter, the bird can depict a sound picture of the forthcoming barriers and see the hereabouts well. Using this system, bats can detect movable things as insects and immovable things as the trees. Bats are amazing birds. They are the only mammals which can fly; also many of them have the advanced locating ability with the help of sound. Often, bats utilize sounds with frequency for location within a certain range. Among all their species, small bats are the well-known examples which use sound as a tool for location. Small pulse bats release very high sounds and process them through the echo caused by the confrontation of pulses with the environment.

The Total Rules of Bat Algorithm:

The first rule: all the bats detect the distance using the reflection and they all grasp the difference between the sound reflected from the food or the things in the environment.

The second rule: bats follow their baits accidentally with vi speed and in xi situation with the frequency of f min and different wave lengths ^ and loudness A0; they can also automatically adjust their own wave lengths and the ratio of the pulses sent [r∈[0,1] according the closeness of their bats.

The third rule: loudness maybe different in many states; although it is assumed that these changes are limited from the maximum value R0 to the minimum value as $R_{\min}$.

Based on the described rules, for mathematical modeling, for each virtual bat i in searching d-dimensional search space, x_i^t situation and v_i^t speed in each algorithm repetition, the relations (1), (2), (3) can be used [3]:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \qquad (1)$$

$$v_i^t = v_i^{t-1} + (x_i^t + x^*) \qquad (2)$$

$$x_i^t = x_i^{t-1} + v_i^t \qquad (3)$$

In the relations stated, t parameter counts algorithm repetition and β is a random numerical with uniform distribution in 0-1 range. x^* Parameter is the best current situation which is selected in each repetition after comparing with the new situations of the bats.

Normally, f frequency can be considered with $f_{\min} = 0$ and $f_{\max} = 100$. In each repetition, in local search, one of the answers is chosen as the best one and the bat`s new situation is updated with random step according to equation (4) [3].

$$x_{new} = x_{old} + \varepsilon A^t \qquad (4)$$

In equation (4), $\varepsilon \in [0,1]$ is a random numerical and $A^t = < A_i^t >$ is the average of the bat loudness in t repetition. Also, the loudness of the sound A_i and the ration of the pulse sent r is updated in the form of equation (5) in each repetition [3].

$$A_i^{t+1} = \alpha A_i^t, \qquad r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \qquad (5)$$

In equation [5], the values of α and γ are fixed and are in [0.9-0.98] and for each $0 < \alpha < 1$ and $\gamma > 0$ we have the equation (6):

$$A_i^{t+1} \to 0, \quad r_i^t \to r_i^0, as \quad t \to \infty \qquad (6)$$

Bat algorithm is one of meta-innovative algorithms inspired from nature which is indicative of how the bird hunts. This algorithm has never been used in classification, but because of its high efficiency in solving difficult problems we have decided to use it in the case that feature selection will be done with its help.

4. The Proposed Model

If a new software project is similar to an old one in size and function, we can benefit from the experience of the previous project; we can even say that the new project will cost the same time and money as the previous one. But, correct estimation of the indices is the solution which can be useful in most of the cases. For example, when a software project is designed if the number of the required people (people in each month) estimated or the cost or time calculated accurately, the software designing and production stages can be correctly scheduled. Financial resources management, supervision and control of costs and determining the overall cost of the project is of great importance for the companies. This issue is very crucial in software production and development companies. Thus, cost estimation, determining and controlling the overall cost and making decisions to reduce the overall cost of the projects have been set as a goal. Among the issues which many software projects are now facing and cause delay in the projects and slow down the production operation can be mentioned to inaccurate recognition of the amounts and values of the projects, wrong estimation of the resources needed, inability in cost management and budgeting of the projects costs, inability in controlling the cost function of the projects, and generally inability in cost management system. In Figure (1), the Flowchart of the proposed model is shown for the estimation of NASA60 dataset factors [22].

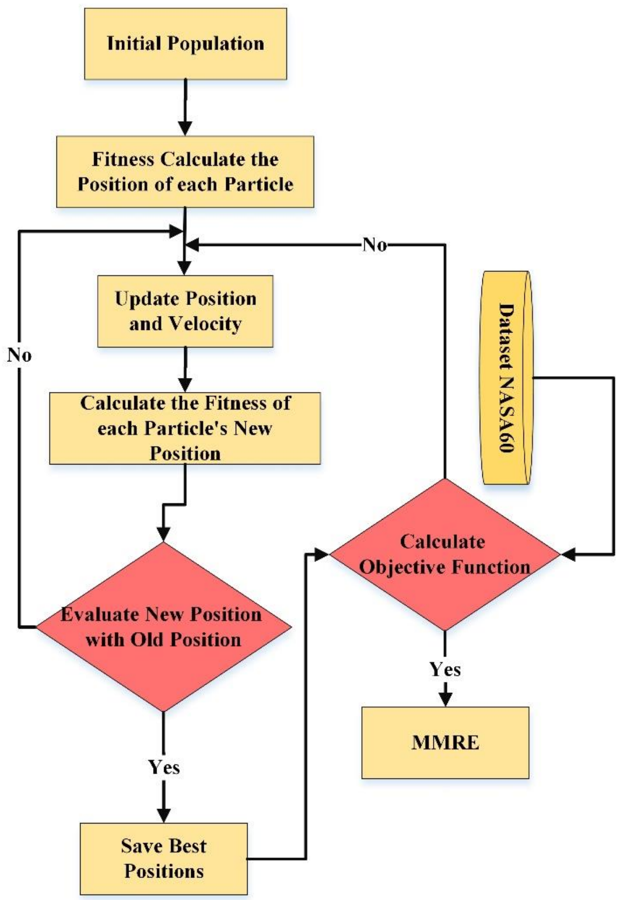

Figure 1. Flowchart of the Proposed Model

Meta-heuristic algorithms are a kind of random searching on the basis of population; their first and strongest point is that they are inherently parallel. They can consider the searching space in various fields. The parallel searching of sub-spaces makes the existence of an absolute optimum point more probable. In non-algorithmic methods, unlike algorithmic methods, answering space is thoroughly searched and there is less probability for involvement in a local optimum point. Meta-heuristic algorithms are not dependent on function continuity or derivation; they only need to determine the amount of goal function in the searching process.

In the proposed method, firstly, according to the limit of problem space, the primary population is created in the [0.4-1.4] limit for minimizing the effort factors. Thus, on the basis of each repetition, first the amount of effort factors will minimize according to the size of code lines. After obtaining the optimum amount for the effort factors, their effect in the modeling step will be assessed based on MMRE criterion. In the proposed model MMRE is considered as the fitness function. The goal of the fitness function is to minimize MMRE in comparison with COCOMO model. Fitness function for the proposed model is defined as equation (8) [22]. In equation (8), act parameter equals the actual amount and est parameter equals the estimated amount that has been achieved.

$$MRE_i = \frac{|act_i - est_i|}{act_i} \times 100 \qquad (7)$$

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \,, i = 1,2,\ldots,n \qquad (8)$$

By using equation (8), we can obtain the total amount of error coming from the effort factors. In the proposed model, effort factors are placed in COCOMO model after optimization and MRE and MMRE values are achieved.

5. Evaluation and Results

In this section, BA was tested on the software projects of NASA60 dataset and its results were compared with COCOMO model. Simulation was carried out in VC#.NET 2013 programming environment on NASA60 dataset that has 15 EM factor. In meta-heuristic algorithms, the adjustment of the values of parameters is very important to reach an optimum answer and can have a significant impact on the function and efficiency of optimum solutions. Some parameters in BA, which include the number of initial population and the number of repetitions, are placed as 50 and 100, respectively. The results obtained from the application of BA in testing and training steps are shown in Table [1]. The experimental results on the sets of data on NASA60 dataset indicate that BA has less MRE error.

The comparison of BA and COCOMO in Figure (1) shows that in comparison with COCOMO BA has less MRE; thus, the impact of particles has caused particles to have much accuracy in data-training and testing steps. In Figure (2), the comparison of MRE in models is shown.

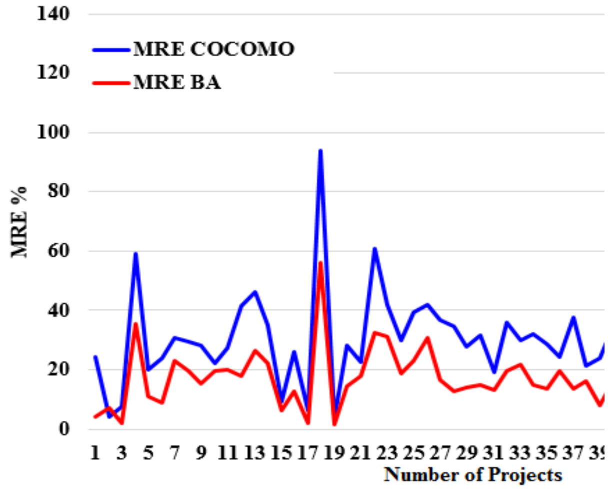

Figure 2. A Comparison BA with COCOMO based on MRE

In Table 1 MMRE evaluation results are shown. MMRE value in BA is equal to 16.98. This indicates that BA has been able to reduce the amount MMRE to about 1.74 in comparison with COCOMO model. Also, BA model was able to reduce the amount of MMRE in comparison with models of GA-BNN [23], GA-ACO [17], ABC [24], ANN-ACO [25], and HS [26].

Table 1. A Comparison of BA with Other Models based on MMRE

Models	MMRE
COCOMO	29.64
GA-BNN [23]	22.25
GA-ACO [17]	27.78
ABC [24]	28.36
BA	16.98

In Figure (3), MMRE graph of the models is shown. As you can see, BA in comparison with COCOMO and other models has less MMRE.

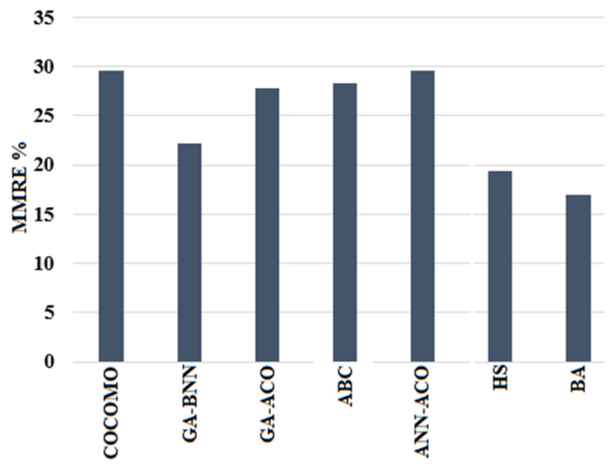

Figure 3. Comparison Chart of BA with Other Models based on MMRE

6. Conclusion and Future Work

Estimating the cost of software projects is one of the most important and most complex aspects in project management. One of the major concerns of project and design managers is planning, budgeting, and controlling cost. One of the common issues in most of the projects is increased cost. Despite the wide research done in this field, the main reasons of this problem have not been completely found and an effort has not been made to solve them.

In software projects, costs are directly or indirectly related to the project environment, and these factors have relative effect on the total function of the costs. Although, direct costs are often fixed costs, they may include part of variant costs. In this article, using BA we evaluated the software projects of NASA60 dataset. The results indicated that MMRE value in the proposed model is 16.98 which have reduced the amount of MMRE in comparison with COCOMO model to 1.74 times. In the future, we hope to minimize more and more the MMRE value using the hybrid of meta-heuristic and data-mining algorithms in comparison with COCOMO and other collective mind algorithms.

References

[1] J. Hihn, L. Juster, J. Johnson, T. Menzies, and G. Michael, Improving and Expanding NASA Software Cost Estimation Methods, 2016 IEEE Aerospace Conference, Big Sky, MT, pp. 1-12, 2016.

[2] Z.H. Wani and S.M.K. Quadri, Artificial Bee Colony-Trained Functional Link Artificial Neural Network Model for Software Cost Estimation, Advances in Intelligent Systems and Computing, Vol. 43, pp. 729-741, 2016.

[3] E.E. Miandoab and F.S. Gharehchopogh, A Novel Hybrid Algorithm for Software Cost Estimation Based on Cuckoo Optimization and K-Nearest Neighbors Algorithms, Engineering, Technology & Applied Science Research, Vol. 6, No. 3, pp. 1018-1022, 2016.

[4] F.S. Gharehchopogh, I. Maleki, A. Kamalinia, and H.M. Zadeh, Artificial Bee Colony Based Constructive Cost Model for Software Cost Estimation, Journal of Scientific Research and

Development, Vol.1, No.2, pp.44-51, 2014.

[5] I. Maleki, L. Ebrahimi, S. Jodati, and I. Ramesh, Analysis of Software Cost Estimation Using Fuzzy Logic, International Journal in Foundations of Computer Science and Technology (IJFCST), Vol. 4, No.3, pp. 27-41, 2014.

[6] F.S.Gharehchopogh, A. Talebi, I. Maleki, Analysis of Use Case Points Models for Software Cost Estimation, International Journal of Academic Research, Part A, Vol.6 ,No.3, pp.118-124, 2014.

[7] I. Maleki, L. Ebrahimi, F.S. Gharehchopogh, A Hybrid Approach of Firefly and Genetic Algorithms in Software Cost Estimation, MAGNT Research Report, Vol.2, No.6, pp.372-388, 2014.

[8] F.S. Gharehchopogh, I. Maleki, N. Ghoyunchizad, E. Mostafaee, A Novel Hybrid Artificial Immune System with Genetic Algorithm for Software Cost Estimation, MAGNT Research Report, Vol.2 , No.6, pp. 506-517, 2014.

[9] X.S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Studies in Computational Intelligence, Vol. 284, pp. 65-74, 2010.

[10] B.W. Boehm, Software Engineering Economics, Prentice-Hall, Englewood Cliffs, New Jersy, 1981.

[11] B.W. Boehm, Software Cost Estimation with COCOMO II, Prentice Hall PTR, Englewood Cliffs, New Jersy, 2000.

[12] Z.A. Dizaji, F.S. Gharehchopogh, A Hybrid of Ant Colony Optimization and Chaos Optimization Algorithms Approach for Software Cost Estimation, Indian Journal of Science and Technology, Vol.8, No.2, pp.128–133, 2015.

[13] F.S. Gharehchopogh, I. Maleki, A. Talebi, Using Hybrid Model of Artificial Bee Colony and Genetic Algorithms in Software Cost Estimation, 9th International Conference on Application of Information and Communication Technologies (AICT), IEEE, Rostov on Don, pp. 102-106, 2015.

[14] Z.A. Dizaji, K. Khalilpour, Particle Swarm Optimization and Chaos Theory Based Approach for Software Cost Estimation, International Journal of Academic Research, Part A, Vol.6, No.3, pp.130-135, 2014.

[15] F.S. Gharehchopogh, I. Maleki, S.R. Khaze, A Novel Particle Swarm Optimization Approach for Software Effort Estimation, International Journal of Academic Re-search, Part A, Vol.6, No.2, pp.69-76, 2014.

[16] F.S. Gharehchopogh, L. Ebrahimi, I. Maleki, S.J. Gourabi, A Novel PSO based Approach with Hybrid of Fuzzy C-Means and Learning Automata in Software Cost Estimation, Indian Journal of Science and Technology, Vol.7, No.6, pp. 795-803, 2014.

[17] I. Maleki, A. Ghaffari, M. Masdari, A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization, International Journal of Innovation and Applied Studies, Vol.5, No.1, pp.72-81, 2014.

[18] T.R. Benala, K. Chinnababu, R. Mall, and S. Dehuri, A Particle Swarm Optimized Functional Link Artificial Neural Network (PSO-FLANN) in Software Cost Estimation, Advances in Intelligent Systems and Computing Proceedings of the International Conference on Frontiers of Intelligent Computing, pp. 59-66, 2013.

[19] T.R. Benala, S. Dehuri, Genetic Algorithm for Optimizing Functional Link Artificial Neural Network based Software Cost Estimation, Proceedings of the InConINDIA, pp. 75-82, Springer-Verlag, 2012.

[20] Z.A. Khalifelu and F.S. Gharehchopogh, Comparison and Evaluation Data Mining Techniques with Algorithmic Models in Software Cost Estimation, Elsevier,

Procedia-Technology Journal, Vol. 1, pp. 65-71, 2012.

[21] 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.

[22] S.G. MacDonell, A.R.Gray, A Comparison of Modeling Techniques for Software Development Effort Prediction, in Proceedings of International Conference on Neural Information Processing and Intelligent Information Systems, pp. 869-872, 1997.

[23] S. Sadouni, A. Jafarian, Hybrid Method for Software Cost Estimation Using Genetic Algorithm and Bayesian Belief Network, MAGNT Research Report, Vol.3, No.1, pp. 1200-1207, 2015.

[24] S. Ebrahimi, A. Jafarian, A New Model for Software Cost Estimation by Using Artificial Bee Colony Algorithm, International Journal of Academic Research, Part A, Vol.6, No.5, pp.178-184, 2014.