Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 7, No. 4, November 2016), Pages: 1-12
www.jacr.iausari.ac.ir

New Approach with Hybrid of Artificial Neural Network and Ant Colony Optimization in Software Cost Estimation

Nader Ebrahimpour✉**, Farhad Soleimanian Gharehchopogh, Zeinab Abbasi Khalifehlou**
Department of Computer Engineering, Mahabad Branch, Islamic Azad University, Mahabad ,Iran
nebrahimpour@ktu.edu.tr; bonab.farhad@gmail.com; tabriz.ce@gmail.com

Received: 2016/01/27; Accepted: 2016/05/29

Abstract

Nowadays, software cost estimation (SCE) with machine learning techniques are more performance than other traditional techniques which were based on algorithmic techniques. In this paper, we present a new hybrid model of multi-layer perceptron (MLP) artificial neural network (ANN) and ant colony optimization (ACO) algorithm for high accuracy in SCE called Multilayer Perceptron Ant Colony Optimization (MLPACO). Current research uses some of features for increasing accuracy of estimation among of the existing parameters has been considered for effort estimation in software projects, and then these selected features will be filtered by ACO algorithm in order to reach highest accuracy in estimation and optimization of MLP ANN method. The results show that this novel approach with high accuracy for more than 80% cases is better than algorithmic constructive cost model (COCOMO) in the majority cases. Also, the results of proposed algorithm show that mean magnitude of relative error (MMRE) in the proposed algorithm is lower than COCOMO model.

Keywords: *Software Cost Estimation, Artificial Neural Network, Ant Colony Optimization, Effort, Optimization*

1. Introduction

One of the issues has been allocated a lot of research over the last decades, is estimating the necessary costs for performing software projects. The algorithmic methods were the first models that were developed in the past decades. They estimate the cost of software projects with COCOMO family. COCOMO 81 is the first and the most stable models, and still is the one of the best methods in some software projects [1]. Estimation software models help project managers to estimate delivery time, cost, manpower that were required to software development [2].

By wide development of technology in current years, algorithmic models couldn't response to the construction of a precise estimation for software projects. So, needs for non-algorithmic models which are based on artificial intelligence and machine learning techniques were sensed. Machine learning techniques work based on data which had been earned from previous success projects and it is an advantage in order to increasing accurate estimation in the face of similar projects during software life cycles. Artificial intelligent models such as decision trees (DTs), artificial neural networks (ANNs), fuzzy systems, evolutionary computation and other methods that has been emerged in SCE studies. The ANNs due to their advantages such as, self-learning, associative memory, high parallelism power, and also because of their fastness and fault tolerance against noise, which may be exist in input parameters, and their cheapness in reuse of existing solutions are powerful in cost estimation [3], [4], [5], [6], [7].

ACO algorithm is one of the most prominent algorithms in studies that introduced in 1996 to solve hybrid optimization problems [8].This algorithm uses momentum transfer rules and updating to find closer path or in other words to find optimum answer. In this paper, we propose a new approach for SCE with hybrid of ACO algorithm and MLP ANN; and evaluate the exactness of the presented model to provide an efficient model for realistic SCE projects. The structure of this paper is organized as follows: Section 2 is the review of previous researches in this field. Section 3 is the scrutiny of the SCE, how to estimating algorithmic and data mining techniques in software projects. In section 4, we utilized the proposed algorithm to estimate time and cost of software production. In section 5, we evaluate the results and discuss and, section 6 concludes the paper and future works.

2. Related Works

In recent decades, ANNs have allocated many studies to SCE because of their learning and modeling complex nonlinear relationship ability. Also, Nassif et. al [9] presented a conformist learning method which is based on MLP ANN for SCE and showed that performance of an ANN was based on its architecture and parameters setting. Network training is done with by feed forward back propagation method. Their proposed model's performance is based on the Mean Relative Error (MRE) criteria which is better than the original COCOMO model and is equal to 3.34.

Another new model based on technique of MLP ANN has been provided [6] and the input parameters of the network in this study are conforming to the factors of the intermediate COCOMO. Used database consisted of 60 projects of NASA, and the MRE model showed that over 90% of the results using ANNs algorithm works better than COCOMO model.

In [10] has been used COCOMO 81 data set for training MLP network with feed forward architecture and obtained experiment results based on MRE criteria for ANN with 13 hidden layers equal to 1.50. Also, Khalifehlou and Gharehchopogh [11] compared data mining techniques and algorithmic models to check and assess the SCE. SCE is presented by using four data mining techniques which are Liner Regression (LR), ANNs, Support Vector Regression (SVM) and K-Nearest Neighbors (K-NN). LR model can be used to determine the dependence of the effective characteristics of software cost estimation. LR model will find parameters between independent and non-independent factors in data. ANNs is trying to be more precise in SCE by training and testing data. SVR model is used to optimize effective factors in SCE. K-NN is a technique of data mining for data classification among classified data set which their properties was specified. By use of K-NN weight of effective characteristics of SCE can be determined. Their experiment results show that SVR model in comparison with other models has less MRE.

Another work based on the MLP architecture done by Attarzadeh et.al [5], to estimate the amount of effort. This model has been made to collate architecture post COCOMO II model with MLP ANN. The architecture is come from the propagation training algorithm with sigmoid function in the hidden layer neurons. Evaluation criteria for the performance of the proposed network MMRE and percentage relative error deviation

(PRED) (25%) is equal to 0.4579 and 45.5%, while these two evaluation criteria for the algorithmic data set of COCOMOII model respectively is 0.5025 and 37.5%.

Also, Khalifehlou and Gharehchopogh, offer a new model based on regression for development of software cost and effort estimation [12]. The results of their experiment show that the regression model can reduce amount of error in MRE, and it has a good effect on the calibration of COCOMO model parameters. In [13], ACO algorithm and Tenet mapping algorithm were used as chaos optimization algorithm and NASA data set were used to training and testing and validation of proposed algorithm. In their research, absolute relative error was used to compare and evaluate experiment results of proposed method vs. COCOMO model. Thus, the results showed that there was a decrease in average relative error up to 0.1097 percent. Also, Marandi and Khan [2] presented a new method to improving quality of software applications by using linear regression to estimate software costs. Their proposed method aims to decrease software production costs and improve techniques as efficiently.

Khalifehlou and Gharehchopogh, [3] studied data mining techniques in SCE. They scrutinized ANNs, neuro fuzzy (NF), Fuzzy Logic (FL), fuzzy decision tree (FDTs), Bayesian network (BN), multilayer regression (MLR) and LR techniques. In MLR and BN models estimation is done by using analysis of interdependent variables. In FL, FDTs, and NF models a range of fuzzy membership is contemplated for each factors of estimation. SCE by particle swarm optimization [14], has been done on the basis of particles swarm with combination of Fuzzy C-means and Learning Automata method [15] and it has been done with a mix of chaos and bee colony optimization in [16]. Huang et. al [17] analysis data preprocessing techniques which influence final SCE with machine learning based methods. They showed that a careful selection of datasets and machine learning techniques play important role to reduce cost estimation.

SCE by using machine learning techniques has been made a great progress, and to be mention that machine learning techniques are more precise than algorithmic models. Artificial Intelligence techniques can optimize effective factors of estimation with constant repetitions and training data and also can minimize cost and effort of software projects.

3. Software Cost Estimation

SCE depends on the potential cost estimation, planning and needed manpower for software development [18]. In the past decades, a lot of cost estimation models have been introduced. Most studies are about identifying and understanding parameters that affect costs of software. Generally, SCE techniques can be divided into three categories [19]: expert judgment, algorithmic models and intelligent computational models. In expert judgment techniques, cost and effort of projects are inferred based on expert's personal experiences of similar projects, hence it cannot develop a clear estimation because of this can't be generalized as a model for other projects [11]. After those algorithmic models have been developed, which was estimation based on a statistical liner regression equation and a set of nonlinear regression equations such as COCOMO [18]. But with today's complexity of software, some major estimation factors have been emerged, which was ignored in algorithmic models. To escape from all restrictions of algorithmic models non-algorithmic models are developed based on artificial intelligence that was proved these techniques have much better results than (in the condition which local data are used) algorithmic techniques [20].

3.1 Algorithmic Models

COCOMO model is the most well-known model of cost estimation based on algorithmic techniques. COCOMO I [1] model and COCOMO II [21] developed by Barry Bohem in 1981 as one of the most supportive and noted as well as widely acceptable model which is used among the models of cost estimation. Constructive cost estimation, as one of the algorithmic models like COCOMO family (according to the size and numbers of written lines for LOC project), is considered as the indicator of function point. It should be noted that the indicator of function point is founded on quantifying the size of the software in conditions of the operation and effort multipliers. Effort multipliers or factors of software cost strongly affect the effort and required cost of life cycle. Three models estimate effort multipliers (EM) on the basis of cost analysis and features of computed projects. Also, they find the closest formula or suitable function for predicting software costs [11]. It is essential to state that this model applies to three classes of software projects named organic, semi-detached and embedded. Effort estimation is as formula (1):

$$Effort = \partial \left(LOC \right)^{b} \tag{1}$$

Where a, b coefficients are in [1].

Table 1. Coefficients of Development Modes of Basic COCOMO Model

Development Mode	a	b
Organic	2.4	1.05
Semidetached	3	1.12
Embedded	3.6	1.20

Basic COCOMO is suitable for fast and early estimation of software projects. But, it is limited because of the loss of factors which show the difference in various characteristics. For creating this model, Bohem et.al [22] were developed COCOMO improved copy named COCOMO II in order to handle new arising problems through new software development methods such as object-oriented programming and new applied software. This model has three sub-models as follows: software compound model, early design model and post architecture model. Some other factors such as the type of software, complexity, participants of projects and so on are participated in COCOMOII. The formulas in COCOMOII model are estimated on the basis of units named person-months. The general calculation of this model is shown in formula (2) [22], [23]:

$$\rho\mu = A \times size^{E} \times \pi_{i=1}^{n} E\,\mu_{i} \tag{2}$$

In formula (2), approximate of A and B is productivity constant which its unit is stated as person-months. Both of them can be calibrated. Also, S is the size of software product and E is the sign of being economic or noneconomic of the project which are taken from scaling factors. If E<1, so the results are economic and the consequences are noneconomic if E>1. And, EM is the effort multipliers and n is the numbers of effort multipliers.

4. Proposed Model

In algorithmic models, constant amounts of cost estimation are not distinguished but they are considered in an average way. Therefore, estimated amounts cannot be easily trusted. In addition, Authors research over the years on how to use meta-heuristic methods to estimate software cost have been done. Hence, we were trying to optimize a new method based on ant colony optimization method in development of software cost estimation methods. ANN approach greatly is as well-known method in the scientific literature for intelligent estimating of software costs have been applied over the past years [4], [5]. While, we utilize ACO method to improve performance of ANN and present optimization in ANN method to show hybrid of meta-heuristic approaches lead to better results in SCE. ACO method with its own capabilities made this possible. According that, this is our effort to overcome challenges in SCE scopes with researches in meta-heuristic applications.

In this paper, the combination of MLP ANN and ACO are used for estimating the cost of software projects. Exact and accurate estimation can be achieved for the development of software projects by using MLP ANN and ACO algorithm through the combination of different factors such as hardware, software and human beings. Effective factors in cost estimation are tested in proposed algorithm. MLP ANN uses back propagation algorithm in training of network and sigmoid function in hidden layer of neurons . There are 17 nodes in input layer, because input parameters to estimate software costs based on COCOMO model are 17 parameters. So, we uses 17 nodes in input layer of network in which results of presented model has been computed based on real data and variables in cost estimation systems.

In addition, operations of instruction are done by using ACO algorithm in order to optimize parameters. In estimation accuracy, the amount of EM has great and considerable importance. Some of EM parameters are: software and hardware requirements, number of required personal, product and project features and etc.

Trying to predicate required effort is a linear algebra problem which can be solved by optimization methods. The results of output based of diagram of training and testing based on COCOMO and presented hybrid model show that our approach with hybrid of ACO and MLP ANN outperforms the classic COCOMO model. By using MLP ANN in proposed algorithm, suitable amount of factors with regard to the size of projects are tested. Afterwards, ACO algorithm is used to optimize instruction well. The flowchart of proposed algorithm is shown in Figure 1. One weight is considered for each factor in which the project cost is able to be estimated with COCOMO by use of them.

On estimating effort related to new projects of system by the utilization of new projects factors and connected weights to all COCOMO factors which are available, the amount of new software projects effort are assessed. In this stage, the earned weights are entered into ACO algorithm and parameters are optimized through this algorithm, too. The act of MLP ANN instruction and optimization factors with ACO algorithm will be continued until MMRE model is dwindled than COCOMO model. For finishing algorithm, fitness function as considered named MMRE must actually be smaller than the amount function in COCOMO model. Quasi code of proposed algorithm of MLP ANN and ACO algorithm is shown in Table 2. The stages of proposed algorithm are as follows:

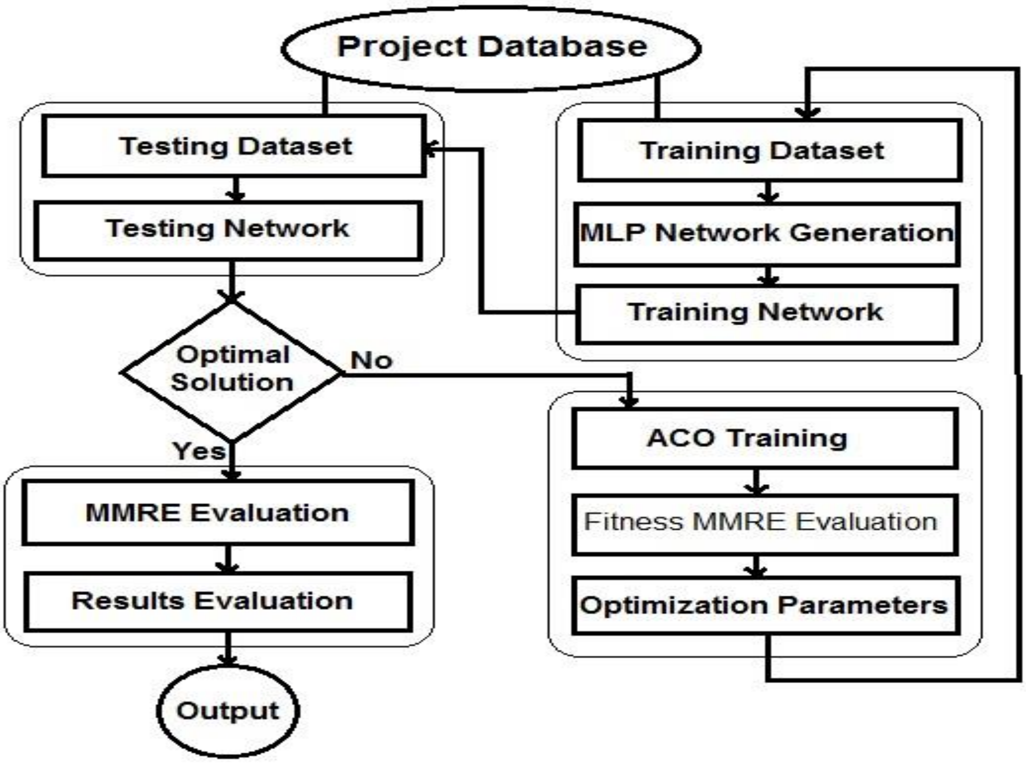

Figure 1. Flowchart of Proposed Algorithm

Table 2. Quasi Code of Proposed Algorithm

Proposed Algorithm
1. Initialize Parameters 2. Do MLP ANN Generation Training ANN 3. Testing ANN 4. If Optimal Solution =Yes go to 8 5. Training ACO 6. Fitness MMRE Evaluation MRE=|Estimated – Actual| / Actual; MMRE=∑ (MRE) / N; 7. While (Not Terminate Condition) 8. MMRE Evaluation 9. Results Evaluation 10. Output

- MLP ANN: limited or random information are created with primary MLP ANN on the basis of the function of MLP according to specified domain. 17 neurons in input layer and three neurons in hidden layer and one neuron in outer layer as estimated good response for project are stated in proposed algorithm including cost factors (17 parameters in input) as imported neurons.
- ACO Algorithm: The numbers of ant population are considered 50 and the amount of Rho variable in interval (1-0) is considered 0.9 in program in other

words. The amounts of Beta and Alpha variables in interval (2- 0.1) and (5-1) equaled, respectively, 2 and 5. The choice of one appropriate operator has influence on the efficiency of algorithm. These parameters are based on experiment results and effectively. A new operator is created for integration in proposed algorithm by the application of ACO algorithm. The neurons are combined with each other in new operator according to pheromone current rule [24]. Integration of neurons is defined on the basis of pheromone current rule. MMRE is taken into consideration as the fitness function in proposed model.

5. Results Evaluation

In training and testing phases, we use validated NASA projects [25] to evaluate our approach. This dataset contains real information of costs of previous projects within their parameters such as required personals and platforms in software cost estimation. In train phase, we extract knowledge of them and using them in test phase to predicate costs of new software by meta-heuristic algorithms.

Propriety aim function is compared with COCOMO algorithmic model in proposed model of minimizing MMRE amount. In order to select the best neuron and its amount, algorithm is repeated until a desirable decrease of MMRE amount. MMRE is defined in formula (3). Error collection can be compared among various models by the employment of formula (4).

$$MRE_i = \frac{|Actual_i - Estimate_i|}{Actual_i} \times 100 \tag{3}$$

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \tag{4}$$

For results evaluation, there is a need for an indicator so as to distinguish them. This unique indicator can be considered based on a mathematic model or qualitative criterion in which algorithm is able to differentiate the best answer following them. Whatever the amount of MMRE is lower for one of response neurons, considered response is more propriety. Meta-heuristic algorithms are highly sensitive to their own parameters and the regulation of parameters is able to have considerable influence on their operation.

Consequently, regulation of parameters causes flexibility and more efficiency of hybrid models. It is noticeable that population choice has great importance in Meta-heuristic algorithms. If the number of population is fewer, the problem will encounter early convergence. Therefore, considered and close response to optimization would be difficult. On the other hand, there will be much time for algorithm to become convergence if the number of population is more. In consequence, the number of population must be appropriate and proportional with considered problems for finding an optimal solution. Those parameters which have great influence on algorithm operation have illustrated in Table 3.

Table 3. Parameters and Values in MLPACO Model

Parameters	Values
No. Ants	50
Rho Value	0.9
Input Layer	17
Hidden Layer	3
Alpha Value	2
Beta Value	5
Fitness Function	MMRE

As shown in table 3, input parameters have been specified based on experiment results and observed outputs of each run in optimization methods. With 50 ants in input parameter, the results is better than when number of ants is lower and upper than of it. The time and computational complexity of our approach is high where number of ants more upper than 50 ants and when the number of ants is more lower than 50 ants, then presented hybrid algorithm don't converge to optimal solutions.

We use a type of cross validation in our approach. Dataset has been divided in two parts training and testing sets to evaluate our approach. Division of testing and training datasets is as 20% and 80% of all data. Therefore, we use 80% of data to train and 20% of data to test data to validate and test of our approach. Data partition and output results have been presented in C# 2012 programming language.

MLPACO algorithm with COCOMO model is operated in this software based on 60 projects of NASA for training and testing the results of program with regard to optimizing algorithm. Generally, collection of projects is described as follows:

- Training data= 80 %
- Testing data= 20 %

12 tentative of software projects of NASA dataset are assessed and compared here as testing data. The results of experiments in Table 4 show that compared models have fewer MRE in comparison with COCOMO model. Consequently, proposed algorithm is helpful for estimation and has lesser error estimation in comparison with COCOMO model.

Table 4. MRE Comparison of Proposed Algorithm and COCOMO Model on Test Data

No.	Actual Effort	COCOMO Estimation	MLPACO Estimation	MRE (COCOMO)	MRE (MLPACO)
1	400	217.02	229.6	45.74	42.6
2	450	277.71	320.99	38.28	28.66
3	215	474.41	411.85	120.65	91.55
4	324	484.39	320.19	49.5	1.17
5	360	198.1	208.53	44.97	42.07
6	360	417.07	359.72	15.85	0.07
7	815	853.76	844.87	4.75	3.66
8	1181	1227.43	1096.85	3.93	7.12
9	1248	1202.55	1212.48	3.64	2.84
10	1368	1132.61	1249.2	17.2	8.68
11	2300	1707	1627.6	25.78	29.23
12	3240	4056.77	3313.03	25.2	2.25

As shown in Table 4, proposed algorithm has better estimation in comparison with COCOMO model by application of convergence and repetition approach. The comparison of proposed algorithm with COCOMO model on testing data is shown in Figure 2.

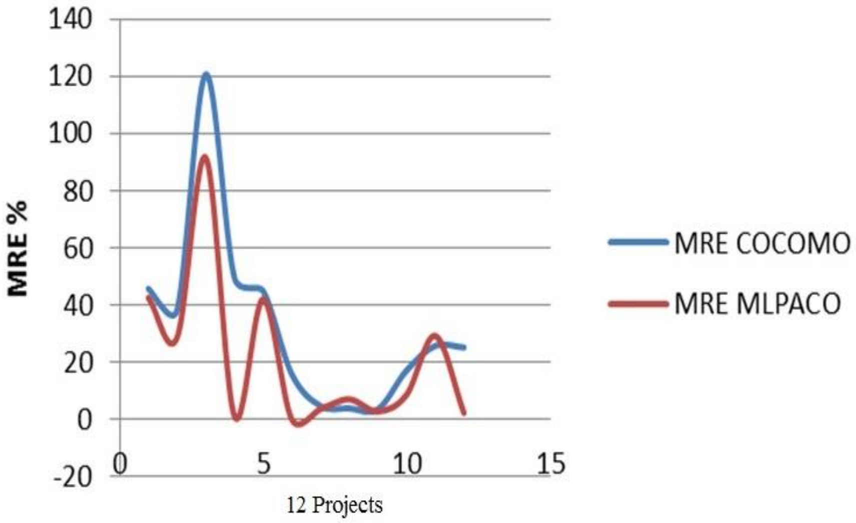

Figure 2. MRE Comparison of Proposed Model with COCOMO Model on Test Data

As it is clear, proposed algorithm has fewer MRE in comparison with COCOMO model. It is noticeable that the obtained consequences of estimation gave positive results in more than 80% of the cases on test data.

The comparison of proposed algorithm with COCOMO model on training software projects are shown in the Figure 3 based on MRE criteria. In Figure 3, comparison of the amounts of MRE on train data is explained between COCOMO and MLPACO models.

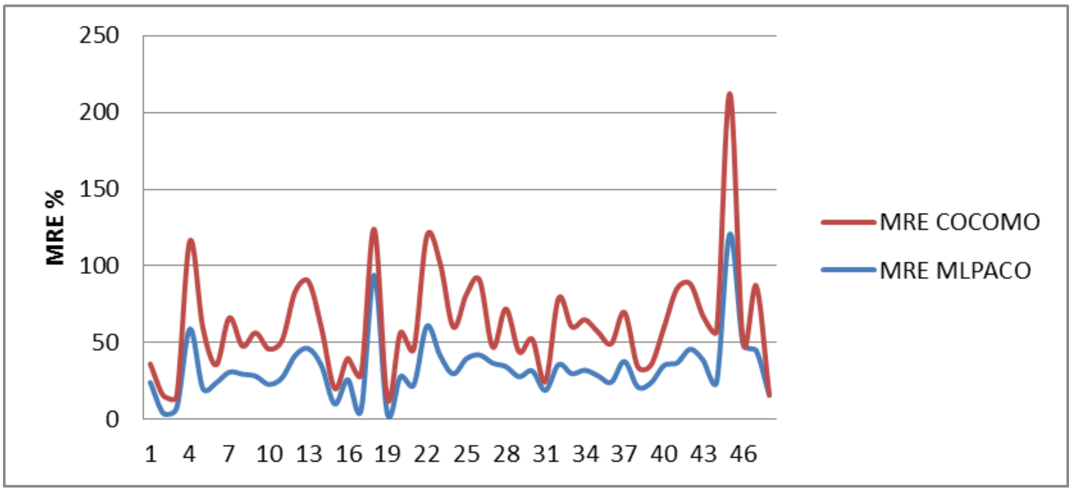

Figure 3. MRE Comparison of Proposed Model with COCOMO Model on Train Data

As shown in Figure 3, estimation results on instructional analyzed data have devoted 80% of total data to themselves. It should be stated that they gave totally better than results to COCOMO algorithm model. The estimation results of COCOMO model and proposed model based on amount of MRE is shown in Figure 4.

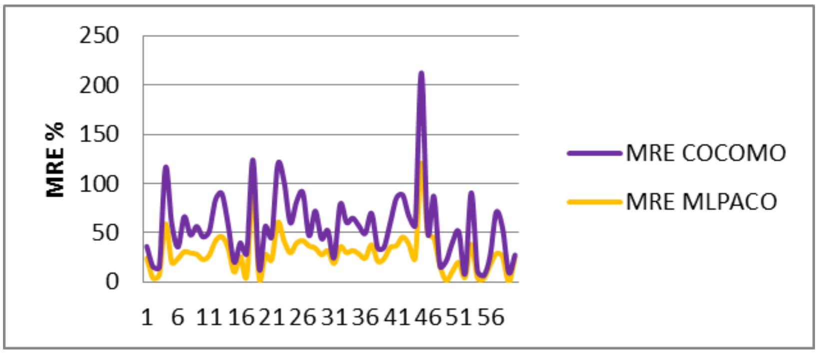

Figure 4. MRE Comparison between MLPACO and COCOMO Models on NASA Projects

As shown in Figure 4, proposed method has lesser MRE error in comparison with COCOMO model on the total data. The comparison of proposed algorithm and COCOMO model on of NASA projects has been shown in Figure 5 (MMRE criteria). Meta-heuristic approaches use extracted knowledge to estimate feature behavior of problem. Therefore, intelligent methods have better performance than algorithmic models to predicate required effort in software developments.

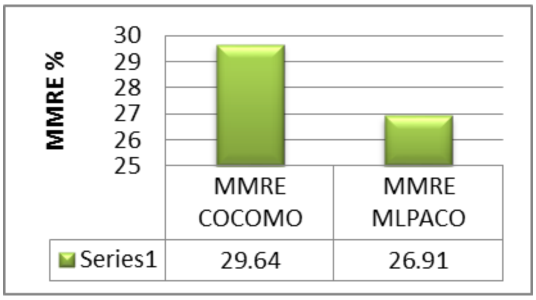

Figure 5. MRE Comparison between Proposed Model and COCOMO model

The results of studies showed that proposed algorithm in comparison with COCOMO model has fewer MMRE. It is necessary to declare that optimization of primary amounts of algorithms highly effective in the accuracy of estimation. Much research into the cost estimation was done for software projects by the application of the techniques of artificial intelligence in recent years. But, it is possible to give a definite conclusion that the methods of artificial intelligence assess exactly 100% of the cost estimation.

6. Conclusion and Future Works

This conclusion was obtained that the methods of artificial intelligence have better efficiency in comparison with algorithmic models according to the studies which have been done. Constant amounts of cost estimation are not specified amounts in algorithmic models but they should be considered in an average way. Thus, it is not easy to trust to the obtained amounts. Various algorithms can be employed for software projects.

Algorithmic models do not have enough accuracy with a view to estimation which MLP ANN and ACO algorithms were used as instruments for estimation in this paper for SCE. Proposed algorithm with the combination of MLP ANN and ACO algorithms called MLPACO model are used for cost estimation of software projects. On the other hand, the results of estimation in proposed algorithm show that estimation in more than

80% of the cases gave positive and highly accurate consequences in comparison to algorithmic model on test data. The results of studies have illustrated that proposed algorithm has fewer MMRE compared to COCOMO model. Also, it should be mentioned that optimization of primary amounts of algorithms have a great influence on estimation accuracy.

As future work, we will try to expand hybrid models in a way that on the basis of previous experiences, can predict SCE accurately and determining accurate metrics and precise criteria to measure SCE.

Acknowledgement

We would like to thank the Islamic Azad University, Mahabad Branch, for Supporting our Research under Research Program (No: 51664911124001).

References

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

[2] A. K. Marandi, D. A. Khan, "An Impact of Linear Regression Models for Improving the Software Quality with Estimated Cost", Vol. 54, pp. 335-342, 2015.

[3] Z. A. Khalifehlou, F. S. Gharehchopogh, "A Survey of Data Mining Techniques in Software Cost Estimation", AWER Procedia Information Technology & Computer Science Journal, Vol:1, pp.331-342, 2012.

[4] T. R. Benala, S. Dehuri, R. Mall, K. ChinnaBabu, "Software effort prediction using unsupervised learning (clustering) and functional link artificial neural networks", 2012 World Congress on Information and Communication Technologies (WICT2012), PP.115-120, IEEE, Trivandrum, 2012.

[5] I. Attarzadeh, A. Mehranzadeh, A. Barati, "Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Effort Estimation", 2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN2012), IEEE, Phuket, PP.167-172, 2012.

[6] T. C. Chen, D. Jian Han, F.T.K. Au, L.G. Tham, "Acceleration of Levenberg-Marquardt training of neural networks with variable decay rate", IEEE Trans. on Neural Net., vol. 3, no. 6, pp. 1873 - 1878, 2003.

[7] 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, 2011.

[8] M. Dorigo, V. Maniezzo, A. Colorni, "Ant system: optimization by a Colony of Cooperating Agents", IEEE Trans. on Systems, Man and Cybernetics, Part B, Vol. 26, No.1, pp. 29-41, 1996.

[9] A. B. Nassif, D. H. Ho, L. F. Capretz, "Towards an Early Software Estimation Using Log-Linear Regression and a Multilayer Perceptron model", The Journal of Systems and Software, Vol. 86, No.1, PP.144-160, 2010.

[10] A. Idri, S. Mbarki, A. Abran, "Validating and Understanding Software Cost Estimation Models based on Neural Networks", 2004 International Conference on Information and Communication Technologies: From Theory to Applications Proceedings ,pp. 433 – 434, 2004.

[11] Z. A. Khalifelu, F. S. Gharehchopogh, "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.

[12] Z. A. Khalifehlou, F.S. Gharehchopogh, "A New approach in Software Cost Estimation Using Regression Based Classifier", AWERProcedia Information Technology & Computer Science”, Vol.2 pp. 252-256, 2012.

[13] 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.

[14] F. S. Gharehchopogh, I. Maleki, S.R. Khaze, " An Novel Particle Swarm Optimization Approach for Software Effort Estimation", International Journal of Academic research, part A, Vol. 6, No. 2, pp. 69-76, 2014.

[15] F.S. Gharehchopogh, L. Ebrahimi, I. Maleki, S. Jodati, "A New 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, Iss. 6, pp. 795-803, 2014.

[16] F.S. Gharehchopogh, Z.A. Dizaji, " A New Approach in Sofware Cost Estimation with Hybrid of Bee Colony and Chaos Optimizations Algorithms", MAGNT RESEARCH REPORT, Vol. 2, Iss. 6, pp. 1263-1271, 2014.

[17] J. Huang, Y-F. Li, M. Xie, "An Empirical Analysis of Data Preprocessing for Machine Learning-based Software Cost Estimation", Information and software Technology, Vol. 67, pp. 108-127, 2015.

[18] Z. B. Mansor, Z. M. Kasirun, N.H.H. Arshad, S. Yahya, "E-cost estimation using expert judgment and COCOMO II", International Symposium in Information Technology (ITSim),VOL.3, PP.1262-1267, 2010.

[19] A. Chavoya, C. Lopez-Martin, M. E. Meda-Campana, "Applying Genetic Programming for Estimating Software Development Effort of Short-scale Projects", 2011 Eighth International Conference on Information Technology New Generations (ITNG2011), pp.174-179, IEEE , Las Vegas, NV, 2011.

[20] A. Kaushik, A. K. Soni, R. Soni, "An adaptive learning approach to software cost estimation", National Conference on Computing and Communication Systems (NCCCS), PP.1-6, 2012.

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

[22] X. Hu, M. Yang, J. Cheng, L. Chen, "A New Product Development Cost Estimation Method Based on the Optimal Weight Combination", Fifth International Conference on Natural Computation (ICNC '09), VOL.2, PP.329-333. Tianjin, 2009.

[23] D. Hunter, A.Yu, M. S. Pukish, "http://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=p_Authors:.QT.Kolbusz,%20J.. QT.&searchWithin=p_Author_Ids:37543168900&newsearch=trueSelection of Proper Neural Network Sizes and Architectures - A Comparative Study ", IEEE Transactions on Industrial Informatics,vol.8,no.2,pp.228-240, 2012.

[24] D. Martens, M. D. Backer, R. Haesen, J. Vanthienen, M. Snoeck, B. Baesens, "Classification with Ant Colony Optimization, IEEE Transactions on Evolutionary Computation", Vol. 11, No. 5, pp. 651-665, 2007.

[25] Menzies T, Port D, Chen Z, Hihn J, Stukes S. Validation Methods for Calibrating Software Effort Models, 27th International Conference on Software Engineering, pp.587-595, 15 May 2005.