Computer & Robotics

A New Approach to Software Cost Estimation by Improving Genetic Algorithm with Bat Algorithm

Sakineh Asghari Agcheh Dizaj [a], Farhad Soleimanian Gharehchopogh [b,*]

[a] Department of Computer Engineering, Bonab Branch, Islamic Azad University, Bonab, Iran

[b] Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

Received 04 July 2017; Revised 17 December 2017; Accepted 20 January 2018; Available online 18 September 2018

Abstract

Because of the low accuracy of estimation and uncertainty of the techniques used in the past to Software Cost Estimation (SCE), software producers face a high risk in practice with regards to software projects and they often fail in such projects. Thus, SCE as a complex issue in software engineering requires new solutions, and researchers make an effort to make use of Meta-heuristic algorithms to solve this complicated and sensitive issue. In this paper, we propose a new method by improving Genetic Algorithm (GA) with Bat Algorithm (BA), considering the effect of qualitative factors and false variables in the relations concerning the total estimation of the cost. The proposed method was investigated and assessed on four various datasets based on seven criteria. The experimental results indicate that the proposed method mainly improves accuracy in the SCE and it reduced errors' value in comparison with other models. In the results obtained, Mean Magnitude of Relative Error (MMRE) on NASA60, NASA63, NASA93, and KEMERER is 17.91, 34.80, 41.97, and 95.86, respectively. In addition, the experimental results on datasets show that the proposed method significantly outperforms GA and BA and also many other recent SCE methods.

Keywords: *Software Cost Estimation, Bat Algorithm, Genetic Algorithm, COCOMO Model, Optimization.*

1. Introduction

Software Cost Estimation (SCE) refers to the process of predicting the cost and time required for developing a software project before initiating it and also predicting the required Cost for preserving the software. In order to SCE, the managers of such projects intend to use optimal and appropriate methods for estimation so that they can completely supervise the progressing of the project, eliminating such problems as shortage of resources, increasing administration costs and undoing the key activities of the project. Nowadays, with the increasing development and alterations in technology, computer and practical applications are more widely used and using software in the majority of working and recreational fields have been regarded very important. The growth of software projects costs for software producers have become a complicated issue. Therefore, in status quo estimating the actual cost of software projects has gained a lot of importance, In addition, the accurate estimates of effort and cost required at the beginning of the life cycle of software is critical for software companies [1]. In order to SCE, first of all, the effective factors on cost estimation must be recognized and if possible an estimation be made

* Corresponding author. Email: bonab.farhad@gmail.com

for any set of factors. The estimation of the amount of work done, the estimation of the needed resources, and an estimation of time must be regarded the most important set of effective factors. Although the size of project is the major factor which determines the amount of activity needed for completing the project; but the relationship between the size and the amount of activity is not clear. The majority of models determine the amount of work based on the size of the project; in practice, this question has been proved but there is not enough evidence to confirm it.

Various algorithms have been presented to SCE that are subcategorized into two category of algorithmic and non-algorithmic. The most famous cost algorithms are COMOCO [2], Putnam's SLIM [3] and Albretcht's Function Point [4]. These models work based on a linear function and a set of input parameters that have a great impact on project and they use features such as LOC and complexity. Basically, the problem with the majority of algorithms for the SCE is that they have a high error rate; so accurate SCE is a very hard task.

From among the presented models, optimal cost model is the most well-known; the model has been made based on regression analysis in order to find the relationship between cost and value drivers and real effort value [5]. This model is a basic method which is used to predict the number of people needed per month to develop software in industry. This model is also able to provide us with an estimation of development time in month. Using this model, we can also estimate the amount of effort needed in each software development phase. Algorithmic models presented by Bohem are available at three levels: Fundamental [2], Intermediate [6], and Advanced [7].

In order to obtain an optimal estimation of software projects, we should recognize the effective factors or the effort coefficient of software projects. Effort coefficient with the following parameters: the amount of effort, the amount of cost estimated, and success or failures of software projects are directly related. Effort parameters in COCOMO model are divided into 4 categories: product (assessment criteria related to software projects); personnel (personnel-related criteria); and project (software-projects-related criteria) each of which has a number of sub-divisions. All these factors in intermediate COCOMO model have ranked qualitative levels. These levels represent the amount of the impact of effort factors. Effort factors are ranked ranging from very low to highly extraordinary. A certain amount is allotted to each area. With a massive expansion of technology in recent years, algorithmic models could not respond to an accurate estimate for software projects and the need for non-algorithmic models based on artificial intelligence techniques and meta-heuristic methods was felt [8].So, in this paper we have used such meta heuristic BA and GAs for the SCE which have already been used by researchers in [9, 10] GAs combination with other algorithms for the SCE. An in [11] using BA in SCE has been presented and we have proposed a model based on enhancement of GA with BA; by investigating them on datasets, we have shown that our proposed method has performed better in the majority of cases in comparison with the two base algorithms. The proposed method covers software projects' quality assurance at all stages including design, development and production, control and design.

The overall organization of this paper is as follows: in the Section 2, we have reviewed the related works; in the Section 3, the proposed method has been fully presented; in the Section 4, the proposed method is completely investigated and assessed; and Section 5, we have dealt with conclusion and future works.

2. Related Works

By reviewing literature, we can find different methods for SCE to some cases of which we will refer. Each of them has its own merits and demerits. A number of these models can be stated as: COCOMO model offered by Berry Bohem in three levels of fundamental, intermediate and advanced. This model is known as one of the most popular, and established Cost estimation models. Among these three versions, intermediate COCOMO model is mainly used in various types of research. By the passage of the time and by the advent of machine-learning algorithms in software engineering, many researchers have used these algorithms to SCE. Therefore, based on the results obtained we can say that the accuracy of estimation in these algorithms are highly improved. Some of these models can be stated as follows:

It has been used unsupervised learning (clustering algorithms) with Functional Link Artificial Neural Networks (FLANNs) for software effort prediction [12]. The unsupervised learning (clustering) indigenously divide the input space into the required number of partitions thus eliminating the need of ad-hoc selection of number of clusters. The FLANNs, on the other hand is a powerful computational model. Chebyshev polynomial has been used in the FLANN as a choice for functional expansion to exhaustively study the performance. Chebyshev polynomials have numerous properties, which make them useful in areas like solving polynomials and approximating functions. Three real life datasets related to SCE have been considered for empirical evaluation of this proposed method. The experimental results showed that proposed method could significantly improve prediction accuracy of conventional FLANN and has the potential to become an effective method for SCE.

F.S. Gharehchopogh in [13] offered a new model based on Artificial Immune System (AIS) with GA for SCE. The proposed method was assessed on NASA60 dataset, and MMRE criterion was regarded as the fitness function in SCE. And MMRE on GA was calculated and assessed to be 15.15, based on AIS algorithm to be 18.20 and based on the proposed method 12.04. The results indicate that the proposed method has operated more appropriately than COCOMO model. And, in [14], the hybrid models of GA and fuzzy logic has been presented for SCE. Trapezoidal and Triangular membership functions are used in fuzzy logic model. To evaluate, four hybrid dataset obtained from NASA2 and COCOMO81 software project are used. The result shows that the MMRE error value in COCOMO model equal to 59.47. That in fuzzy logic model, the error values are 52.90, 56.16, 50.51 and 52.99, respectively. Also, in the GFUZZY hybrid model, the error values are 47.90, 51.08, 48.44 and 47.64, respectively. The results show that the PRED (25) error in COCOMO model on DATASET1, DATASET2, DATASET3 and DATASET4 is equal to 48.38. That in fuzzy logic model, the PRED (25) error values are 45.16, 41.93, 41.93 and 39.78, respectively. Also, in the GFUZZY hybrid model, the error values are 46.23, 46.23, 43.01 and 45.16, respectively. According to the evaluation of criteria such as PRED and MMRE, it can be said that the hybrid model has much better performance.

A model based on GAs combined with neural network Support Vector Machine (SVM) is used to predict the reliability of software [15]. The impact of important parameters of SVM is having crucial roles in its performance. GA is used to optimize the parameters of SVM model. So by taking into account the specific range of these parameters as search space and coding to convert it into a chromosome as decimal in the GA has been done. Evaluation has been done based on MSE criterion. According to the obtained results in this study, it has been proved that the proposed method is better than the other models that use SVM only without optimizing the parameters. And they increased the MSE value to 3.55 for its proposed method.

Oliveira, A.L.I et.al. in [17], used another method based on GAs for feature selection and optimal parameters for machine learning regression to estimate software effort. The study that has been inspired from Wang and Hung's model and reviewed and compared the three techniques of Support Vector Regression (SVR), neural network Multilayer Perceptron (MLP) and tree model based on GA and simulation results show the performance of this method to the recent methods to estimate the software effort. And Z.A. Dizaji and F.S. Gharehchopogh in [18] have used chaos optimization algorithm and Bee Colony Optimization (BCO) for SCE. The assessment of the proposed method was conducted on NASA63. The researchers used Lorentz writing to produce random data as the Chaos Optimization Algorithm and BCO was used for training.

The result obtained from the proposed method equated 0.07 which is indicative of the model`s appropriate operation in comparison with COCOMO model and has less MMRE in contrast to COCOMO model. As an example of a hybrid of COCOMO model and Artificial Bee Colony Algorithm (ABC) for the SCE, we can refer to [18] which have been assessed on NASA63 dataset. In this research, the cost of software projects based on COCOMO model was 58.80, based on ABC it became 38.13 and on the basis of the proposed method it became 32.22.

A. Mitta et al. used FL for SCE [19]. The proposed method of the researchers is thought of as a model in the development of software projects. They utilized 14 existent projects in KEREMER set of projects. The results that they obtained show that the percent of MARE and PRED (N) in the proposed method are improved in comparison with Algorithmic methods. In software projects, obtaining accurate and right Cost estimation requires a lot of parameters. LOC and KLOC are one of software process criteria which directly affect the estimated Cost. In another study, researchers [20] offered a new model based on regression to SCE. They used ISBSG dataset to test and evaluate regression model and the result they ended up with was that MMRE error value in regression model has reduced in comparison with COCOMO model.

MLP networks that are the most common Artificial Neural Networks (ANN) architecture are used mostly for the estimation of software projects. Kaushik et.al. in [21] proposed an adaptive learning technique based on MLP neural network techniques to SCE. And they demonstrated that the performance of a neural network depends on the architecture and setting the parameters. And also, this study explored the effects of parameters' value and type of network topology in order to achieve a high-precision project Cost's estimation model. In addition, they studied the impact of variations of activation function on the accuracy of software projects' Cost estimation. Their proposed method is consistent with the architectural model of post COCOMOII. The artificial neural network model is trained with main projects of two information database of COCOMO and COCOMONASA2. The assessment involves comparing the estimated accuracy of the effort with real effort. The conducted experiments in this study show that the proposed method based on the Magnitude of Relative Error (MRE) is better than the original COCOMO model.

Researchers in [22] have used ANN to train and classify the datasets; GA to give values to the parameters; comparative ANN to test and COCOMO II as the basic model to make comparisons and the results indicate that the accuracy of estimated Cost is improved in comparison with artificial neural network.

A model has been proposed for software effort (person-month) estimation based on three levels Bayesian network and 15 components of COCOMO and software size [23]. The Bayesian network works with discrete intervals for nodes. However, it has been considered the intervals of all nodes of network as fuzzy numbers. Also, authors have obtained the optimal updating coefficient of effort estimation based on the concept of optimal control using GA and PSO for the COCOMO NASA database. In the other words, estimated value of effort is modified by determining the optimal coefficient. Also, it has been estimated the software effort with considering software quality in terms of the number of defects which is detected and removed in three steps of requirements specification, design and coding. If the number of defects is more than the specified threshold then the model is returned to the current step and an additional effort is added to the estimated effort. The results of model indicated that optimal updating coefficient obtained by GA increases the accuracy of estimation significantly. Also, results of comparing the proposed method with the other ones indicated that the accuracy of the model is more than the other models.

In [24] they have made use of Fuzzy Logic Model to SCE. Using fuzzy rectangular numbers as the input parameters for COCOMOII model, they have turned this model into a fuzzy model which is defuzzified after the estimation. The results of this study with COCOMOII model and Alaa Sheta model were compared based on MMRE, PRED (n), VAF methods. The results of the comparisons show the improved accuracy of estimated Cost in comparison with the stated methods.

It has been proposed a new model hybrid based on BA and GA [25]. GA has helped in overcoming global search problem. The results of MMRE are obtained for normal BA based cost optimization, COCOMOII model and hybrid BAT algorithm. The dataset used for this purpose is NASA63. Hybrid model produced the best results. The average MMRE for hybridized BAT is around 23.9% which is better than that of BA based cost optimization which is around 26%. Thus hybrid algorithm produces better results as compared to the performance of original BA.

Another optimized algorithm is presented in [26]. This paper has proposed and implemented Human Opinion Dynamics (HOD) for tuning the parameters of COCOMO model for SCE. HOD is a novel approach to solve complex optimization problems. The input is coding size or lines of code and the output is effort in Person-Months. Mean Absolute Relative Error (MARE) and Prediction are the two objectives considered for the fine tuning of parameters. The dataset considered is COCOMO. The results demonstrated that use of HOD illustrated promising results. It has been observed that when compared with standard COCOMO it gives better results.

In [27], a GA based method was proposed for optimization of the COCOMO model coefficients both for organic and semi-detached modes. In a series of experiments, the GA was tested and the obtained results showed that in most cases the results obtained using the coefficients optimized by the GA are close to the ones obtained using the current coefficients. Comparing organic and semi-detached COCOMO model modes, it can be stated that use of the coefficients optimized by the GA in the organic mode produces better results in comparison with the results obtained using the current COCOMO model coefficients. Results showed that Mean Relative Estimation value using the coefficients optimized by GA was 33.252; Mean Relative Estimation value using current COCOMO model coefficients was 41.387.

In [28], the authors analysed SCE based on Back-Propagation neural networks. The model is designed in such a manner that accommodates the widely used COCOMO model and improves its performance. It deals effectively with imprecise and uncertain input and enhances the reliability of SCE. The model is tested using three publicly available software development datasets. The test results from the trained neural network are compared with that of the COCOMO model. From the experimental results, it was concluded that using the proposed neural network model the accuracy of cost estimation can be improved and the estimated cost can be very close to the actual cost. The model used identity function at the input layer and sigmoidal function at the hidden and output layer. The model used COCOMO dataset and COCOMO NASA2 dataset to train and to test the network.

It have constructed a cost estimation model based on ANN [29]. The neural network that is used to predict the software development effort is the Perceptron network. It is used to COCOM0’81 dataset to train and to test the network. It is observed that the obtained accuracy of the network is acceptable.

3. Proposed Method

The main factor in SCE is estimating the effort factors. The exact amount of estimation is difficult for these factors and also the efficiency of these factors increases the efficiency of the accuracy of the estimates. Algorithmic models such as COCOMO consider only some qualitative factors and do not regard precise value and quantity for them and they are more based on guess and probability [30]. Also, these models only with regarding an objective such as minimizing some cost factors try to develop of software projects. This weakness makes good quality factors are not assessed well and also the single objective of algorithmic models causes to reduce the quality of software development projects. Due to the multi-factor nature of the problem of estimating the Cost of software, the use of non-algorithmic models can be useful. In non-algorithmic models, the optimal mode is used to determine the weight of factors which this causes the estimate be more accurate.

In the proposed method using the operators of BA and GA algorithms, we tried to test and train the effort factors with respect to the size and factors of the project. In Figure (1), the flowchart of proposed method is shown.

In Figure (2), the pseudo-code of the proposed method is shown in which different phases are followed.

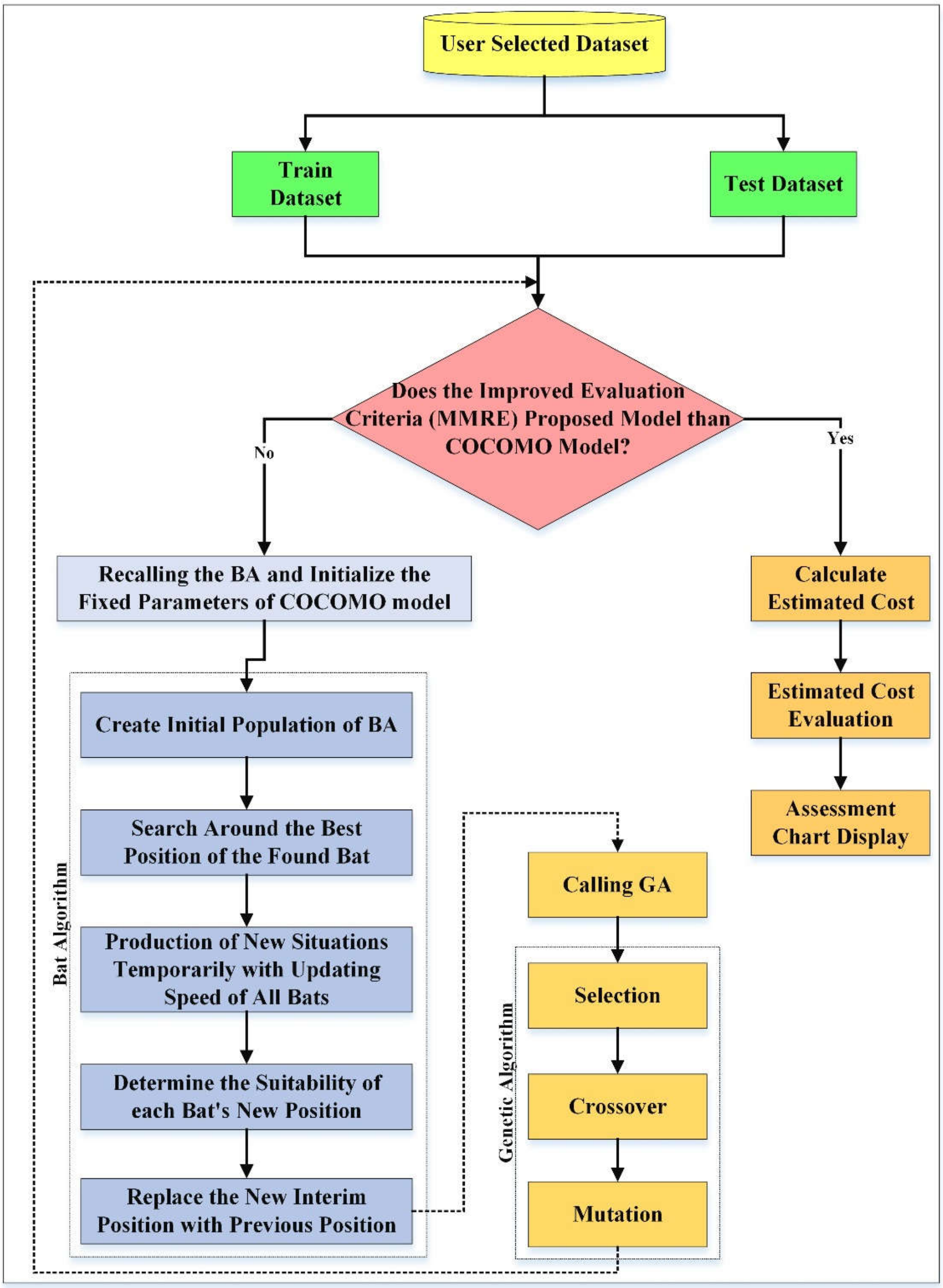

Fig. 1. Flowchart of the Proposed Method

Inputs: the user' selected data set include: factors affecting the estimate and the actual Cost of any software project

Outputs: values of software projects Cost estimation of COMOCO model, GA, BA and the proposed method

Step 1: reading the existed data in the selected data set

Step 2: Breakdown of training and testing data

Step 3: Investigating the recovery criteria to evaluate the proposed method by model COMOCO (fitness function). If values are not optimal, recalling the BA and initialize the fixed parameters of COMOMCO model

Step 4: Create initial population of BA (coefficient factors considered as working bats).

Step 5: search around the best position of the found bat

Step 6: Production of new situations temporarily with updating speed of all bats

Step 7: determine the suitability of each bat's new position

Step 8: replace the new interim position with previous position

Step 9: Submit the best position (the amount of effort shows the optimized coefficients

Step 10: Calling GA

Step 11: Apply GA operators (selection, crossover, mutation)

Step 12: Check the recovery criteria to evaluate the proposed method by COMOCO model, if the answer is positive, save the values.

Fig. 2. Pseudo-code of the Proposed method

In the proposed method, NASA60 [31], NASA63 [31], NASA93 [31], and KEMERER [32] dataset with 60, 63, 93, and 15 projects have been used. The proposed method is based on GA and BAs that at first, we perform the GA with the bat and at the first stage of the proposed method, data set is called and two data sets of training and testing are randomly selected with the proportion of 20% (for testing) and 80% (for training). After improving the training dataset to BA, the training process of the proposed method is began. The training data is used for operations and testing data for evaluation.

In the proposed method, at first, we create the initial population based on the amount of effort factors that generally are in the range [0.9 to 1.4] using BA. After creating the initial population, random coordinates of each bat are specified with parameter and include values of training data. In equation (1), each vector represents the position of the bat. In BA, situations that merit a more efficient advantage are returned as the best answer. With each iteration, each bat speed is updated according to Equation (2). Then, the next positions of each bat are updated according to the Equation (3). Updates include previous position and current speed.

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

$$x_i^t = x_i^{t-1} + v_i^t \tag{2}$$

$$x_{new} = x_{old} + \varepsilon A^t \tag{3}$$

In each algorithm's iteration, based on the best answer, a new position of each bat is updated locally with random step according to the Equation (3). In this equation, the value of ε is equal to 0 and 9 that this parameter is used for uniform search and setting the vectors and vector A includes training data. With this method, any search areas are evaluated and accuracy of algorithm in diagnose is increased.

In order to create a variety for the effort factors, the optimization operations should be carried out by operators of BAs. Initially, each bat finds a position which is the same amount of effort factors in problem space that this amount should be updated and it should be reach to a better position. When those obtained values by BA achieved the nearest solution and the number of iterations is finished, then the operations of GA are started.

The objective of GA is optimization of the obtained values by BA. In the GA, chromosomes are selected that sum of them is less and are better for placement in COMOCO model. The crossover phase in order to avoid local optimization is carried out with the aim of creating a variety of values for effort factors and then using the mutation operator, the places that are not optimized change to the right amount. Finally, the obtained values are placed by the proposed method in equation (4) [33] and MMRE value is calculated.

$$PM = a * (Size)^b * \prod_{i=1}^{15} EM_i \quad (4)$$

In Equation (4) [33], the parameters (a) and (b) are constant coefficients whose value depends on the data set. Size parameter is the project size in terms of number of lines of source per thousand. EM parameters that are called effort factors, in multiple forms reduce or increase the amount of effort [33] in terms of Person/Month (PM) [33]. In the middle COMOCO model, parameters of (a) and (b) are initialized according to Table (1).

Table. 1. Parameters of (a) and (b) various class in the COCOMO model [33]

Class of Projects	**a**	**b**
Organic	3.2	1.05
Semidetached	3.0	1.12
Embedded	2.8	1.20

The Organic class includes relatively small projects that are carried out by teams with high experience. Usually if the project size is 100KSLOC, they are placed in Organic class. Semidetached class includes the modest projects that are neither complicated nor simple and usually if the size of the project is 100KSLOC to 300KSLOC, they are placed in Semidetached class.

The Embedded class includes projects that the size of the projects is more than 300KSLOC. This class is used when the hardware requirements and practices have been identified already, and no change is needed.

In Figure (3), cost factors and relationships between them are shown. In Table (2), effort coefficients are shown with their sizes. As it can be seen, the size of factors is considered for different projects [2].

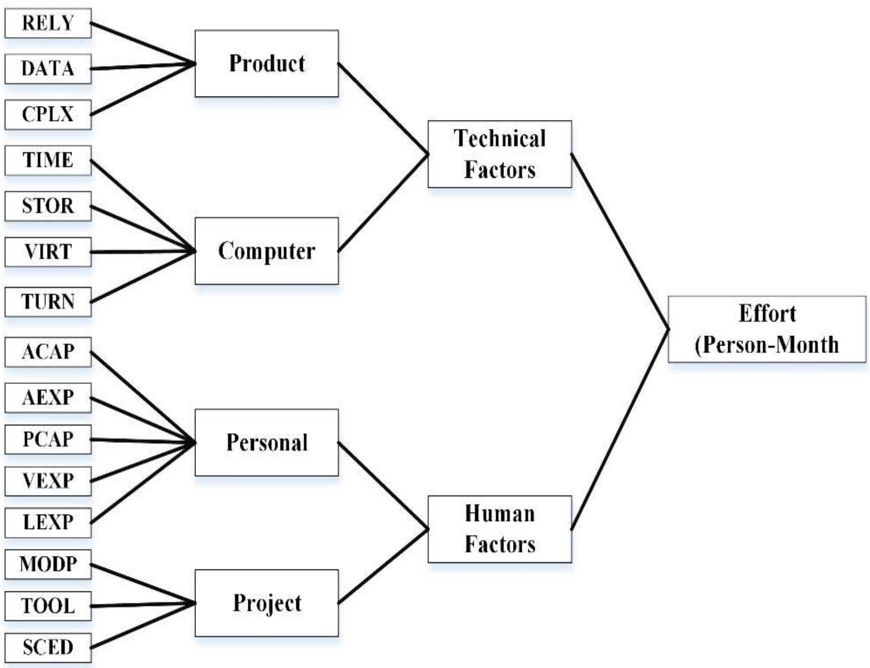

Fig. 3. Cost Factors and Relationships between Them [33]

Table. 2. The Cost factors and their weights in COCOMO II [33]

Cost Factors	Description		Rating				
			Very Low	low	Normal	High	Very High
RELY	Product	Required Software Reliability	0.75	0.88	1.00	1.15	1.40
DATA		Database Size	-	0.94	1.00	1.08	1.16
CPLX		Product Complexity	0.70	0.85	1.00	1.15	1.35
TIME	Computer	Execution Time Constraint	-	-	1.00	1.11	1.35
STOR		Main Storage Constraint	-	-	1.00	1.06	1.21
VIRT		Virtual Machine Volatility	-	0.87	1.00	1.15	1.30
TURN		Computer Turnaround Time	-	0.87	1.00	1.07	1.15
ACAP	Personnel	Analyst Capability	1.46	1.19	1.00	0.86	0.71
AEXP		Application Experience	1.29	1.13	1.00	0.91	0.82
PCAP		Programmer Capability	1.42	1.17	1.00	0.86	0.70
VEXP		Virtual Machine Experience	1.21	1.10	1.00	0.90	-
LEXP		Language Experience	1.14	1.07	1.00	0.95	-
MODP	Project	Modern Programming Practice	1.24	1.10	1.00	0.91	0.82
TOOL		Software Tools	1.24	1.10	1.00	0.91	0.83
SCED		Development Schedule	1.23	1.08	1.00	1.04	1.10

In the proposed method, MMRE is intended as a fitness function. The goal of fitness function in the proposed method is minimizing the MMRE compared with COMOMCO model. Fitness function to the proposed method is defined as equation (5) [9, 10]. In equation (5), parameter y equals to the actual amount and ȳ parameter equals to the estimated amount that is obtained from proposed method.

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|y_i - \bar{y}_i|}{y_i} \times 100\right) \tag{5}$$

$$MMER = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|y_i - \bar{y}_i|}{\bar{y}_i} \times 100\right) \tag{6}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \tag{7}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{8}$$

$$MAPE = \sum_{i=1}^{n}\left(\frac{|y_i - \bar{y}_i|}{y_i}\right) / n \times 100 \tag{9}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \bar{y}_i| \tag{10}$$

Using equation (5), the total error obtained from the effort factors can be estimated. MMRE calculates the mean magnitude of relative error, which measures for a given project the difference between actual (yi) and estimated effort (ẏi) relative to the actual effort. Mean of Magnitude Error Relative (MMER) calculates the mean magnitude of relative error, which measures for a given project the difference between actual (yi) and estimated effort (ẏi) relative to the estimated effort. Mean Squared Error (MSE) is the mean of the square of the differences between the actual and the predicted efforts. The Root Mean Square Error (RMSE) is a measure for the difference between values predicted by a model and the values actually observed from the data that is being modelled. Mean Absolute Error (MAE) and RMSE are two of the most common metrics used to measure accuracy. MAE is equal to RMSE, to this different that MAE do not have radical. The Mean Absolute Percentage Error (MAPE) is commonly used in quantitative prediction methods because it produces a measure of relative overall fit. An effort factor in the proposed method is placed in COMOCO model after optimization and MMRE value is achieved. A model that has less MMRE is better than a model that has a higher MMRE.

4. Result and Discussion

In this section, in order to observe the results, COCOMO model, BA, GA and the proposed method are run on NASA60, NASA63, NASA93, Kemerer datasets; we used global dataset NASA60, NASA63, NASA93, KEMERER to examine the proposed method. The proposed method evaluation is done in VC#.NET 2013. Also, some bats and GAs' parameters that are represented in Table (3), have an important impact on the process of results that their values are determined based on training. The parameters of Np, Nb, Iter, ε, Pc, Pm, Ng, E and S, respectively show the number of initial population, number of bats, number of repetitions, the rate constant search, crossover rate, mutation rate, elitism and percentage of their choice.

All the results are shown in Tables (4-7). Table (4) shows the results of running the proposed method and assessments on NASA60. In NASA60 dataset all assessment criteria of the proposed method are more optimal than COCOMO model. In Table (5), we show the result of running and assessing the proposed method on NASA63. In NASA63 dataset, all assessment criteria except for MdMRE and MMER were more optimal than COCOMO model. MMER and MdMRE criteria had more appropriate performance in comparison with COCOMO model.

Table. 3. Values of the Parameters

Parameters	Value
N_p	50
N_b	
Iteration	100
ε	0.9
Pc	0.7
Pm	0.3
Ng	50
E	0.20
S	30%
Fitness Function	MMRE

In Table (6), we show the results of running the proposed method on NASA93 dataset. In NASA93, all assessment criteria of the proposed method were more optimal than COCOMO model. In Table (7) we ran and assessed the proposed method on KEREMER dataset. In KEREMER dataset, all assessment criteria were more optimal in the proposed method except for MMER criterion. It had also better performance in MMER criterion in comparison with COCOMO model. In order to demonstrate the effectiveness of the proposed method, we selected a number of methods already used to SCE for comparison.

As shown in Tables (4-7), in most cases the proposed method had better and more appropriate operation in comparison with GA, BA, and COCOMOII model. Furthermore, by way of comparing the proposed method with the methods discussed in the second section, we can state that the proposed one had better performance than some of the methods offered. In addition, we can say the proposed method for the functions of GA and BAs and present it as one of the best methods for the SCE.

Table. 4. Evaluation of the Proposed Method on the Nasa60 Dataset

Approach	MdMRE	MAE	MAPE	RMSE	MSE	MMER	MMRE
COCOMO Model	28.23	91.71	29.67	92.31	31908.53	40.18	29.64
GA Model	28.23	86.95	27.04	83.64	31704.41	38.97	25.04
BA Model	25.66	86.63	27.19	83.21	31493.25	41.42	25.19
KNN[36]	14.68	43.3	15.52	83.64	6995.79	16.67	15.52
Cuckoo[36]	15.7	47.14	18.6	83.21	6924.33	14.77	18.6
Gharehchopogh and Miandoab[36]	8	37.47	14.86	67.77	4592.82	13.85	14.86
PSO[37]	14.16	36.61	14.98	67.52	4558.43	13.19	14.98
Hasanluo and Gharehchopogh [37]	14.09	35.10	14.02	65.77	4325.98	12.61	14.02
SEER-SEM[38]	0.27	-	-	-	287180	-	0.57
Wei Lin Du et.al.[38]	0.24	-	-	-	261332	-	0.39
OABE[39]	25.8	-	-	-	-	44.4	61.2
LSE[39]	39.4	-	-	-	-	49.3	58.3
MLFE[39]	44.1	-	-	-	-	53.0	55.7
RTM[39]	36.6	-	-	-	-	80.5	54.9
NN[39]	81.3	-	-	-	-	279.4	99.2
Proposed method	25.09	36.94	16.91	64.77	31706.04	19.30	17.91

Table. 5.Evaluation of the Proposed Method on the Nasa63 Dataset

Approach	MdMRE	MAE	MAPE	RMSE	MSE	MMER	MMRE
COCOMO Model	37.51	210.43	102.55	639.10	40844.85	39.49	36.00
GA Model	43.57	182.43	85.75	586.81	34434.88	42.41	35.75
BA Model	42.27	176.04	82.56	573.92	329382.13	43.66	32.56
KNN[36]	14.68	43.64	15.16	81.44	6633.15	16.67	14.52
Cuckoo[36]	10.56	54.4	16.76	99.63	9925.84	15.21	16.76
Gharehchopogh and Miandoab[36]	13.75	36.49	12.94	66.96	4483.69	13.25	12.94
PSO[37]	10.21	40.48	13.49	75.49	5698.75	12.73	13.49
Hasanluo and Gharehchopogh [37]	10.15	32.89	12.93	64.27	4131.13	11.96	12.93
Proposed method	32.54	110.43	74.80	481.66	32832.46	42.44	34.80

Table. 6. Evaluation of the Proposed Method on the Nasa93 Dataset

Approach	MdMRE	MAE	MAPE	RMSE	MSE	MMER	MMRE
COCOMO Model	48.14	1137.84	115.55	64.00	4096.40	49.05	58.8
GA Model	46.60	807.94	92.94	38.79	1504.70	80.79	42.94
BA Model	49.35	706.50	88.15	27.60	761.54	6.53	38.15
KNN[36]	18.18	40.04	17.24	70.5	4970.08	21.27	17.24
Cuckoo[36]	11.59	31.28	14.87	51.68	2671.27	15.84	14.87
Gharehchopogh and Miandoab[36]	8.69	23.77	11.55	41.06	1686.23	11.03	11.55
PSO[37]	12.37	30.13	13.01	52.83	2717.17	14.47	13.01
Hasanluo and Gharehchopogh [37]	11.08	29.18	12.53	52.18	27940.85	13.35	12.53
UKF-FLANN[12]	0.41	-	-	-	-	-	0.38
DBSCAN-FLANN[12]	0.49	-	-	-	-	-	0.45
FLANN[12]	0.48	-	-	-	-	-	0.49
SVR[12]	0.55	-	-	-	-	-	0.51
RBF[12]	0.55	-	-	-	-	-	0.52
CART[12]	0.66	-	-	-	-	-	1.28
Proposed method	15.14	601.54	91.97	3.85	14.83	5.81	41.97

Table. 7. Evaluation of the Proposed Method on the Kemerer Dataset

Approach	MdMRE	MAE	MAPE	RMSE	MSE	MMER	MMRE
COCOMO Model	415.09	1024.08	502.81	1504.18	339382.84	75.71	502.81
GA Model	100	203.29	94.94	327.28	16067.17	70.11	94.94
BA Model	100	208.06	95.39	330.12	16346.76	27.43	95.39
KNN[36]	36.58	2603	43.76	3809.67	145135	118.44	43.76
Cuckoo[36]	34.77	3111.74	39.99	4919.96	242060	103.81	39.99
Gharehchopogh and Miandoab[36]	27.67	2851.59	35.82	4539.28	206051	91.91	35.82
PSO[37]	43.9	121.32	46.27	262.72	689	59.93	46.27
Hasanluo and Gharehchopogh [37]	35.6	120.56	45.6	261.65	684	59.71	45.6
OABE[38]	23.3	-	-	-	-	51.3	39.6
LSE[38]	21.3	-	-	-	-	59.7	41.4
MLFE[38]	39.6	-	-	-	-	55.5	64.5
RTM[38]	46.1	-	-	-	-	53.8	44.6
NN[38]	128.5	-	-	-	-	73.3	166.0
Proposed method	99.65	107.80	95.86	120.35	8098	156.15	95.86

5. Conclusions

SCE is amongst the most important and at the same time the most complex aspects in project management. One of the major concerns of project managers and designers on planning issues is budgeting and cost controlling. In software projects, costs are directly or indirectly related to the environment of the project and these factors have partial impact on the total function of projects cost. Although, direct costs are mainly fixed costs but they can be part of variable costs. In this paper, in order to analyze these Costs, we have made use of such Meta heuristic algorithms as BA and GAs and offered the proposed method as an informed GA with BA; the proposed method was run and assessed on Kemerer, NASA93, NASA63, NASA60 in terms of 7 assessment criteria including: MAE, MAPE, RMSE, MSE, MMER, MMRE, and MDMRE. According to the results obtained, we can state that the proposed method on Kemerer, NASA93, NASA63, NASA60 were approximately more optimal in all criteria than COCOMO model. However, on NASA63 dataset, in MMER and MDMER criteria and on Kemerer dataset in MMER criterion it did not have optimal operation in comparison with COCOMO model. We are hopeful that in the future using other Meta heuristic algorithms we can obtain a combination to solve various problems in software technology and boost and/or improve software development.

References

[1] Hu, X.; Yang, M.; Cheng, J.; Chen, L., "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).

[2] Boehm, B. W., Software engineering economics, Englewood Cliffs, NJ: Prentice-Hall (1981).

[3] Putnam, L. H., "A General Empirical Solution to the Macro Software Sizing and Estimating Problem". IEEE Transactions on Software Engineering, SE-4(4) pp. 345-361 (1978).

[4] Cot, V.; Bourque, P.; Oligny, S.; Rivard, N., "Software Metrics: an Overview of Recent Results", in Journal of Systems and Software, Vol. 8, pp.121-131 (1988).

[5] Sheta, A.; Rine, D.; Ayesh, A., "Development of Software Effort and Schedule Estimation Models Using Soft Computing Techniques" IEEE Congress on Evolutionary Computation, pp. 1283-1289 Hong Kong (2008).

[6] Li, Z., "Intelligently Predict Project Effort by Reduced Models", International Conference on E-Business and E-Government (ICEE), pp.1536-1542, Guangzhou (2010).

[7] Boehm, B. W., "Anchoring the Software Process", IEEE Software, 13, 4 July 1996, pp. 73-82 (1996).

[8] Boehm, B. W., "Making RAD Work for Your Project", extended version of March 1999, IEEE Computer Column; USC Technical Report USC- SCE-99-512 (1999).

[9] Kaur, J.; Singh, S.; Kahlon, K. S.; Bassi, P., "Neural Network – A Novel Technique for Software Effort Estimation", International Journal of Computer Theory and Engineering (IJCTE), vol. 2, no. 1, pp. 17-19 (2010).

[10] Maleki, I.; Ebrahimi, L.; Gharehchopogh, F.S., "A Hybrid Approach of Firefly and Genetic Algorithms in Software Cost Estimation", MAGNT Research Report, vol. 2, no. 6, pp. 372-388 (2014).

[11] Maleki, I.; Gharehchopogh, F.S.; Ayat, Z.; Ebrahimi, L., "A Novel Hybrid Model of Scatter Search and Genetic Algorithms for Software Cost Estimation", MAGNT Research Report, vol. 2, no. 6, pp. 359-371 (2014).

[12] Ibrahim, M., "A New Model for Software Cost Estimation Using Bat Algorithm", International Journal of Academic Research in Computer Engineering, vol. 1, no. 1, pp. 53-60 (2016).

[13] Rao Benala, T.; Dehuri, S.; Mall, R.; ChinnaBabu, K., "Software Effort Prediction Using Unsupervised Learning(Clustering) and Functional Link Artificial Neural Network", World Congress on Information and Communication Technologies, pp. 115-120 (2012).

[14] Gharehchopogh, F.S.; Maleki, I.; Ghoyunchizad, N.; Mostafaee, E., "A Novel Hybrid Artificial Immune System with Genetic Algorithm for Software Cost Estimation", MAGNT Research Report, vol. 2, no. 6, pp. 506-517 (2014).

[15] Hamdy, A., "Genetic Fuzzy System for Enhancing Software Estimation Models", International Journal of Modeling and Optimization, vol. 4, no. 3, pp. 227-232 (2014).

[16] Lo, J-H., "Early software reliability prediction based on support vector machines with genetic algorithms" The 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 2221 – 2226, Taichung (2010).

[17] Oliveira, A.L.I.; Braga, P.L.; Lima, R.M.F.; Cornélio, M.L., "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation", Information and Software Technology, vol. 52, no. 11, Pages 1155-1166 (2010).

[18] Dizaji, Z.A.; Gharehchopogh, F.S., "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).

[19] Gharehchopogh, F.S.; Maleki, I.; Kamalinia, A.; Zadeh, H. M., "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).

[20] Mittal, A.; Parkash, K.; Mittal, H., "Software Cost Estimation Using Fuzzy Logic", ACM SIGSOFT Software Engineering, vol. 35, no. 1, pp. 1-7 (2010).

[21] Adalier, O.; Ugur, A.; Korukoglu, S.; Ertas, K., "A New Regression Based Software Cost Estimation Model Using Power Values", Intelligent Data Engineering and Automated Learning (IDEAL), Lecture Notes in Computer Science, vol. 4881, pp. 326-334 (2007).

[22] Kaushik, A.; Soni, A.K.; Soni, R., "An Adaptive Learning Approach to Software Cost Estimation", National Conference on Computing and Communication Systems (NCCCS), PP.1-6 (2012).

[23] Benala, T. R.; Dehuri, S.; Satapathy, S. Ch.; Sudha Raghavi, Ch., "Genetic Algorithm for Optimizing Neural Network Based Software

Cost Estimation", Lecture Notes in Computer Science, pp.233–239 (2011).

[24] Zare, F.; Zare, H.K.; Fallahnezhad, M.S., "Software effort estimation based on the optimal Bayesian belief network", Applied Soft Computing, vol. 49, pp. 968-980 (2016).

[25] Ziauddin, N.; Kamal, S.; Khan, S.; Abdul Nasir, J., "A Fuzzy Logic Based Software Cost Estimation Model", International Journal of Software Engineering and Its Applications, vol. 7, pp. 7-18 (2013).

[26] Gupta, R.; Chaudhary, N.; Pal, S.K., "Hybrid model to improve Bat algorithm performance", International Conference on Advances in Computing Communications and Informatics (ICACCI), IEEE, New Delhi, India, pp. 1967-1970 (2014).

[27] Guleria, D., "A Novel Evolutionary Approach to Software Cost Estimation", International Journal of Computer Applications, vol. 126, no.4, pp. 47-52 (2015).

[28] Galinina, A.; Burceva, O.; Parshutin, S., "The Optimization of COCOMO Model Coefficients Using Genetic Algorithms", Information Technology and Management Science, vol. 15, pp. 45-51 (2012).

[29] Kaushik, A.; Soni, A.K.; Soni, R., "A Simple Neural Network Approach to Software Cost Estimation", Global Journal of Computer Science and Technology Neural & Artificial Intelligence, vol. 13, Issue 1 (2013).

[30] Kaushik, A., "COCOMO Estimates Using Neural Networks", International Journal Intelligent Systems and Applications, Published Online August 2012 in MECS, vol. 9, pp. 22-28 (2012).

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

[32] http://promise.site.uottawa.ca/SERepository/datasets-page.html [Last Available: 2017.08.08].

[33] http://openscience.us/repo/effort/function-points/kemerer.html[Last Available: 2017.08.08].

[34] Menzies, T.; Port, D.; Chen, Zh.; Hihn, J., "Validation Methods for Calibrating Software Effort Models", ICSE ACM, USA (2005).

[35] Capretz, L.F.; Marza, V., "Improving Effort Estimation by Voting Software Estimation Models", Hindawi Publishing Corporation, Advances in Software Engineering, pp. 1-8 (2009).

[36] Gharehchopogh, F.S.; Maleki, I.; Sadouni, S.; "Artificial Neural Networks Based Analysis of Software Cost Estimation Models", MAGNT Research Report, vol. 2, no. 6, pp. 597-505 (2014).

[37] Miandoab, E.E.; Gharehchopogh, F.S., "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).

[38] Hasanluo, M.; Gharehchopogh, F.S., "Software Cost Estimation by a New Hybrid Model of Particle Swarm Optimization and K-Nearest Neighbor Algorithms", Journal of Electrical and Computer Engineering Innovations, vol. 4, no. 1, pp. 49-55 (2016).

[39] Du, W.L.; Capretz, L.F.; Nassif, A.B.; Ho, D., "A Hybrid Intelligent Model for Software Cost Estimation", Journal of Computer Science, vol. 9, no. 11, pp.1506-1513 (2013).

[40] Azzeh, M.; Elsheikh, Y.; Alseid, M., "Optimized Analogy-Based Project Effort Estimation", International Journal of Advanced Computer Science and Applications, vol. 5, no. 4, pp. 6-11 (2014).