International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 5 No. 1 Jan. 2014, pp. 72-81

http://www.issr-journals.org/ijias/

A New Approach for Software Cost Estimation with Hybrid Genetic Algorithm and Ant Colony Optimization

Isa Maleki[1], Ali Ghaffari[2], and Mohammad Masdari[3]

[1]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, West Azerbaijan, Iran

[2]Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

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

ABSTRACT: One of the most important effective factors the software companies face is the Software Cost Estimation (SCE) in software development process time. SCE is one of the subjects which have been considered in late decades in many researches. The real estimation in software development needs effort and cost factors which are done by use of the algorithmic and Artificial Intelligence (AI) models. Boehm used the COCOMO model which is an algorithmic model in 1981 for SCE. The low accuracy and non-reliable structures of the algorithmic models led to high risks of software projects. So, it is needed to estimate the cost of the project annually and compare it to the other techniques. The Meta-Heuristic algorithms have been developed well lately in software fields and SCE. Meta-heuristic and Genetic Algorithms (GA) and Ant Colony Optimization (ACO) solve the problems according to the optimization of the problems and are very efficient in optimizing the algorithmic models and the effective factors in cost estimation. In this paper we have proposed a hybrid model based on GA and ACO for optimization of the effective factors' weight in NASA dataset software projects. The results of the experiments show that the proposed model is more efficient than COCOMO model in software projects cost estimation and holds less Magnitude of Relative Error (MRE) in comparison to COCOMO model.

KEYWORDS: Software Cost Estimation, COCOMO, Artificial Intelligence, Meta-Heuristic Algorithms, Genetic Algorithm, Ant Colony Optimization.

1 INTRODUCTION

One of the most effective and important factors in development of the software projects, is SCE. In development of the software projects, there are many effective factors value and limits of which must be identified using the accurate estimation. The highest costs of software projects and totally the vast value of the costs are related to the work forces. So, the most important problem in software engineering is the cost estimation and the needed effort for development of the software projects.

The accurate estimation of the costs of software projects causes the projects be done in format of the identified time and costs [1]. With no suitable estimation about the costs for development of the software projects, the project manager would not be able to diagnose the how much time or cost is needed for the project and if there any mistake happens, the project would be defeated or face risks [2]. The accurate cost estimation and the reliable one especially at the beginning of the project, is an important factor for the success of the project. The accuracy of the software cost estimation causes project manager makes powerful decisions in the lifecycle of the software. Also the project manager, analyzer, designer, programmer and all other work forces of the software development must know that how much effort and cost are needed for production of software. SCE is one of the most critical tasks in managing software projects. Development costs tend to increase with project complexity, and hence accurate cost estimates are highly desired during the early stages of development. An important objective of the software engineering community has been to develop useful models that constructively explain the software development life-cycle and accurately estimate the cost of software development. So, the time estimation would be effective when it is enough for the software development needs. The software needs are being changed day by day, but the effect of the time changes which are identified via the accurate estimation will lead to the reduction of the time needed for software projects development.

Algorithmic models were presented in the early times of the software development. COCOMO model was first introduced by Boehm in 1981 [3]. COCOMO II is a model of estimation of time and costs in software projects. COCOMO II model is used for decision making in different software projects [4]. COCOMO II model works better for software projects estimation and is innovated for the projects in which different factors are effective. In COCOMO II, the parameters value is achieved by the experience of the previous projects like COCOMO 81.

Now the meta-heuristic algorithms are used vastly in hybrid optimization problems. One of the basic applications of these algorithms is to contribute the optimization and efficiency of the optimized solutions [5]-[6]-[7]-[8]-[9] and [10]. Meta-heuristic algorithms are the algorithms which search the near optimized answers in the problems spaces indecisively. These algorithms are very efficient in solving the hard and complex problems. A usual procedure which can be used in estimation of the software projects is to use a favored function with algorithmic methods for finding the values of the cost estimation which lead to the upmost favor. When dimensions of the problem go high by the increase of the numbers of the factors and the variables of the problem, the algorithmic methods will be unable to achieve the real answers. So, in this paper we have used the combination of the GA and ACO to present a new model for estimation of the software projects.

We have organized the paper as follows: in Section 2, we have introduce the related works; in Section 3, the COCOMO model is introduced; in Section 4, we have introduced the meta-heuristic algorithms; in Section 5, the proposed algorithm is described; in Section 6, the evaluation and the results of the proposed algorithm are presented and at finally in Section 7, we have presented the conclusion and future works to be done.

2 RELATED WORKS

Lately many researches have taken place in cost estimation field for the software projects using the AI techniques. But it is not possible to say that AI is 100% percent to estimate the costs accurately. But according to the studies, we have resulted that the AI techniques have been more efficient in comparison to the algorithmic techniques. COCOMO is a model for cost and time estimation of the software projects among the algorithmic methods.

In [11] has used the Artificial Neural Networks (ANNs) for estimation of the software projects estimation. In this research, 11 projects of 60 projects of NASA dataset [12] have been tested using ANNs. And they were compared to COCOMO model and it is shown that COCOMO model error is more than ANNS in many cases. The results show that in more than 90 % of the cases, ANNs has presented better estimation than COCOMO model. So, it is possible to conclude that the methods based on AI are as complements and good replacement for the algorithmic methods. For accurate study of cost estimation in software projects, KEMERE [13] has studied the FP, COCOMO, SLIM and ESTIMACS patterns. According to the results of his paper a high error percent is seen in all four models. So it is possible to say that the estimation models are very effective in accuracy of effort and costs and the ending of the project in a defined time. In [14], it has utilized the GA for the optimized value of the COCOMO model parameters. One of the problems of COCOMO model is identifying the optimized value for the parameters. In this research, the projects of NASA dataset are used for presenting the better efficiency of the proposed model. Also, the two DLOC and ME factors which are very effective in estimation of costs, are studied on 18 projects. According to the results it is possible to say that using GA, it is possible to achieve better estimation. Researchers in [15] have studied the SCE using ANNs. Studying the results shows that ANNs are very good in estimation of the costs of software projects. Some other researches [16] have studied the three methods based on Machine Learning (ML) like ANNs, Case-Based Reasoning (CBR) and Rule Induction (RI). They have studied the cost estimation of the software projects using 77 projects. The results of this study show that ANNs are more accurate than other methods.

In [17] has tested SEE using GA. In this method, it is cited that COCOMO model is not good in effort estimation in comparison to the other models. So, it is tried to make better the value of the parameters in proposed model and make effort estimation more accurate. In this research, the NASA dataset software projects are used for the results of the experiments. According to the results, the proposed model has achieve better estimation and has made Mean Magnitude of Relative Error (MMRE) value to 0.2298% in comparison to COCOMO model. Researchers in [18] have tested the software projects cost estimation using the soft computing techniques. They have used Fuzzy Logic (FL) in their paper and also the Particle Swarm Optimization algorithm for better cost estimation. They have used 30 projects of NASA dataset for their results of experiments. According to the results, the suggested model has reached better estimation and has made MMRE up to 7.512% in comparison to the other models. Researchers [19] have used FL for estimation of the software projects. They have introduced SCE as one of the challenges and the important activities in software development. Their proposed method shows that using FL is a model in software development. They have used 14 projects of KEMERE projects set. According to their results it is possible to say that Mean Absolute Relative Error (MARE) and PRED (N) are better in proposed model than algorithmic methods. Cost function has many parameters in software projects. Some of the factors of software process which are directly effective on cost estimation are: Line of code (LOC) and Kilo Line of Code (KLOC). Researchers [20] have used Multi-Objective Particle Swarm Optimization (MOPSO) algorithm for SCE. They have minimized MARE using MOPSO algorithm for optimization of the parameters of COCOMO model. For more studying of the results, the proposed model was tested on small and large projects. According to the experiments, MARE is 16.1306% in small projects in COCOMO model and 9.0143% in proposed model, and 18.1548% in large projects in COCOMO model and 20.9717% in proposed model. The results of experiments show that proposed model is better in estimation. Researchers [21] have used data mining techniques and algorithmic models to study and evaluate the SCE. One of the most critical subjects in software development is the right estimation of the software costs. They have studied COCOMO model using the data mining techniques. SCE using four data mining techniques are Linear Regression (LR), ANNs, Support Vector Regression (SVR) and K-Nearest Neighbors (KNN). Using LR model it is possible to identify the dependency of the effective adjectives in SCE. LR model finds the relationship between independent and dependent factors among the data. ANNs try to train and test the data to make more accurate cost estimation. SVR model is used for optimization of the effective factors in SCE. KNN is a technique in data mining which is used for classification of the data in a set of them which were classified before and hold specified characteristics. Using KNN, the weight of the effective adjectives in SCE is identified. The results of the experiments show that SVR model holds less MRE than others. Researchers [22] have proposed a new model based on regression for cost and effort of software development. The efficiency of the proposed model is evaluated on NASA projects. The results of the experiments show that regression model has lower MRE and is more effective in calibration of the COCOMO model. Researchers [23] have studied data mining techniques in the SCE. They have studied ANNS, LR, Multiple LR (MLR), Bayesian Networks (BN), Fuzzy Decision Trees (FDTs), FL and Neuro Fuzzy (NF). Any input factor is weighted using ANNs and is calculated by the hidden layers and in output layer the optimized value is reached, in MLR and BN models, the estimation takes place using analyzing the dependent variables. In FDTs, FL and NF models, any estimation factor is considered a fuzzy member.

SCE has developed using the AI models and it is possible to say firmly that the AI methods are more accurate than the algorithmic models. AI models are repeated continuously and train the data repeatedly and are able to optimize the effective factors in estimation and minimize the cost and effort of the software projects.

3 COCOMO MODEL

COCOMO is a model of cost and time estimation in software projects. It is an experienced model which is achieved by gathering the data from many software projects [3]. Base COCOMO model is not suitable in cost estimation because of many changes in software development projects and so the intermediate COCOMO was presented for improving the base COCOMO model. In intermediate COCOMO the cost estimation calculation for the software projects is identified by the equation (1) [24, 12].

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

In equation (1), a and b are the constant parameters the value of which depend on the data of the dataset. Parameter Size is the size of the project in Thousands of Source Lines of Codes (KSLOC). Parameter EM which is named Effort Multipliers (EM) is a coefficient which causes increasing or reducing the effort rate in person/month [12]. In intermediate COCOMO parameters a and b are initialized according to Table (1).

Table 1. a and b Values in Different Classes

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

Organic class includes relative small projects which are done by the high experienced teams. Usually if the projects sizes are 100 KSLOC, they fall in organic class. Semidetached class includes medium projects which are not complex or simple and if the size of the projects is 100 to 300 KSLOC, fall in semidetached group. Embedded class includes projects the size of which is more than 300 KSLOC. This class is used when the hardware and operations are defined before and there is no need for any changes.

4 META-HEURISTIC ALGORITHMS

Nowadays the use of the meta-heuristic algorithms in optimizing the problems and achieving an accurate solution has grown very much. Because of the increase of the complexity of the optimization problems and disability of the algorithmic methods in optimized solution, the meta-heuristic algorithms are a suitable method for optimization problems. GA and ACO are the algorithms which solve the optimization problems according to the population. These algorithms hold searching and evaluation specification and try to repeat the problem many times until reaches the optimized answer. In this section, we will study the GA and ACO.

4.1 GENETIC ALGORITHM

GA is one of the meta-heuristic algorithms which were first by John Holland in 1975 for the first time [25]. GA is an optimization algorithm which takes into consideration a set of the points of answer space in any calculation repetition effectively and searches different areas of answer space. Execution of this algorithm starts with creation of the initial population. Any individual of the population which is called a chromosome is considered as an answer for optimization problem and any element of the chromosomes which is called gene shows the value of the optimization problem. The new generation takes into consideration the suitability of the chromosomes function and is created by using the GA operators (crossover and mutation) and the suitability function of the individuals is improved in any repetition. GA execution includes the following stages:

- **Initial Population**: In primary state, a population of the chromosomes is created randomly. In production of the initial population, the limits of the problem must be considered. Any chromosome is a solution for optimization problems.
- **Evaluation:** This operator uses an evaluation function for identification of suitability of any chromosome in relation to the other chromosomes. In fact this operator shows the suitability of the chromosome for solving the problems.
- **Selection:** This operator is the most suitable chromosome from the population for the next generation. If a chromosome does not meet the limits of a problem, will not be selected.
- **Crossover:** Production of the new chromosomes from the selected chromosomes takes place by the crossover operator.
- **Mutation:** This operator changes the random genes from the random chromosomes. The probability of its incidence on the population is low and the goal of it is to maintain the genetic diversity of the population in convergence to the optimized solution.

GA is a meta-heuristic algorithm which takes into consideration a set of the solution space points in each calculation repetition effectively ad searches the different solution spaces. In GA, select, crossover and mutation operators cause new points of the solution space to be searched in each calculation repetition. GA operators ban the solutions to be locally convergent and also cause searching ability increase and finding optimized points of the searching space and finally leads increase of the searching ability of the algorithm.

4.2 ANT COLONY OPTIMIZATION

ACO is one of the most known meta-heuristic algorithms. ACO algorithm was first presented by Dorigo in 1996 [26]. ACO algorithm is inspired from the natural life of the ants. Ants leave an odorous stuff on the path named pheromone. This stuff evaporates but is left in short time as the ant trace on the earth. The ants are able to produce pheromone to find the nearest path to the food. The ants selecting the nearest path create more pheromone than the ones selecting the longer paths. As the more pheromone attracts more ants, the more and the more ants will select the shorter path and then all ants will find the shorter path and move on it. To study the subject more, we assume that there are two paths on the way to the food which are different in length. The ants select both paths by the same probability. The ants which have been on the shorter path will produce the most pheromone before the others. So the other ants will select this path and will improve the pheromone of this path. At last all ants will find the shortest path to the food.

4.2.1 THE RULE OF TRANSITION PROBABILITY

The probability of movement from city *i* to city *j* for ant *k* in time *t* is calculated according to the equation (2). In equation (2), η_{ij} is the visibility which is $1/d_{ij}$ (the nearer cities will be selected with higher probability).

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[\eta_{ij}\right]^{\beta}}{\sum\limits_{j\in allowed_k}\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[\eta_{ij}\right]^{\beta}} & \text{if } k\in allowed_k\\ 0 & otherwise\end{cases} \tag{2}$$

In equation (2), τ_{ij} is the pheromone volume poured on any mane and α and β are the poured pheromone effect on mane and the scope of that mane, respectively. And $allowed_k$ is the set of the cities ant *k* has not moved to till now.

4.2.2 RULES OF PHEROMONE UPDATE

To pay attention to the other ants like the best ant, and to be able to use the valuable information of them, the local updating method is used for updating pheromone rule. In this method the search space is gone fast and on fact the risk of losing the suitable paths and getting into local minimum trap are gone less. The updating law of pheromone on the manes takes place according to the equation (3).

$$\tau_{ij}(t+n)=(1-\rho)\times\tau_{ij}(t)+\Delta\tau_{ij} \tag{3}$$

In equation (3), (1-ρ) is the evaporation rate of the pheromone in t distance to (t + n). To avoid the increase of the pheromone on one mane, the 0<ρ<1 limit is used for ρ. The more ρ, the more will be the evaporation speed.

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{4}$$

$\Delta\tau_{ij}$ Is the pheromone volume the ant *k* leaves on (i, j) path in time *t* to t + n which is identified as follows.

$$\Delta\tau_{ij}^{k}=\begin{cases}Q/L_k & \text{if k ant uses edge (i,j) at time (t,t + n)}\\ 0 & otherwise\end{cases} \tag{5}$$

In equation (5), parameter Q is the increasing volume of pheromone in Local Updating and L_k is the path length the ant *k* moves.

5 PROPOSED MODEL

By the software development, there are high risks of error in software projects cost estimation using the COCOMO algorithmic models and so this process is very hard. In algorithmic models, the cost estimation constant values are not defined values and are considered as the mean values and so it is not easy to find reliable answers. The relations of the factors of estimation in software projects are very effective in software projects success. For software projects it is possible to balance the estimation factors and estimate the most accurate cost for the software projects. For software projects it is possible to use different algorithms. In this paper we have tried to hybrid the meta-heuristic algorithms for estimation of the software projects. The software projects cost estimation using meta-heuristic algorithms cause software upgrade, software project control and software quality. The problem of software projects development is the numeral estimations like costs and the effort needed for the projects. At first stage of the development of software projects, there is no accurate information accessible about the system operation, limits and the responsibilities of the projects. Using the GA and ACO it is possible to hybrid the different factors like hardware, software and human forces for development of the software projects development.

In estimation of the software projects costs, some of the project factors are varying. For example the programming factor in software projects is a team activity and there is no total process for it. Some of the programmers use the techniques in programming which need les codes and some others use the techniques of more codes and spend more time on programming. So, the attention to the programming bases leads to the reduction of development time and increase of utilization of the software projects. By reducing the codes costs, the produced software will be more efficient from cost point of view. In the proposed model we test the costs of estimation [12] using the GA and using the ACO, the training operation is done for making the parameters more optimized. In estimation accuracy the EM factors value are very important [12]. In proposed model, we use the GA and test the suitable value of these factors according to the projects' size and then use the ACO to train more optimized factors. In GA, the longer the gene length in chromosomes, the mower the estimation error and the more accurate results will be accessible for the cost estimation. In Fig. 1 the flowchart of the proposed model is shown.

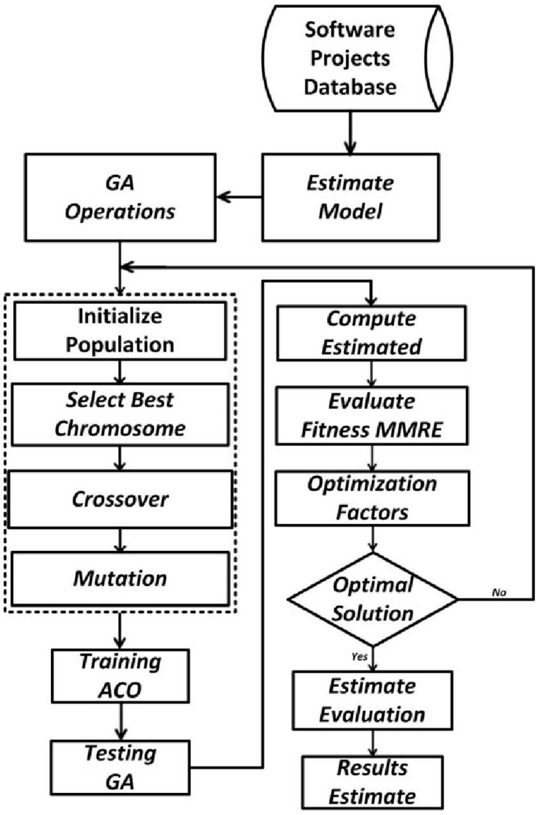

Fig. 1. Flowchart of the Proposed Model

In Fig. 2, the quasi code of the proposed model is shown.

1. Initialize Parameters
2. Do
Initialize population
Select best chromosome
Crossover
Mutation
3. Training ACO
4. Testing GA
5. Compute Estimate
6. Evaluate Fitness
MRE=|Estimate – Actual| / Actual;
MMRE=∑ (MRE) / N;
7. While (not terminate condition)
8. Estimate Evaluation
9. Results Estimate

Fig. 2. Quasi Code of Proposed Model

In the proposed model, MMRE is set as the fitness function. The suitability function fitness is to minimize the MMRE in comparison to the COCOMO algorithmic model in proposed model and the algorithm is repeated till the MMRE is reduced down to the favored rate and then the best chromosome and gene value of it is selected. MMRE is defined according to the equation (7) [27].

$$MRE_i = \frac{|Actual_i - Estimate_i|}{Actual_i} \times 100 \tag{6}$$

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \tag{7}$$

Using the equation (7), it is possible to compare the error sum of the different models of estimation. The SCE using the combination of GA and ACO is implemented in C#.NET 2008 programming environment. In Fig. 3, the scheme of the program is shown.

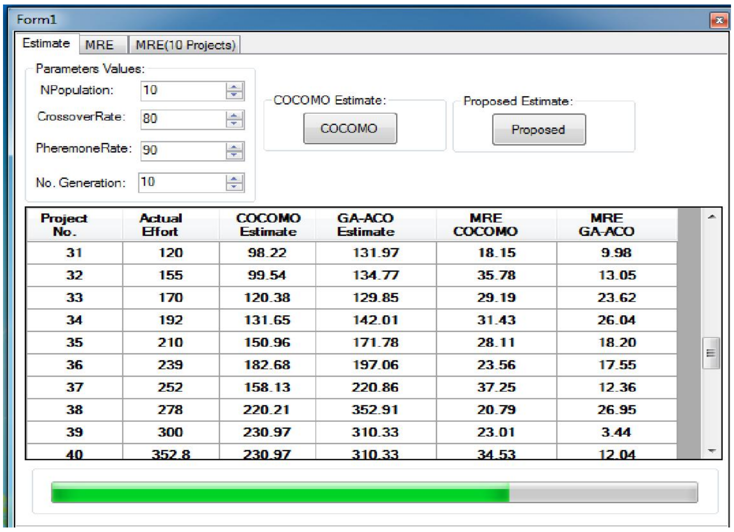

Fig. 3. The Program of the Proposed Model for SCE

6 EVALUATION AND RESULTS

In this section, the results of proposed model and COCOMO model are evaluated. In meta-heuristic algorithms, it is important to identify the primary parameters for evaluation of the results. So, meta-heuristic algorithms are very delicate in their parameters and setting the parameters is very effective in their operation. So, setting the parameters leads more flexibility and efficiency of the proposed model. The population selection is very important in meta-heuristic algorithms. If the number of the population is low, the problem will be soon convergent and will not achieve the favored and near global optimum solution and if the population number is high, long time will be spent for the algorithm to be convergent. So, the number of the population must be suitable and in harmony to the problem to reach the optimized solution. In Table (2), the parameters affecting the operation of the proposed model are shown.

Table 2. Parameters Value

Parameters	Value
No. Population	10
Crossover	0.6
Mutation	Randomized
Pheromone Rate	0.9
Population Rang	0.9-1.4
No. Generation	10
Fitness Function	MMRE

To show the efficiency better in Table (3), 10 projects of the NASA dataset software projects are evaluated. The results of the experiments in Table (4) show that the proposed model holds lower MRE than COCOMO model. So, proposed model is suitable for estimation and has less estimation error than COCOMO model.

Table 3. MRE Comparison of the Proposed Model and COCOMO Model on 10 Projects

No	Project No.	MRE using COCOMO	MRE using Proposed
1	9	28.08	16.60
2	11	26.99	10.56
3	13	46.02	33.88
4	16	25.15	8.67
5	28	34.14	19.32
6	37	37.25	12.36
7	43	37.64	15.30
8	47	44.43	31.93
9	57	28.79	23.16
10	60	25.57	27.50

In Fig. 4 the MRE comparison of the proposed model to the intermediate COCOMO model on 10 projects of the NASA dataset software projects is shown. As it is clear, the suggested model holds less estimation error than COCOMO model.

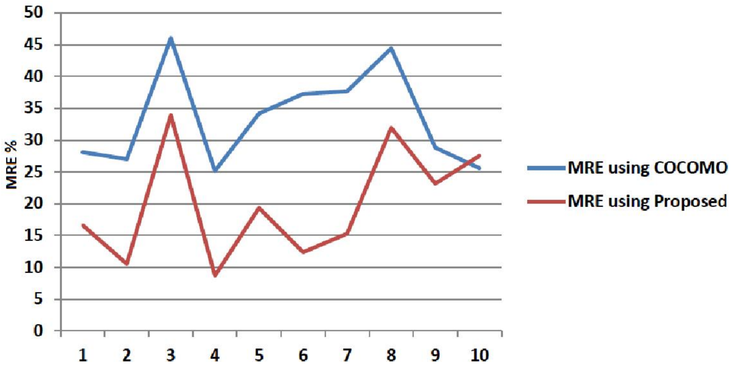

Fig. 4. MRE Comparison of the Proposed Model and COCOMO Model on 10 Projects

Fig. 5 shows the comparison diagram of MRE of proposed model with effect of the number of the different generations to minimize the error.

Fig. 5. MRE Comparison of the Proposed Model Affected by the Number of the generations on 10 Projects

As it is clear in Fig. 5, the proposed algorithm holds more ability in increasing the number of the generations in minimizing the MRE. In Fig. 6 the diagram of comparing the MMRE of the proposed model and intermediate COCOMO model for 60 projects of the NASA dataset software projects [12] is shown.

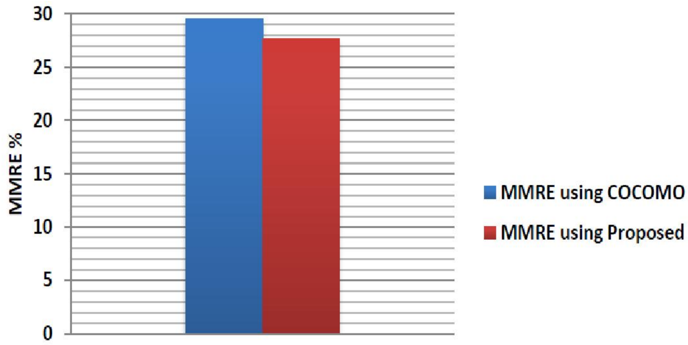

Fig. 6. The Comparison Diagram of Proposed and Intermediate COCOMO Model

7 Conclusion And Future Works

The accuracy of SCE causes the managers to schedule the projects of software development in a known format. In this paper, we have presented a new model for estimation of the costs of the software projects using a combination of GA and ACO for NASA software projects. In the proposed model, the effective factors in estimation are using the GA the test and using the ACO the training and better results are achieved in comparison to COCOMO model. By this paper, we hope in future will present better models for estimation of the SCE of implementation and designing with more accurate estimation by combining other meta-heuristic algorithms.

References

[1] K. Vinaykumar, V. Ravi, M. Carr, N. Rajkiran, “Software Development Cost Estimation using Wavelet Neural Networks”, *Journal of Systems and Software*, Vol. 81, No. 11, pp. 1853-1867, 2008.

[2] N. Mittas, L. Angelis, “Visual Comparison of Software Cost Estimation Models by Regression Error Characteristic Analysis”, *Journal of Systems and Software*, Vol. 83, pp. 621-637, 2010.

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

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

[5] F.S. Gharehchopogh, I. Maleki, B. Zebardast, “A New Solutions for Continuous Optimization Functions by using Bacterial Foraging Optimization and Particle Swarm Optimization Algorithms”, *Elixir International Journal Computer Science and Engineering (Elixir Comp. Sci. & Engg.),* Vol. 61, pp. 16655-16661, July 2013.

[6] F.S. Gharehchopogh, I. Maleki, B. Zebardast, “A New Approach for Solving N-Queens Problem with Combination of PMX and OX Crossover Operators”, *Elixir International Journal Computer Science and Engineering (Elixir Comp. Sci. & Engg.),* Vol. 61, pp. 16650-16654, July 2013.

[7] F.S. Gharehchopogh, I. Maleki, S.R. Khaze, "A New Optimization Method for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Particle Swarm Optimization", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET),* Vol. 2, Issue 2, pp. 352-358, February 2013.

[8] F.S. Gharehchopogh, I.Maleki, S.R. Khaze, “A New Approach in Dynamic Traveling Salesman Problem: A Hybrid of Ant Colony Optimization and Descending Gradients”, *International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT),* Vol. 3, No. 2, pp. 1-9, December 2012.

[9] I. Maleki, S.R. Khaze, F.S. Gharehchopogh, “A New Solution for Dynamic Travelling Salesman Problem with Hybrid Ant Colony Optimization Algorithm and Chaos Theory”, *International Journal of Advanced Research in Computer Science (IJARCS),* Vol. 3, No.7, pp. 39-44, Nov-Dec 2012.

[10] F.S. Gharehchopogh, I. Maleki, M. Farahmandian, "New Approach for Solving Dynamic Traveling Salesman Problem with Hybrid Genetic Algorithms and Ant Colony Optimization", *International Journal of Computer Applications (IJCA),* Vol. 53, No.1, pp. 39-44, September 2012.

[11] 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.

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

[13] C.F. Kemerer, "An Empirical Validation of Software Cost Estimation Models", *Communications of the ACM*, Vol. 30, No.5, pp. 416-429, 1987.

[14] A.F. Sheta, "Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects", *Journal of Computer Science*, Vol. 2, No. 2, pp.118-123, 2006.

[15] J. Hakkarainen, P. Laamanen, R. Rask, "Neural Networks in Specification Level Software Size Estimation," *Proceeding of the 26th Hawaii International Conference on*, Vol. 4, pp. 626-634, 1993.

[16] C. Mair, G. Kadoda, M. Lefley, K. Phalp, Ch. Schofield, M. Shepperd, S. Webster, "An Investigation of Machine Learning Based Prediction Systems", *Journal of systems and software*, Elsevier, Vol. 53, Issue 1, pp. 23-29, 2000.

[17] B.K. Singh, A.K. Misra, "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, December 2012.

[18] Ziauddin, Sh. K.Tipu, Kh. Zaman, Sh. Zia, "Software Cost Estimation Using Soft Computing Techniques", *Advances in Information Technology and Management (AITM),* Vol. 2, No. 1, pp. 233-238, 2012.

[19] A. Mittal, K. Parkash, H. Mittal, "Software Cost Estimation Using Fuzzy Logic", *ACM SIGSOFT Software Engineering*, Vol. 35, No. 1, pp. 1-7, November 2010.

[20] Prasad Reddy P.V.G.D, Hari CH.V.M.K, S. Rao.T, "Multi Objective Particle Swarm Optimization for Software Cost Estimation", *International Journal of Computer Applications*, Vol. 32, No.3, pp. 13-17, October 2011.

[21] Z.A. KHALIFELU, 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.

[22] Z.A. KHALIFELU, F.S. GHAREHCHOPOGH, "A New Approach in Software Cost Estimation Using Regression Based Classifier", *AWER Procedia Information Technology & Computer Science Journal*, Vol. 2, pp. 252-256, December 2012.

[23] Z.A. KHALIFELU, 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.

[24] C.S. Reddy, P.S. Rao, K. Raju, V.V. Kumari, "A New Approach for Estimating Software Effort Using RBFN Network", *IJCSNS International Journal of Computer Science and Network Security*, Vol. 8, No. 7, pp. 237-241, 2008.

[25] J. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan, Michigan, USA, 1975.

[26] 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.

[27] 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.