Volume 4, Issue 1, January 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

COCOMO model Coefficients Optimization Using GA and ACO

Vishali *
CSE &M.M U Sadopur Ambala
India

Anshu Sharma
CSE&M.M U Sadopur Ambala
India

Suchika Malik
ECE &M.M U Sadopur Ambala
India

Abstract— The COCOMO Model is well known as the currently predominate model for software cost estimation. It allows one to work from linguistic variables to estimate software project effort and schedule. This basis in linguistic variables encourages research of the COCOMO Model as a fuzzy system. As is known in fuzzy circles and is shown here, fuzzy arithmetic based on the popular fuzzy extension principle may produce unacceptable results under fuzzy multiplication. This makes fuzzy results of some computations too fuzzy to be useful. Nevertheless, in the case of software cost estimation using COCOMO, we find and show that this characteristic of fuzzy arithmetic may be used to advantage. To evaluate our calibration and to compare it with the original model, and with a calibration obtained with linear regression, we used our new objective function. The results show that our calibration algorithm performs better than linear regression and leads to a new version of the model that is better than the original one. Software effort estimation is one of the essential steps to be carried out in the project planning. The effective and efficient development of the software requires accurate estimates. Software researchers are providing many cost estimation methods for several decades. Among those methods, COCOMO is the most commonly used model because of its simplicity for estimating the effort in person-month for a project at the different stages. Today's effort estimation models are based on soft computing techniques as neural network, genetic algorithm, the fuzzy logic modeling, aco etc . for finding the accurate predictive software development effort and time estimation. Genetic Algorithm and ant colony optimization can offer some significant improvements in accuracy and has the potential to be a valid additional tool for software effort estimation.

Keywords— COCOMO model, cost estimation, genetic algorithm, ant colony optimization.

I. Introduction

This paper is concerned with a method of improving project outcome using Constructive Cost Models (COCOMO).
Software cost estimation is essential for software project management. Accurate software estimation can provide good support for the decision-making process like the accurate assessment of costs can help the organization to better analyse the project and effectively manage the software development process, thus significantly reducing the risk. Once the planning is too pessimistic, it may lose business opportunities, but too optimistic planning can cause significant loss.
Software cost estimation is process of predicting the effort required to develop a software engineering project. While the software cost estimation may be simple in concept, it is difficult and complex in reality. The major part of cost of software development is due to human-effort and most cost estimation methods focus on this aspect and give estimates in terms of person-month It determines the amount of effort necessary to complete a software project in terms of its scheduling, acquiring of resources, and meeting of budget requirements. The effective and efficient development of the software requires accurate estimates. Software researchers are providing many cost estimation techniques for several decades but the main problem persist in software engineering field. Early software estimation models are based on the regression analysis or mathematical derivations. Among those methods, COCOMO is the most commonly used model. Today‟s models are based on simulation, neural network, genetic algorithm, soft computing, the fuzzy logic modeling etc. In this paper, COCOMO model used the most frequently and widely used genetic algorithm and ant colony optimization approach for optimizing the current coefficients that estimate the optimized predictive effort required for the development of software project. Genetic algorithms and ant colony optimization are optimization algorithms in the evolutionary computing techniques and proposed in 1975 by a scientist Holland. It is a natural heuristic algorithm that is used to find exact and approximate solutions. Algorithm is based on iterative improvement of current solution, but a solution set is used instead of one solution.

II. COCOMO MODEL

A project manager needs to clearly identify the cost estimate of software development so that he/she can evaluate the project progress against expected budget, expected schedule and potentially improve resource utilization in. It was found that the main cost driver for software development is the effort, where effort is translated into cost. The primary element which affects the effort estimation is the developed kilo line of code (KLOC). The KLOC include all program instructions and formal statements. Many software cost estimation models where proposed to help in providing a high quality estimate to assist project manager in making accurate decision about their projects. A well known mathematical model for software cost estimation is the COCOMO model. COCOMO model was first provided by Boehm. This model was built based on 63 software projects. The model helps is defining mathematical equations that identify the developed time, the effort and the maintenance effort. COCOMO model is used to make estimates based upon three different software project estimates. The three ways of estimating software project effort/cost with increasing levels of accuracy are simple, intermediate and complex models. These three models are defined using increasingly detailed mathematical relationship between the developed time, the effort and the maintenance effort. The estimation accuracy is significantly improved when adopting models such as the Intermediate and Complex COCOMO models. The COCOMO model has the form given in Equation 1.

E = a(KLOC)b (1)
E presents the software effort computed in man-months. The values of the parameters a and b depend mainly on the class of software project. Software projects were classified based on the complexity of the project into three categories. They are:
- Organic
- Semidetached
- Embedded.

These models exhibit some nonlinearity characteristics. Extensions of COCOMO, such as COMCOMO II, can be found however, for the purpose of research reported, in this paper, the basic COMCOMO model is used. The three models are given in Table I. These models are expected to give different results according to the type of software projects (i.e. Organic, semi-detached and embedded).

Table1. Basic COCOMO Models

A. Model name	*B. Effort (E)*	*C. Time(D)*
D. Organic Model	*E.* $E = 2.4(KLOC)^{1.05}$	*F.* $D = 2.5(E)^{0.38}$
G. Semi-Detached Model	*H.* $E= 3.0(KLOC)^{1.12}$	*I.* $D = 2.5(E)^{0.35}$
J. Embedded Model	*K.* $E = 3.6(KLOC)^{1.20}$	*L.* $D = 2.5(E)^{0.32}$

III. GENETIC ALGORITHM BASED APPROACH FOR OPTIMIZE ESTIMATED EFFORT

Today's Software development effort estimation models are based on soft computing techniques as neural network, genetic algorithm, the fuzzy logic modeling ,ant colony optimization etc. for finding the accurate predictive software development effort and time estimation. As there is no clear guideline for designing neural networks approach and also fuzzy approach is hard to use. Genetic Algorithm and ant colony optimization can offer some significant improvements in accuracy and has the potential to be a valid additional tool for software effort estimation. It is a non-parametric method since it does not make any assumption about the distribution of the data and derives equations according only to the fitted values . Genetic Algorithm is one of the evolutionary methods for the effort estimation. The solution is achieved by means of a cycle of generations of candidate solutions that are pruned by criteria, survival of the fittest. Genetic Programming (GP) is a global search technique which makes it less likely to get stuck in the local optimum. This is different from other techniques such as neural networks and gradient descent which is prone to the local optimal values. It is particularly well suited for hard problems where little is known about the underlying search space and the concept is easy to understand .

Genetic Algorithm

Genetic algorithm is based on the next 4 main components:
1. *Chromosome* – the line of numbers that could be encoded using the binary encoding, integer number encoding etc. Each position in chromosome is called a bit, gene. Chromosome is an individual representing one of task solutions.
2. *Initial population*. The first population is a set of task solutions that is generated randomly. The main condition of the generation process of the first population is to achieve a variety of solution sets. If this condition is false – local extreme will be achieved early. It is not good for searching of the optimal solution.
3. *Operator set.* Operator set allows generating new solutions on the base of current population. Operator set contains selection, crossover and mutation. When selection is used, individuals are selected in the intermediate population. Different types of selection are known: Roulette wheel selection – each individual probability to be chosen in the

intermediate population is proportional with it fitness function value, it is called the proportional selection; Tournament selection – all individuals have an equal probability to be chosen in the intermediate population.
Respectively the crossover is chosen – the one-point crossover, two-point crossover, *n*-point crossover – individuals chosen to the intermediate population must make the exchange of chromosome parts. This process result is the generation of new individuals. The use of the mutation chromosome gene with defined probability exchanges its value. The new value of gene is also determined with defined probability. The mutation process protects population from the local extreme points, as well as enlarges the searching solution area.
4. *Fitness function*. The fitness function is the individual estimation attribute. It shows the suitability for each solution. On the one hand, the fitness function allows defining solutions that are more adapted – these solutions get a chance to be chosen in intermediate population. On the other hand, the fitness function allows defining solutions that are less adapted – these individuals are removed from the solution set. Therefore, the average fitness function value of new generation is larger than the average fitness function value of previous generation.

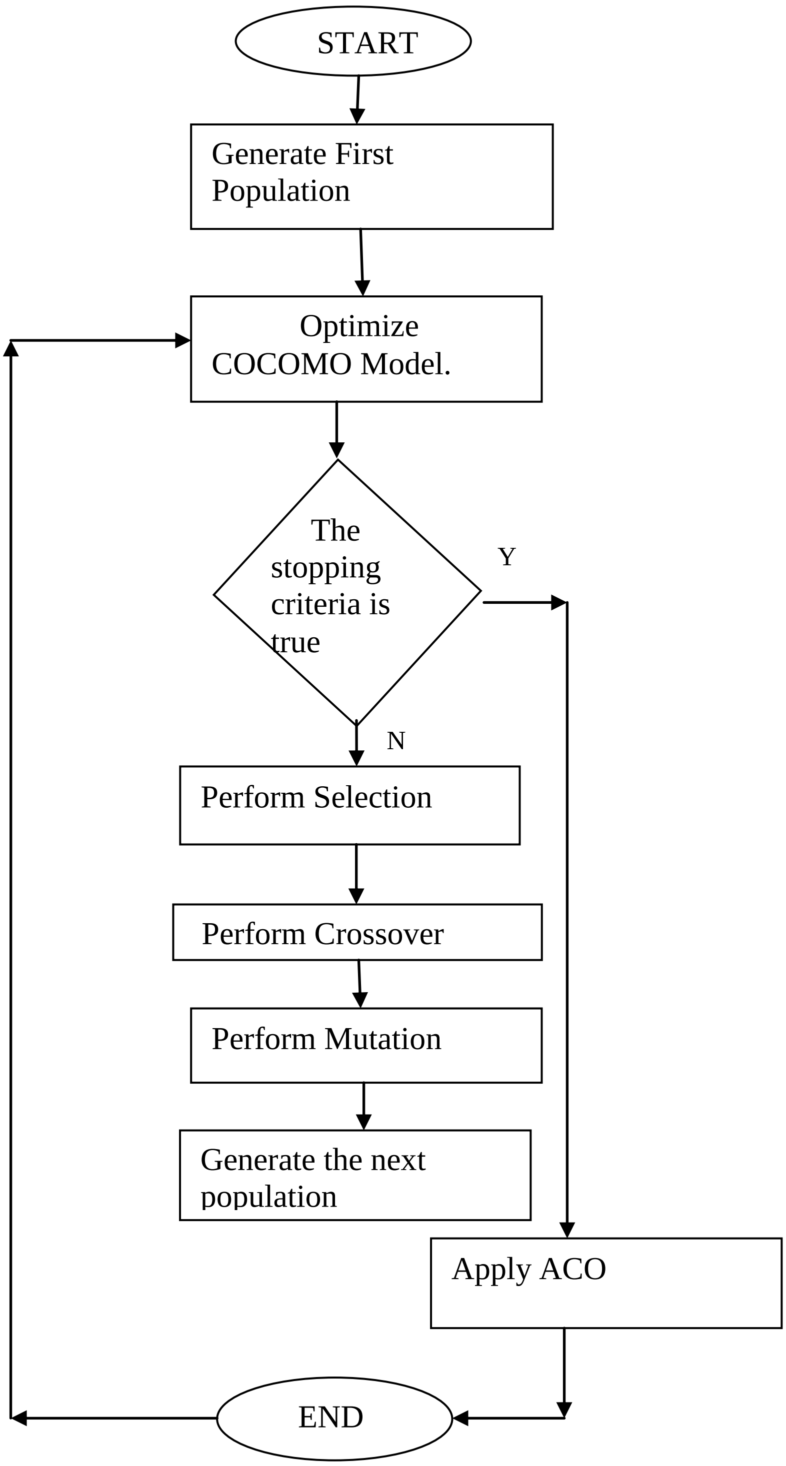

Figure 1. The flowchart representation of the Proposed System.

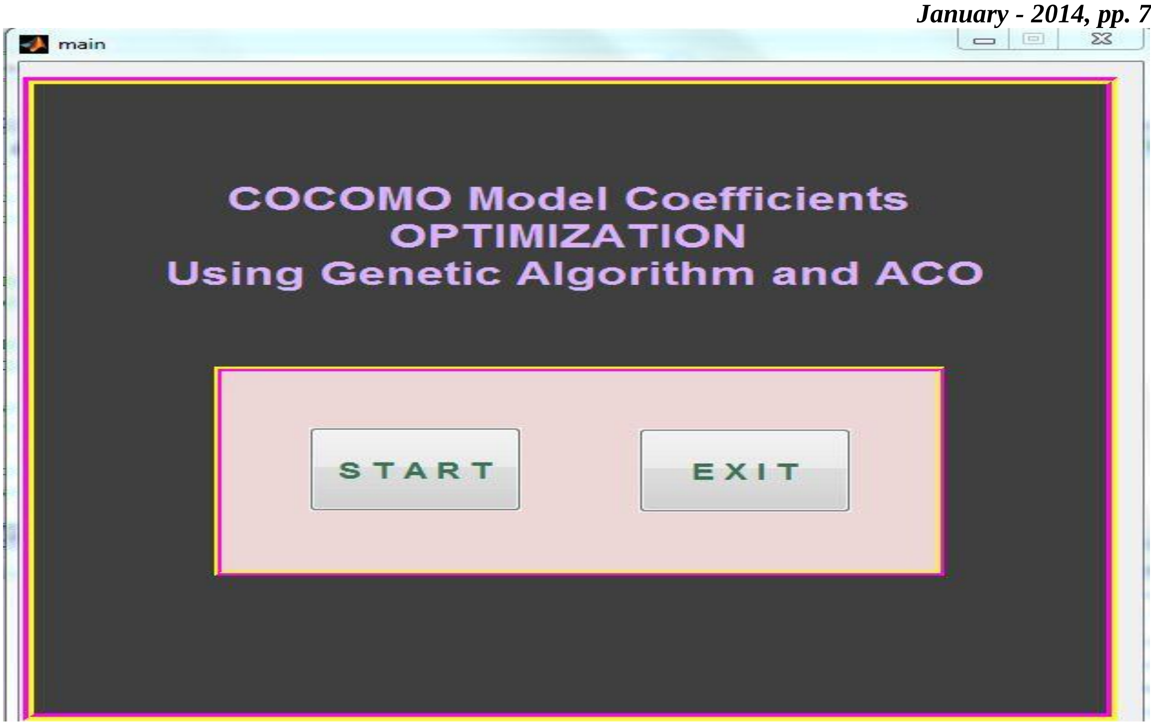

Figure 1. The basic layout of the proposed system.

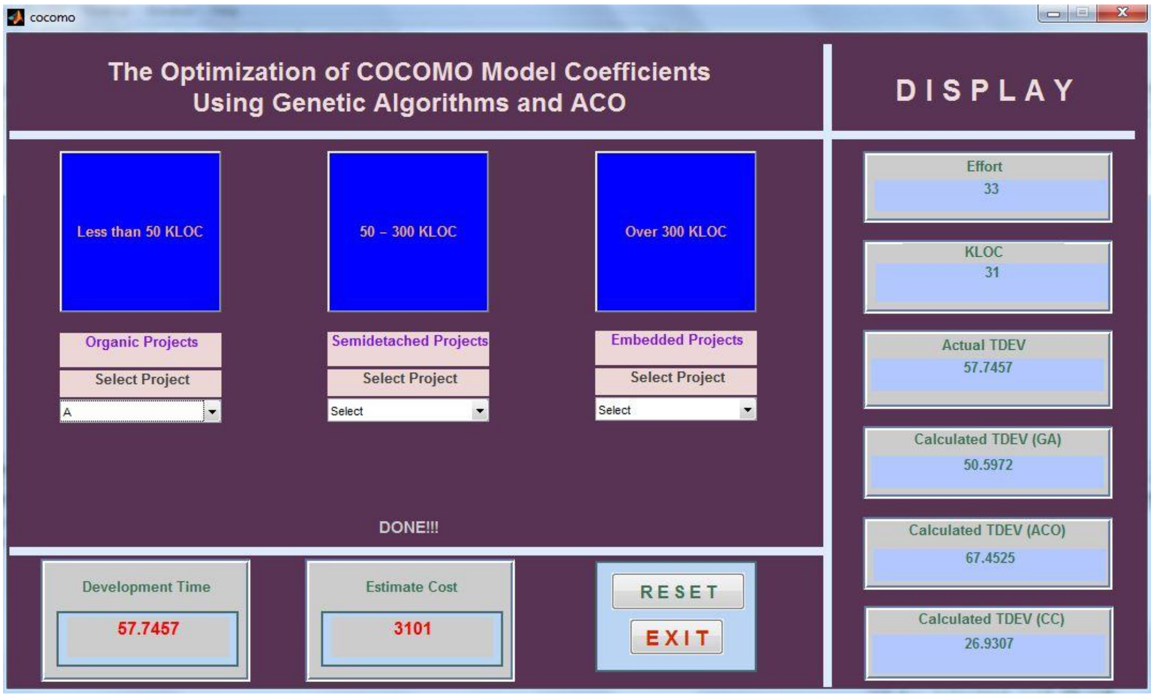

Figure 2: The calculation of the COCOMO Model of Sample 1.

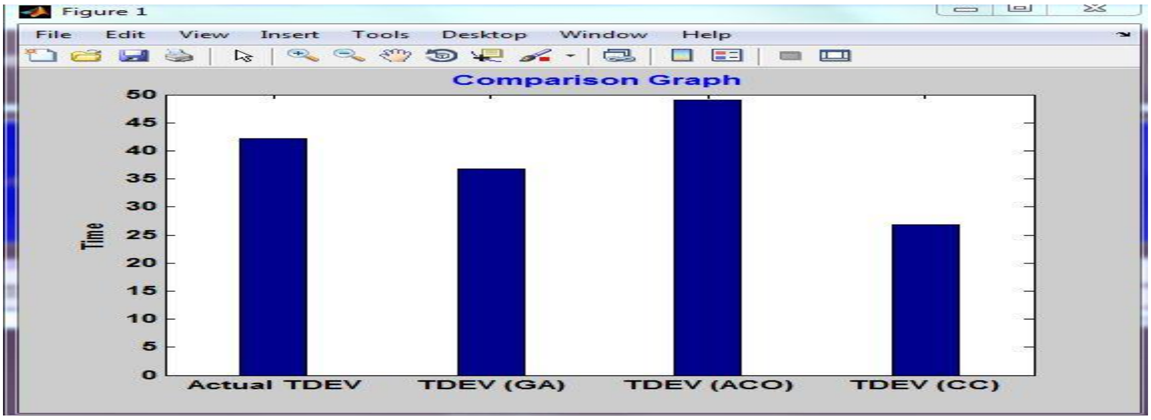

Figure 3: The graphical representation of the above fig.

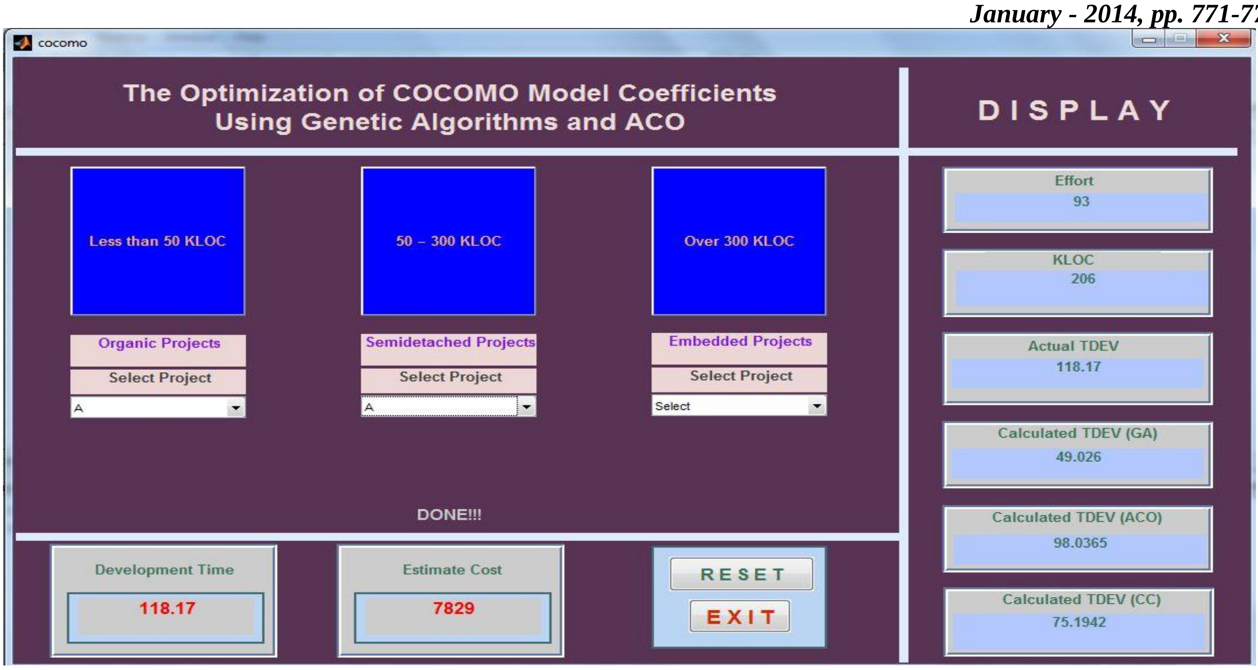

Figure 4: The calculation of the COCOMO Model of Sample 2.

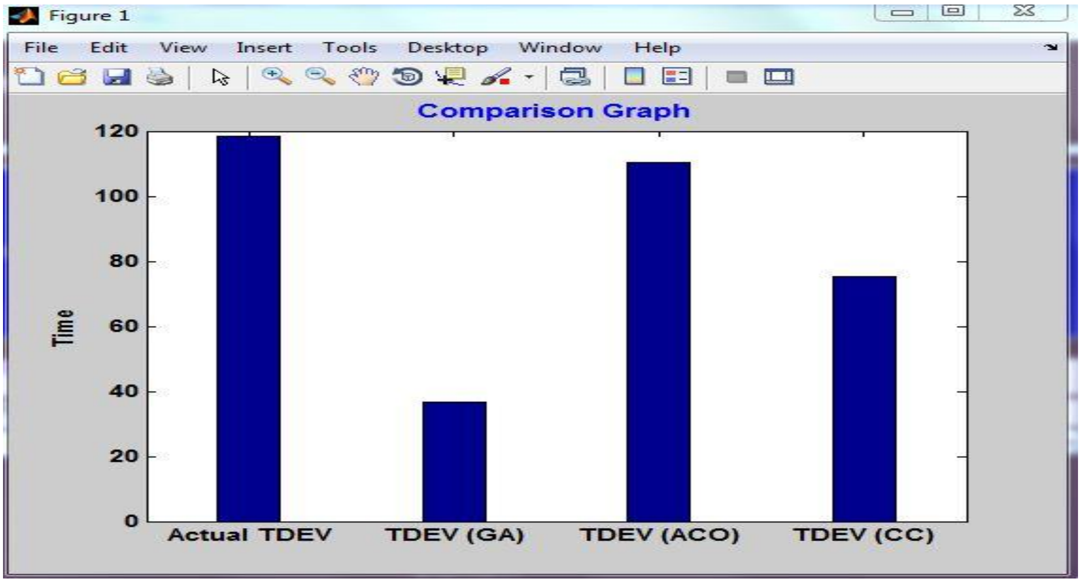

Figure 5: The graphical representation of the above fig.

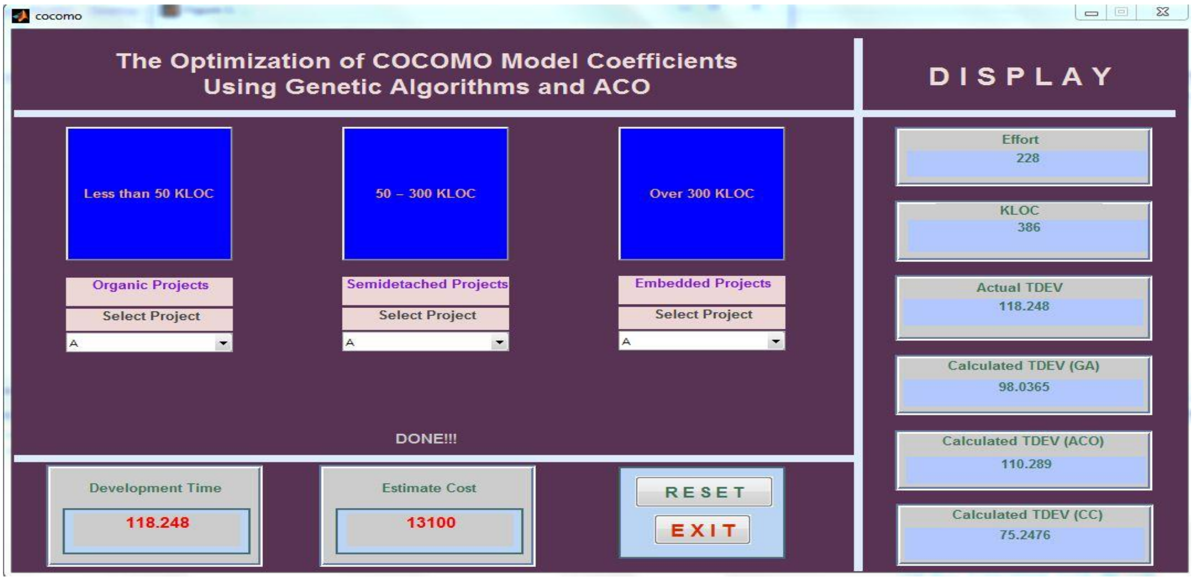

Figure 6: The calculation of the COCOMO Model of Sample 3.

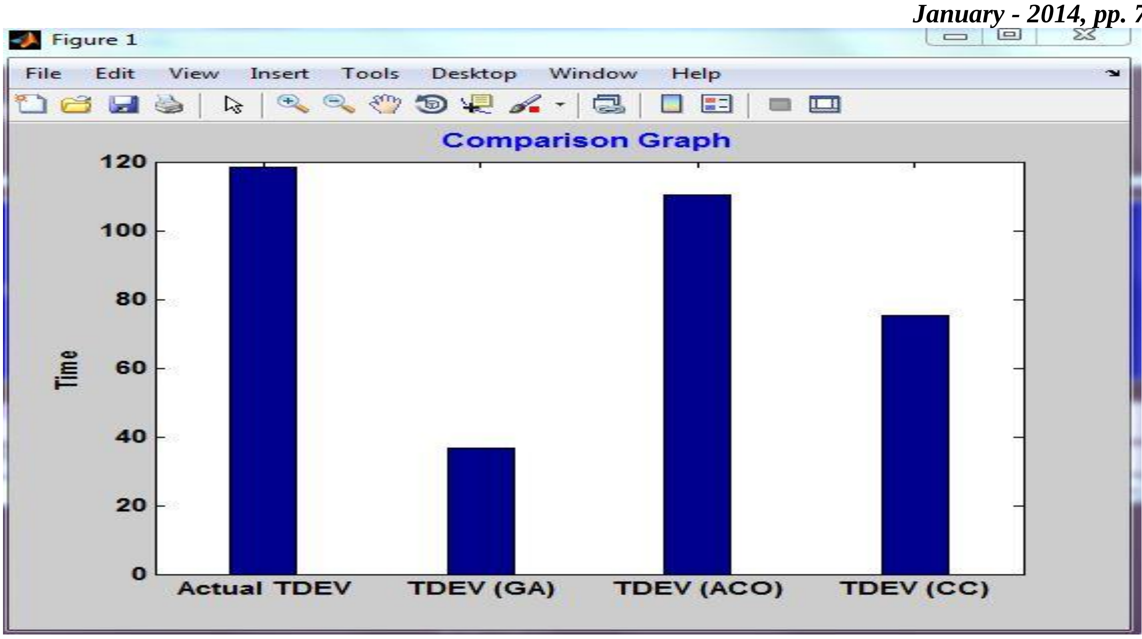

Figure 7: The graphical representation of the above fig.

IV. CONCLUSION

This research indicates directions for further research. The proposed framework can be analyzed in terms of feasibility and acceptance in the industry. Trying to improve the performance of existing methods and introducing the new methods for estimation based on today‟s software project requirements can be future works in this area. So the research is on the way to combine different techniques for calculating the best estimate. According to the findings of the research, it should be stated that having the appropriate statistical data describing the software development projects, genetic algorithms and ant colony optimization can be used to optimize the COCOMO model coefficients. The objective of this research was to optimize the COCOMO model coefficients using the genetic algorithms and ant colony optimization. The task of the COCOMO coefficient optimization is not new; different methods such as neural networks, fuzzy algorithms, object-oriented methods etc. were applied to it by a number of scientists.

The current research proposes a genetic algorithm and ant colony optimization based method for optimization of the COCOMO model coefficients both for organic and semi-detached modes. In a series of experiments, the proposed algorithm was tested and the obtained results were compared with the ones obtained using the current COCOMO model coefficients. The results show that in most cases the results obtained using the coefficients optimized by the proposed algorithm 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 and ACO in the organic mode produces better results in comparison with the results obtained using the current COCOMO model coefficients. At the same time, coefficients for the semi-detached mode produced by the proposed algorithm do not result in the high forecasting accuracy, which is the same or slightly worse than the accuracy obtained using the current COCOMO model coefficients. According to the findings of the research, it should be stated that having the appropriate statistical data describing the software development projects, genetic algorithms and ant colony optimization can be used to optimize the COCOMO model coefficients.

REFERENCES

[1] Maged A. Yahya, Rodina Ahmad, Sai Peck Lee, "Effects of Software Process Maturity on COCOMOII‟s Effort Estimation from CMMI Perspective" In 2008 IEEE, Department of Software Engineering, University of Malaya, pp. 255-256.

[2] Anna Galinina, Olga Burceva, Sergei Parshutin, "The Optimization of COCOMO Model Coefficients Using Genetic Algorithms" Riga Technical University, 2012/15, pp. 45-50.

[3] Kavita Choudhary, "GA Based Optimization of Software Development Effort Estimation" International Journal of Computer Science and Technology, Vol. 1, No. 1, Sep. 2010, ISSN: 0976-8491, pp. 38-40.

[4] Nancy Merlo–Schett, "Seminar on Software Cost Estimation" Requirements Engineering Research Group, Department of Computer Science, WS 2002/2003, pp. 3-19.

[5] Liming Wu, "The Comparison of the Software Cost Estimating Methods" University of Calgary, wul@cpsc.ucalgary.ca, pp. 2 -12.

[6] Chander Diwaker, Astha Dhiman, Sanjeev Dhawan, "Size and Effort Estimation Techniques for Software Development" International Association of Scientific Innovation and Research, Vol. 1 & 2, No. 4, ISSN: 2279-0071, pp. 35-37.

[7] Barry Boehm, Chris Abts, Sunita Chulani, "Software development cost estimation approaches –A survey" Annals of Software Engineering 10, 2000, pp. 182-190.

[8] R. Bhatnagar, V. Bhattacharjee, M.-K. Ghose, Software Development Effort Estimation – Neural Network Vs. Regression Modeling Approach, Vol. 2, International Journal of Engineering Science and Technology, 2010, pp. 2950-2956.

[9] R. Chandrasekaran and R.-V. Kumar, On the Estimation of the Software Effort and Schedule using Constructive Cost Model – II and Functional Point Analysis, Vol. 44, Department of Statistics, Tambaram: 2012, pp. 38-44.

[10] J. Horn, N. Nafpliotis and E. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization. Orlando, USA: IEEE; 1994, pp. 82-87.

[4] X. Huang, D. Ho, J. Ren, L.F. Capretz, Improving the COCOMO model using a neuro-fuzzy approach. Vol. 7, Applied Soft Computing, 2007, pp. 29-40.

[11] A. Kaushik and A. Chauhan, COCOMO Estimates Using Neural Networks, Vol. 9, Intelligent Systems and Applications, Delhi: Modern Education and Computer Science Press, 2012, pp. 22-28.

[12] Y. Miyazaki, K. Mori, COCOMO evaluation and tailoring. Proc. Eighth Int. Conf. Soft. Eng., London, UK, 1985, pp. 292-299.

