IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Optimization Estimation Parameters of COCOMO Model II Through Genetic Algorithm

Arfiha Khatoon[1*], Rupinder Kaur[2]

[1*] Dept. of CSE, Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India
[2] Dept. of CSE, Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India

**Corresponding Author: arfiha.raj@gmail.com, Tel.: 8051222876*

Available online at: www.ijcseonline.org

Accepted: 14/May/2018, Published: 31/May/2018

Abstract— Software cost estimation is very important in software project management.a major cause of failure of many software projects is the lack of accurate and early estimation. However, irrespective of great deal of importance estimating the time and development cost accurately is still a challenge in software industry. It is used to predict the effort and time need to complete the project. The need of optimization comes in various approaches like genetic algorithm of COCOMO MODEL II for providing better effort estimates and reliability.

Keywords—Genetic Algorithm, Optimization, Evolutionary Algorithms.

I. INTRODUCTION

Accurate software cost estimation has a great significance for both software development team and customers involved in the project. Estimating the effort, time plan and staffing levels required to develop a software project is referred as software cost estimation. It is important task in software project management .which will lead to reduce the risk. there are large number of parameter which affects the effort estimation and hence many technique to estimate it. The aim of our work is to propose a model that would provide optimum results. One of such model is the COCOMO MODEL(constructive cost model) it is a software cost estimation models to help project managers to make the right decision. COCOMO Model proposed by Barry Boehm in 1991 is generally utilized for assessing the exertion and improvement time using regression formula with parameters got from verifiable task information and normal for the present undertaking for evaluating the cost of software. This model is a high risk due to low accuracy and lack of reliability. This is where the need of optimization comes in various approaches like genetic algorithm have already been applied for tuning of the parameters of COCOMO in order to increase its accuracy and reliability. Accurate software cost estimation has a great significance for both software development team and customers involved in the project estimating the effort ,time plan and staffing levels required to develop a software project is refered as software cost estimation . The development of large scale software projects gain a growing intrest. Having the capacity to characterize the product measure the improvement length and the required offices turn out to be progressively a challenging task. The reason is that software requirement tools and techniques become more complex and the modern software is becoming more expansive to build and maintain software development management and quality goals are necessary the challenge of developing software system in a fast moving evolutionary algorithm scenario give rise to a number of demanding situation first situation is identifying software component is a crucial task in software development and the second one is to minimize number of test cases develop for the testing purpose.

One of the important and powerful algorithms is Genetic Algorithm (GA) it is a natural heuristic algorithm which makes it easy to search a large search space and exact the approximate solutions.GA is the most popular type of EA(Evolutionary Algorithm).it converts design space into genetic space and work with a coding variables.GA is based on the iterative improvement on a set of possible solutions to the problem. Most GA applications are linked to the development of prediction model and large scale information processing.the aim of this work is to genetic algorithm technique for the development of software assesment model for thr Nasa software project dataset.

COCOMO MODEL II:

COCOMO Model proposed by Barry Boehm in 1991 is generally utilized for assessing the exertion and improvement time using regression formula with parameters got from verifiable task information and normal for the present undertaking for evaluating the cost of software. COCOMO model have been broadly utilized for the computation of exertion .Effort figured by COCOMO show is estimated as far as size and constent value venture parameters a and b. These values of project parameters a,b depend on the class of software project. These values of project parameters a,b depend on the class of software project.it is classified into three categories. Firstly we defined organic model in this model Small project being developed by a small team and the second is semidetached model in this model Medium-scale projects being developed by a relatively small team. And the last one is embeded model Large project being developed by a large team requires many innovations. The model aides in characterizing numerical conditions that distinguish the cost calendar and nature of a product item. The COCOMO Model is presented by the equation.(1)

$$\text{Effort}= a(kloc)^{b} \qquad (1)$$

For the complex project using following formula

$$\text{Effort}= a*kloc^{b}*p \qquad (2)$$

Table 1: BASIC COCOMOMODELS [3]

Model Name	Effort (E)	Time (T)
Organic	E=2.4(KLOC)1.05	T = 2.5(E)0.38
Semi-Detached	E=3.0(KLOC)1.12	T = 2.5(E)0.35
Embedded	E= 3.6(KLOC)1.20	T = 2.5(E)0.32

COCOMO MODEL II provides three stage series of model for estimation of software projects. First Application composition model, Second Early design model and third Post architecture (PA) model.

(1) Application composition model : For soonest stages or spiral cycles [prototyping, and some other prototyping happening later in the life cycle].in this model includes prototyping to determine potential high-chance issues, for example, UIs, programming/framework interaction, technology development, or execution. The costs of this type of effort are best estimated by the Applications Composition model.

(2) Early design model : For next stages or spiral cycles includes investigation of compositional or incremental improvement strategies. level of detail reliable with level of data accessible and the general level of estimation precision required at this stage. In this model includes investigation of elective software/framework structures and ideas of task. At this stage, insufficient is by and large known to help fine-grain cost estimation.

(3) Post architecture (PA) model : Once the task is prepared to create and manage a handled framework it ought to have an life cycle engineering, which gives more exact data on cost driver inputs, and empowers more precise cost estimates. In the accompanying segments the early plan and post design models are displayed. This model includes the real advancement and upkeep of a product item. This stage continues most cost-viably if a product life-cycle engineering has been created; approved concerning the framework's main goal, idea of activity, and chance; and built up as the system for the item.

It is reasoned that the COCOMO suite offers a capable instrument to predict software costs shockingly not the majority of the augmentations are as of now adjusted and therefore still experimental despite this disadvantage the projects. It underpins process change analyses, tool purchases, architecture changes part make/purchase tradeoffs and basic leadership process with sound results. Many tries were done to measure up to the adjustments in software life cycles, technologies, notations and hierarchical cultures since the principal form of COCOMO. In current study most of the present estimation techniques have been illustrated systematically. Since sofware venture directors are utilized to choose the best estimation technique in view of the conditions and status of the undertaking, portraying and including estimation systems can be valuable for decerasing the project disappointments. There is no estimation strategy which can be available the best estimation in every single different circumstance and every system can be reasonable in the extraordinary undertaking. It is important understanding the principals of every estimation technique to pick the best. Because performance of each estimation method depends on several parameters such as complexity of the project , duration of the project, expertise of the staff, development method and so on. Some evaluation metrics and an actual estimation example have been presented in this paper just for describing the performance of an estimation method (for example COCOMO). Trying to improve the performance of the existing methods and introducing the new methods for estimation based on today's software project requirements can be the future works in this area.

Importance of software cost estimation

Software cost estimating has been developing in significance till today. When the computer era began, very few computers were in use and most of the applications were small. As time moved on, computers became widespread the applications is use grew in number, size and importance and along with this costs to develop software grew as well. As a result of the growth, the consequences of errors in software cost estimation became more vulnerable. Indeed, even today, a great deal of cost estimates of software projects are not exceptionally exact truth be told, the greater part of them are too low. This isn't an surprising that we need to confront different difficulties while evaluating software costs. There are few costs which are not at all hard to determine and can be estimated in advance and are even fixed sometimes as the hardware or software requirement purchase or the license costs . But also there exists cost which are not easy to be estimated. The by far greatest amount of the total costs of a project arises from the salaries of the personnel. The costs for the human workers are highly correlated to the effort we need to perform the project. Therefore, it becomes necessary to get an accurate enough estimate of the total effort in order to make more precise estimate of the costs. Size and complexity are the basis of estimating effort for the project and both of these are derived from the specification Because the requirements of the software are likely to change at any given instant, we have to consider it into account too when estimating the effort. The difference in productivity of software developers is a major issue to solve during the estimation process. An experienced developer will have far more productivity than a beginner. But, because each project is unique and uses its own tools and languages, the experience level of the development team is hard to judge. Another issue shows up when people are assessing. Sometimes, unknowingly we tend to underestimate immaterial things like software which later becomes a problem

in the later development phase of the software. Today's world would not be the same if there was no software.

Genetic Algorithm Methodology : In the 1960s, after Charles Darwin introduced the concept of automated problem solving three ideologies or interpretations took birth simultaneously at three different locations. The use of Darwinian principles for automated problem solving originated in the 1950s. Lawrence J. Fogel introduced Evolutionary Programming in the US, while John Henry Holland called his method a genetic algorithm. Genetic Algorithm (GA) it is a natural heuristic algorithm which makes it easy to search a large search space and exact the approximate solutions. GA is the most popular type of EA(Evolutionary Algorithm).it converts design space into genetic space and work with a coding variables.GA is based on the iterative improvement on a set of possible solutions to the problem. Most GA applications are linked to the development of prediction model and large scale information processing. The aim of this work is to genetic algorithm technique for the development of software assesment model for the Nasa software project dataset.

Genetic algorithm is based on 4 main components:

1. Chromosome : The line of numbers that could be encoded using the binary encoding, whole number encoding and so on. Each situation in chromosome is known as a bit, gene. Chromosome is an individual representing to one of task solutions.

2. Initial population: The primary population is an arrangement of assignment arrangements that is produce randomly. The fundamental state of the generation process of the main population is to accomplish a variety of solution sets. if this condition is false – local extreme will be accomplished early. It isn't useful for searching of the optimal solution.

3. Operator set: operator set permits creating new arrangements on the base of current population. It contains selection, crossover and mutation delivering new arrangements on the base of current population. Fig. 1 shows the essential genetic algorithm.

A. Selection: In selection, people are chosen in the intermediate population. At the point when selection is used, people are chosen in the intermediate population. Different types of selection are known: Roulette wheel selection– every individual probability to be picked in the intermediate population is corresponding with it fitness function value , it is known as the relative choice; Tournament choice all people have an equivalent probability to be picked in the intermediate population.

B. Crossover : crossover is connected on a person by essentially exchanging one of its hubs with different hubs from another person in the population. With a tree-based representation, replacing a hubmeans repalcing the entire branch. This adds more effectiveness to the crossover operator. The expressions resulting from crossover are very different from their initial parents.

C. Mutation : Influences a person in the population. It can repalce an entire hub in the chose individual, or it can replace only the hub's data. With a specific end goal to look after honesty, tasks must be safeguard or the kind of data the hub holds must be considered.

4. Fitness function: It is the individual estimation characteristic. An objective function is used that choose the fitness of an individual . The fitness function is the individual estimation quality. It demonstrates the appropriateness for every arrangement. the fitness function permits characterizing arrangements that are more adjusted –

these arrangements persuade an opportunity to be picked in intermediate population. Then again, the fitness function permits characterizing arrangements that are less adjusted – these individual are expelled from the arrangement set. Hence, the average fitness function estimation of new generation is bigger than the average fitness function estimation of past generation.

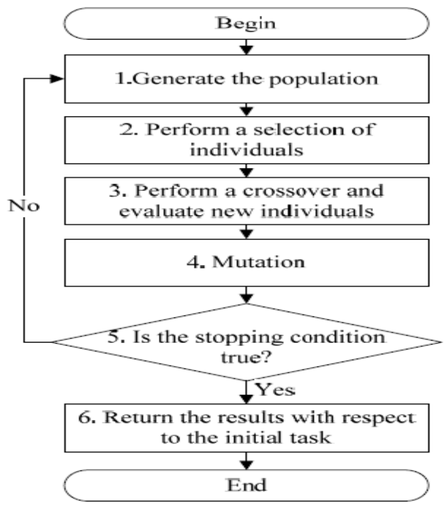

Fig. 1.The steps of Basic genetic algorithm [10]

Evolutionary algorithms (EAs) and its process:
Evolutionary algorithms (EAs) are a wide class of stochastic enhancement calculations, motivated by science and specifically by those natural procedures that enable population of living beings to adjust to their surrounding environment: genetic inheritance and survival of the fittest. Charles Darwin initially presented the idea in the nineteenth century is still today generally recognized as valid. These calculations receive Darwinian standards, and have a place with a group of experimentation issue solvers. These algorithms adopt Darwinian principles, and belong to a family of trial and error problem solvers. These can be considered as global optimization methods and the inspiration to these algorithms is biological mechanisms such as reproduction, mutation, recombination, selection. Recombination and mutation create the necessary diversity and thereby facilitating novelty, while selection acts as a force for increasing the quality. Candidate solutions to the optimization issue assume the part of individuals in a population, and the fitness function decides the quality of the solutions. Evolution of the population at that point happens after the repeated application of the above operators. Artificial evolution (AE) depicts a procedure including individual evolutionary algorithms; EAs independently take an interest as components in artificial evolution . The disallowing factor for EAs in real applications is the computation complexity which emerges because of estimation of the fitness function. Regardless of this, evolutionary algorithms can be found in fields diverse as engineering, , science, financial matters, genetics, tasks research,robotics.

The evolutionary process of GAs begins by the calculation of the fitness of each individual in the initial population. While halting standard isn't yet achieved we do the accompanying;

- Select individual for multiplication utilizing some choice components (i.e. tournament , rank, and so on.).
- Make a posterity utilizing crossover and mutation operators. The probability of crossover and mutation are chosen in based of the application.
- Figure the new generation. This procedure will end either when the ideal arrangement is found or the most extreme number of generation is reached.

Genetic Programming: Genetic Programming is another significant subclass of evolutionary algorithms that discover computer programs which are equipped for solving complex optimization issues and takes motivation from natural advancement. It is a particular genetic algorithm in which every individual is considered as a computer program. It comprises of a set of directions and how well a computer perform its task is estimated by a fitness function. At the point when GA is utilized for the determination of genuine issues, a population contained random set of individuals is created. The population is assessed during the evolution process. For every individual a rating is given, reflecting the level of adjustment of the individual to the environment. A level of the most adjusted individual is kept, while that the others are disposed of. The individual kept in the choice procedure can endure changes in their essential attributes through a component of reproduction. This system is applied on the current population planning to investigate the search space and to discover better answers for the issue by methods for crossover and mutation operatorsc reating new people for the people to come. An effort based model is proposed for estimation of COCOMO demonstrate utilizing genetic algorithm. The algorithm considers strategy straightly related to effort. The model estimate the estimation of parameters of COCOMO display. The execution of created show is tried on NASA software projects data. The created demonstrate is discovered compelling in precise exertion estimation. A technique has been proposed for highlight determination and parameters optimization for machine learning regression for software exertion estimation. Reproductions are finished using benchmarkdata sets of software projects, in particular, Desharnais , NASA, COCOMO. The created display was tested for NASA software project data. The table 2 demonstrates correlation of estimated exertion and evaluated exertion utilizing genetic algorithms From the table, plainly the created show can give great estimation capacities.

Sr..No.	Project No.	MeasuredEffort	GA's Estimated Effort
1	1.	115.8000	131.9154
2	2.	96.0000	80.8827
3	3.	. 79.0000	81.2663
4	4.	90.8000	91.2677
5	5.	39.6000	60.5603
6	6.	98.4000	106.7196
7	7.	18.9000	31.6447
8	8.	10.3000	27.3785
9	9.	28.5000	46.2352
10	10.	7.0000	11.2212
11	11.	9.0000	14.0108
12	12.	7.3000	22.0305
13	13.	5.0000	8.4406
14	14.	8.4000	15.9157
15	15.	98.7000	119.2850
16	16.	15.6000	25.8372
17	17.	.23.9000	31.1008
18	18.	138.3000	143.0788

Table: 1. Estimated values for Genetic Programming [12]

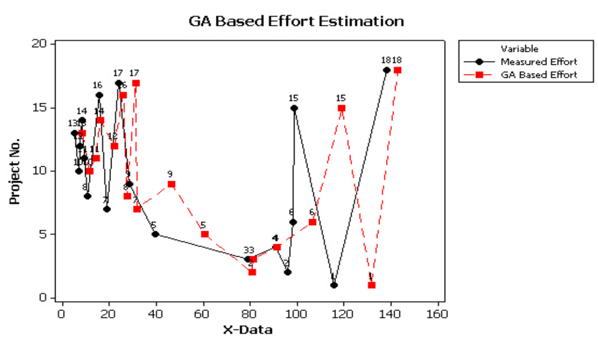

Fig: 2. GA Based Effort Estimation with Measured effort

This is visible from Fig.2 that genetic programming based exertion demonstrate gives comes about which are more strong and accurate. The solution gave by Genetic programming is more ideal and worldwide in nature. It can deliver a advanced mathematical function that the computed exertion is more exact.

II. Conclusion and Future Scope

In order to fulfill the requirements of IT industry it is necessary estimating the software cost in terms of Effort, Development Time and man power required in the early development phase becomes more important. Many approaches are used to estimate the software costs; out of which one of the widely used formularize approach is the Constructive Cost Model (COCOMO). But, this model also faces some issues and cannot provide the best estimates in most of the cases. It is a powerful instrument to predict software costs. It additionally supports process change investigations, engineering changes, and basic leadership process with credible outcomes. The performance of the created models were tested on NASA software project information introduced in . The created models could give great estimation abilities. We propose the utilization of Genetic Programming (GP) procedure to fabricate appropriate model structure for the software exertion. This survey indicates directions for further research. Trying to enhance the performance of existing methods and familiarize 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 integrate different techniques or methods for evaluating the best estimate.

III. References

[1] Boehm, B., 1995. Cost Models for Future Software Life Cycle Process: COCOMO2Annals of Software Engineering.

[2] S. Bhatia, A. Bawa, and V. K. Attri, "A Review on Genetic Algorithm to deal with Optimization of Parameters of Constructive Cost Model,"International Journal of Advanced Research in Computer and Commu-nication Engineering, vol. 4, no. 4, April 2015.

[3] S. A. Alaa F. Sheta, "Software effort estimation inspired by cocomo and fp models: A fuzzy logic approach," International Journal of Advanced Computer Science and Applications, vol. 4, no. 11, pp. 192–197, 2013.

[4] Barry Boehm. Software Engineering Economics. Englewood Cliffs, NJ:PrenticeHall,1981. ISBN 0-13-8221227

[5] B. K. Singh and A. K. Misra, "Software Effort Estimation by GeneticAlgorithm Tuned Parameters of Modified Constructive Cost Model for NASA Software Projects," International Journal of Computer Applica-tions, vol. 59, no. 9, pp. 22–26, Dec 2012.

[6] Bailey, J. W. and V. R. Basili, 1981. A meta model for software development resource expenditure. Proc. Intl. Conf. Software Engineering, pp: 107-115.

[7] K. Tang, K. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," Signal Processing Magazine, IEEE, vol. 13, no. 6, pp.22–37, 1996.

[8] Ekrem Kocaguneli, Tim Menzies, and Jacky Keung "On the Value of Ensemble Effort Estimation", Journal of IEEE Transactions on Software Engineering, Vol. X, 2012.

[9] X. Huang, J. Ren and L.F. Capretz, (2004), "A Neuro- Fuzzy. Tool for Software Estimation", Proceedings of the 20th IEEE International Conference on Software Maintenance, pp. 520.

[10] Anna Galinina1, Olga Burceva2, Sergei Parshutin3, 1-3Riga Technical University "The Optimization of COCOMO Model CoefficientsUsing Genetic Algorithms"2015.

[11] Sheta. A. and K. DeJong, 1996. Parameter estimation of nonlinear systems in noisy environment using genetic algorithms. Proc. IEEE Intl. Symp. Intelligent Control (ISIC'96), pp: 360-366.

[12] Alaa Sheta., (2006), "Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects", Journal of Computer Science 2 (2): 118 123.

[13] Dolado, J.J., 2001. On the problem of the software cost function. Information and Software Technology, 43: 61-72.

[14] Houck, C., J. Joines and M. Kay, 1996. A genetic algorithm for function optimization: A Matlab implementation. ACM Transactionson Mathmatical Software.

[15] Dr. Devesh Kumar Srivastava "Measure Performance of VRS Model using Simulation Approach by Comparing COCOMO Intermediate Model in Software Engineering" International Journal of Computer Science & Engineering Technology (IJCSET)2013.

[16] Kristinsson. K. and G. Dumont, 1992. System identification and control using geneti algorithms. IEEE Transaction on Systems, Man and Cybernetics, 22: 1022-1046.

[17]. Fonseca, C., E. Mendes, Fleming and S.A. Billings, 1993. Nonlinear model term selection with genetic algorithms. Proc. IEE/IEEE Workshop on Natural Algorithms in Signal Process., pp: 27/1 –27/8.

[18] L. H. Putnam, (1987), "A general empirical solution to the macrosoftware sizing and estimating problem". IEEE Transactions on Software Engineering, SE-4(4) pp 345-361.

[19] A. Idri, T. M. Khoshgoftaar, A. Abran, (2002), "Can neural networks be easily interpreted in software cost estimation?", IEEE Trans. Software Engineering, Vol. 2, pp.1162 –1167.

[20] Boraso, M., C. Montangero and H. Sedehi, 1996. Software cost estimation: An experimental study of model performances. Technical Report TR-96-22, Departimento Di Informatatica, Universita Di Pisa, Italy.

[21] Belkacem Mahdad, Tarek Bouktir and Kamel Srairi. (2008).Optimal power Flow of the Algerian Network using Genetic Algorithms/Fuzzy Rules, Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, pp. 1-8.

[22] Bhim Singh, B P Singh and (Ms) K Jain. (2003). Implementation of DSP Based Digital Speed Controller for Permanent Magnet Brushless dc Motor, IE(I) Journal-EL, 84, pp.16-21.

[22] B Mrozek, Z Mrozek. (2000). Modeling and Fuzzy Control of DC Drive, in Proceedings of 14th European Simulation Multi conference, pp 186-190.

[23] Chu-Kuei Tu and Tseng-Hsien Lin. (2000). Applying Genetic Algorithms On Fuzzy Logic System For Underwater Acoustic Signal Recognition, Proceedings of the 2000 International Symposium on Underwater Technology, pp. 405-410.

[24] Ching-Hung Wang, Tzung-Pei Hong, and Shian-Shyong Tseng. (1998). Integrating Fuzzy Knowledge by Genetic Algorithms, IEEE Transactions On Evolutionary Computation,2(4), pp.138-149.

[25] P. K. Nandam, and P. C. Sen. (1986). A comparative study of oportional-integral (P-I) and integral-proportional (I-P) controllers for dc motor drives, Int. Jour. Of Control,44, pp. 283-297.

[26] Raj Subbu, Arthur C. Sanderson and Piero P. Bonissone. (1998). Fuzzy Logic Controlled Genetic Algorithms versus Tuned Genetic Algorithms: An Agile Manufacturing Application, Proceedings of the Intelligent Control (ISIC), IEEE Joint Conference, pp.434-440.

[27] S. H. Yakhchali and S. H. Ghodsypour. (2008). A Hybrid Genetic Algorithms for Computing the Float of an Activity in Networks withImprecise Durations, Proceedings of the IEEE International Conference on Fuzzy Systems, pp.1789-1794.

Authors Profile

Ms. Arfiha Khatoon completed her Bachelor of technology in Computer Science and Engineering in 2017 and now pursuing her Mater of technology in Department of computer science and Electronics Engineering from Jayoti VidyapeethWomen'sUniversity,Jaipur (Rajasthan), India since 2017. She is an active member of Springer and IEEE journals since 2017 and published her 3 research papers in the very reputed journals and it is also available online. Her fundamental research work on optimization Optimization Estimation Parameters of COCOMO Model II Through Genetic Algorithm.She is also exploring her knowledge in another fields including Data Mining,Datawarehouse,Software Engineering,Cloud Computing and Cryptoraphy . She has 2 years of experience in Research field as a Research Scholar Student.

Ms. Rupinder Kaur completed her Bachelor in technology and Master in technology in Computer Science and Enginnering as integrated Degree from Jayoti Vidyapeeth Women's University Jaipur, India in the year of 2015.She is currently working as Assistant Professor in Department of Computer Science and Electronic Engineering in Jayoti Vdyapeeth Women's University Jaipur,India since 2017.She has published 7 International Paper. Her main research work focus on Network Security, Image Processing of recognition, Compiler Desiging Algorithm, Cryptography, Real Time System and Software Engineering. She has two years of teaching experience and three years of research Experience.

