

STUDY OF MODIFIED GENETIC ALGORITHM-SIMULATED ANNEALING FOR THE ESTIMATION OF SOFTWARE EFFORT AND COST

I.Thamarai.¹, S. Murugavalli.²

¹Research Scholar, Sathyabama University, Chennai, India,

²Supervisor, Sathyabama University, Chennai.

Email: ¹ilango.thamarai@gmail.com, ²murugavalli26@rediffmail.com

Abstract

Software Effort Estimation is very crucial estimation task because of the intangible nature of software but it is very essential for developing a software application. Numerous approaches are available in software estimation method such as non-algorithmic models and algorithmic models. The main problems of SE prediction are identification of components of software project and feature selection between the projects. In this paper Modified Genetic Algorithm-Simulated Annealing (MGASA) is proposed to predict the software effort and cost estimation. In MGASA, the estimation of software is based on similar projects. Modified Genetic Algorithm (MGA) is a stochastic search algorithm. The adaptive search process has been effectively utilized in different types of challenging numerical optimization problems. The Simulated Annealing (SA) is an analogy between the way, in which the metal freezes and cools into a minimum energy crystalline structure. The main advantage of using Modified Genetic Algorithm-Simulated Annealing (MGASA) tool is to enhance the accuracy of software effort estimation.

Key words: Software Estimation (SE), Modified Genetic Algorithm-Simulated Annealing (MGASA), Similarity Measure, Effort Estimation, Cost estimation.

I. INTRODUCTION

Software effort estimation is an important step in the project planning. The software parameters should be considered to estimate the software effort and cost. One of the most significant parameter is the project size which should be monitored carefully. The software effort estimation and cost depends on the accurate forecasting of the software size (1). Typically, the software effort and cost estimations are very challenging process in the software project. The software projects are repeatedly not unique and there may be no previous or background knowledge about them. The software estimation models which predict the software development cost, effort and schedule of the project modules more quickly and easily are very much essential for the growth of software projects.

Even though a great amount of software research time and money have been spend to improve the accuracy of software effort estimation using different types of estimation models, due to the integral uncertainty in software development projects for example dynamic and complex interaction factors, pressure on

standardization, lack of software data and intrinsic software complexity, it is unrealistic to anticipate very accurate effort estimation of a software (2) Dejaeger et al addressed this issue using large scale benchmarking study. Different types of methods are used in this study such as tree or rule based model, linear model, nonlinear model, linear regression and estimation techniques. Furthermore, the rigorous statistical testing is used for finding the subjected results and combination of least squares regression and the logarithmic transformation is performed as the best predictive attributes. Predictive attributes are the development of project, project size and environment related attributes. (3).The software project accuracy can be definite based on understanding the standardization of the software data. Since the reliability and accuracy of the software effort estimation is very significant for the software companies to gain attention, the researchers and enterprises have put their determined effort to develop the accurate software effort estimation model to anticipate the effort close to the accurate levels. In (4), Ning Nan and Donald E. Harter emphasize the role of budget and schedule in effort calculation. Magne Jorgenson et al suggest the idea of

receiving feedback about the software professional's performance in estimation and its consequences (5). Tim Menzies et al evaluated multiple projects and concluded that the estimation should be based on clusters from available data (6). The principles of software effort estimation were explored by Ekrem Kocaguneli (7). An application based on search based optimization was proposed by Mark Harman and Ashin Massouri (8). They claimed that the main advantages of using optimization method are scalability and robustness.

Numerous estimation methods have been proposed and can be classified depending on their simple formulation schemes such as Estimation by Expert, Algorithmic Methods including Empirical methods, Analogy based estimation schemes, Artificial Neural Network based approaches, Rule Induction methods, Decision Tree based methods, Bayesian Network approaches and Fuzzy Logic based estimation schemes. This paper describes the importance and utilization of the software effort and cost estimation method and their effort on the software project. The main advantage of using Modified Genetic Algorithm-Simulated Annealing (MGASA) tool is to enhance the accuracy of software effort estimation. The rest of this paper is structured as follows. Section 2 describes the related work in this context. Modified Genetic Algorithm-Simulated Annealing (MGASA) is described in Section 3, Section 4 discusses the Software Effort Estimation based on MGASA and Conclusion is given in Section 5.

II. RELATED WORK

A predictive model is needed to be accurate in order to gain confidence in a business setting. Various types of software Cost Estimation methods are analyzed and discussed in (9) and (10). Magne Jorgenson provided many guidelines based on evidences to improve the uncertainty in cost assessments and effort estimation (11). He also provided a detailed review on expert estimation methods of software effort estimation methods in (12).

Tim Menzies et al proposed COSEEKMO model in (13) which is the extended version of COCOMO model to avoid the deviation problem that arises because of comparing two or more effort estimation methods. In (14), Pichai Jodpimai et al discussed characteristics of

various measures among interrelationship such as software project size, ranging from project model which is utilized for administration of the project. Underlying model is used for analysis of the different types of measures in various dimensions. For measuring characteristics of project technically, metrics are used in certain models and metrics are not suitable for very small project because of insufficient information. The project manager should give proper attention of the selection of project parameters for accurate estimations.

Attarzadeh et al proposed a novel artificial neural network (ANN) prediction model (15). It combines with Constructive Cost Model (COCOMO) model as it provides better estimation results at early phase of software development. The ANN model utilized different types of attributes of past project data. This proposed approach uses 156 set of project data from NASA93 and COCOMO I. The proposed method acquired results that show 8.36% of estimation accuracy from ANN, when compared with original COCOMO II model. In (16), Sathyananda Reddy et al proposes Artificial Neural Networks based software effort estimation method. This method will be improving the neural network performance that ensembles to the COCOMO Model. Single layer Feed Forward Neural Network is proposed. The neural network evolves with back propagation learning algorithm. The iterative process is done with Resilient Back Propagation algorithm. COCOMO dataset is utilized for training and testing process. The training and testing results are compared with the Single Layer Artificial Neural Network (SLANN) and SLANN with Back Propagation (BP). The proposed method shows promising results compared to other two approaches.

In software development process, confined budget and time to develop a project is very important activity. Therefore the accuracy of effort estimation plays a very significant role in software system. So the author proposes a new method for predicting time and cost of software project where two algorithms are used for this prediction namely Fuzzy Logic and Genetic Algorithm (17). Different types of parameters are used for this prediction such as memory utilized, accuracy, search time, error bit rate and build time. This approach helps to estimate the software effort efficiently. The Software Development Framework (SDF) is used to implement the Capability Maturity Model Integrated (CMMI) for

improving the effort estimation. The Multiple Linear Regression (MLR) may be used for testing and development process (18).

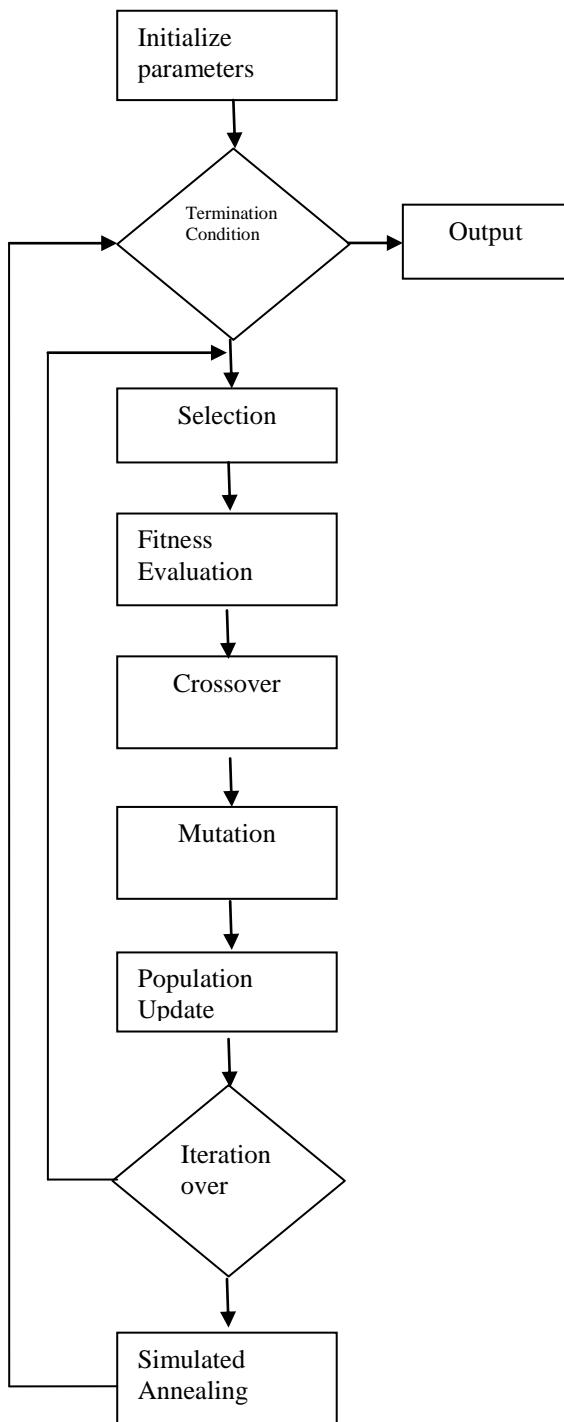


Fig. 1. Modified Genetic Algorithm-Simulated Annealing (MGASA)

III. MODIFIED GENETIC ALGORITHM- SIMULATED ANNEALING (MGASA)

Genetic Algorithms (GAs) is a stochastic search algorithm. The adaptive search process which has been effectively utilized in different types of challenging numerical optimization problems. They have been effectively utilized to resolve signal processing, system identification and path planning issues. Same way, Simulated Annealing (SA) is typically utilized heuristic algorithm for combinatorial optimization which is inspired by biological species. The Simulated Annealing (SA) is an analogy between the way, in which the metal freezes and cools into a minimum energy crystalline structure (The Annealing Process). In the simulated annealing optimization process, the probability is determined by Metropolis principle while optimization solution randomly walks in its neighborhood. In this proposed approach, the software effort estimation process is done based on Modified Genetic Algorithm-Simulated Annealing (MGASA) approach. A Flow chart for the proposed MGASA is shown in the Fig.1.

IV. SOFTWARE EFFORT ESTIMATION BASED ON MGASA

Genetic algorithm uses mathematically utilizing vectors, which are having two quantities such as magnitude and direction. Genetic algorithm parameters are denoted with one dimensional vector which is represented as a chromosome in genetic terms. Therefore the values utilized in each genetic algorithm parameters can be defined as genes, which are modified utilizing natural selection. Effort estimation rules may contain different types of parameters like Exponential Moving Average (EMA), Moving Average Convergence-Divergence (MACD), etc. A Modified Genetic algorithm uses these parameters as an input value. If any small changes are attained in the estimated value, then it will be retained for the next generation in genetic algorithm phase.

There are three types of genetic operations performed such as Crossover, Mutation and Selection. Crossover denotes the reproduction. For example certain characteristics of a child are taken from its parents. Mutations are utilized to transfer one generation's genetic diversity to next generation by introducing small changes randomly. Final process is selection, in which stage

individual genomes are selected from future breeding (crossover).

Crossover operator: Define $S_1 = \{S_{11}, S_{12}, \dots, S_{1n}\}$, $S_2 = \{S_{21}, S_{22}, \dots, S_{2n}\}$ are two chromosomes. Offspring of crossover S_1 and S_2 are S_3 and S_4

$$S_3 = \{s_i | \text{if } i \leq r, s_i \in S_1, \text{ else } s_i \in S_2\} \quad (1)$$

$$S_4 = \{s_i | \text{if } i \leq r, s_i \in S_2, \text{ else } s_i \in S_1\} \quad (2)$$

Mutation operator: a chromosome $S_1 = \{S_{11}, S_{12}, \dots, S_{1n}\}$ $S_1 = \{s_{11}, s_{12}, \dots, s_{1n}\}$, choose a random integer number $0 \leq r \leq n$ $0 \leq r \leq n$ is represents the mutation of S_1 .

$$S_3 = \{s_i | \text{if } i \neq r, \text{ then } s_i = s_{1i}, \text{ else } s_i = \text{random of } (S_1)\} \quad (3)$$

Selection operator: Assume there are m individuals, then select $(m/2) \lceil m/2 \rceil$ individuals and remove others. The selected individuals have "more fitness" that denotes their profits which are greater than fitness value. In Genetic algorithm fitness values are used for optimization process. Simulated annealing is used to find the optimized value. These solutions determine the probability value by using Metropolis principle.

Software effort estimation model is defined as

$$E = f(LOC, ME) \quad (4)$$

Where LOC is the number of line and E denotes the Effort, ME denotes methodology utilized in software project and f be the nonlinear function according to the ME and LOC . *ME and LOC*

Nonlinear function f defines the following equation

$$E = a \cdot LOC^b + c \cdot ME^d + e \quad (5)$$

where, a, b, c, d and e *a, b, c, d and e* are the five different parameters. The nonlinear function are modified in the form for proposed study

$$E = a \cdot LOC^b + c \cdot ME^d + e \cdot \ln(ME) + f \cdot \ln(LOC) + g \quad (6)$$

Modified function has seven different parameters which are a, b, c, d, e, f and g . *a, b, c, d, e, f and g* The main goal of this approach is to find a global optimum value of an objective function in the given searching space. Heuristic and deterministic method is used to solve multidimensional optimization problem of effort

estimation. The optimal value modified function uses following equation

$$\text{Minimize } \sum_{i=1}^n (E_{meas} - E_{comp})^2 \quad (7)$$

Where E_{comp} defines the computed value of effort according to MGASA and E_{meas} is defined as measured effort value. With the intention of minimizing the total squared error, the MGASA is utilized to change the parameter values of nonlinear function.

V. CONCLUSION

The paper suggests a new methodology for the software effort and cost estimation for the software projects. These are the tasks that are difficult for any Software Project Manager. Hence the Modified Genetic Algorithm-Simulated Annealing (MGASA) soft computing technique makes the modeling method of estimating the effort and cost in more efficient manner. The MGASA approach is developed by the cost model inspired from the COCOMO model. It is implemented based on the Expert Estimation technique and tested using NASA data set. In future this methodology can be implemented and proved using other data sets also. This proposed approach is simple to implement and provide efficient result.

REFERENCES

- [1] Zhang Dan, "Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization", IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 180 – 185, 2013.
- [2] Jyoti G. Borade, Vikas R. Khalkar, "Software Project Effort and Cost Estimation Techniques", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 3, pp. 730-739, 2013.
- [3] Dejaeger, K., Katholieke, Verbeke, W. Martens, D. Baesens, B., "Data Mining Techniques for Software Effort Estimation: A Comparative Study", IEEE Transactions on Software Engineering, Vol.. 38, pp. 375-397, 2012.
- [4] Ning Nan and Donald E. Harter, "Impact of Budget and Schedule Pressure on Software Development Cycle time and Effort", IEEE Trans. on Software Engg., vol. 35, pp. 624-637, 2009
- [5] Magne Jorgensen, Tanja M. Grusehke and R. Gupta, "The Impact of Lessons- Learned sessions on Effort Estimation and Uncertainty Assessments", IEEE Trans. on Software Engg., pp. 368-383, 2009.

- [6] Tim Menzies, Andrew Butcher, David Cok, Lucas layman, Forrest Shull, Burak Turhan, "Local vs Global lessons for defect Prediction and Effort Estimation", IEEE Transactions on Software Engg., Vol.39, pp. 822-834, 2013
- [7] Ekram kocaguneli , Tim Menzies, Ayse Basar Bener and Jacky W Keung, "Exploiting the essential assumptions of Analogy based Effort Estimation", IEEE Transactions on Software Engg., pp. 425-437, 2012
- [8] Mark Harman, Afshin Mausouri, "Search based Software Engineering : Introduction to special issue of IEEE Trans. on Software Engg.", IEEE Transactions on Software Engg., pp. 737-741, 2010
- [9] VahidKhatibi, Dayang N. A. Jawawi, "Software Cost Estimation Methods: A Review",Journal of Emerging Trends in Computing and Information Sciences, Volume 2 No. 1,2079-8407,2010.
- [10] Saleem Basha, Dhavachelvan P, "Analysis of Empirical Software Effort Estimation Models", International Journal of Computer Science and Information Security (IJCSIS), 1947-5500,Vol. 7, No. 3, 2010.
- [11] Magne Jorgensen, "Evidence based guidelines for assessment of Software development Cost Uncertainty", IEEE Transactions on Software Engg., pp. 942-954, 2005
- [12] M.Jorgensen, "A Review of Studies on Expert Estimation of Software Development Effort", Search conducted at Simula Research Lab, pp. 695-707, 2011
- [13] Tim Menzies, Zhihao chen, Jairus Hihn and Karen Lum, "Selecting best practices for Effort Estimation", IEEE Transactions on Software Engg., pp. 883-895, 2006
- [14] PichaiJodpimai, PeraphonSophsathit, ChidchanokLursinsap, "Analysis of Effort Estimation based on Software Project Models",ISCIT,IEEE 4244-4522, 2009.
- [15] Attarzadeh, I, Mehranzadeh, A.Barati, A., "Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Effort Estimation", International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), IEEE, 24-26, 4673-2640, 2012.
- [16] Ch.Satyananda Reddy, KVSVN Raju, "An Optimal Neural Network Model for Software Effort Estimation", International Journal of Software Engineering (IJSE), Vol.3 No.1, 2010.
- [17] GeetikaBatra, "An Intelligent Algorithm for Software Effort Estimation", International Journal of Emerging Technology and Advanced Engineering 2250-2459, Volume 2, Issue 5, 2012.
- [18] Olga Fedotova, Leonor Teixeira, Helena Alvelos, "Software Effort Estimation with Multiple Linear Regression: Review and Practical Application", Journal of Information Science and Engineering 29, 925-945, 2013.