Software Cost Estimation using Enhanced Artificial Bee Colony Algorithm

Sevgi Yigit-Sert, Pinar Kullu
Computer Engineering Department
Ankara University
Ankara, Turkey

***Abstract*—Cost estimation is very important in software development progress so that resource and time planning can be successfully performed. Accurate estimation of cost is directly related to the decision making mechanism in the software development process. The underestimated cost might lead to fewer resources and budget problems; in contrast, customer satisfaction might diminish due to waste of resources. This study represents an estimation model for the effort required for the development of software projects using a variant of artificial bee colony (ABC) algorithm. The proposed model is performed over a dataset consisting of NASA software projects and has better performance than the previous studies.**

Keywords*—Software cost estimation; enhanced artificial bee colony algorithm; NASA software

I. Introduction

As technological advances have risen, the competition in the software industry has been increased. Thus the prediction of features that affect software processes such as performance, cost, and reliability becomes essential for software companies. The cost is the most fundamental element to make a software project manageable. With the precise cost estimation, companies might evaluate the project progress and analyze what is effective for the project and get better decisions during software life cycle. It provides a reliable budget and delivery of the software product within the promised period of time.

In the literature, there are a number of models for estimating the software cost. These models are developed using algorithmic approaches, expert judgments, analogy, top-down and bottom-up strategies and machine learning techniques. Algorithmic models, one of the successful and simple software cost estimation techniques, use mathematical formulas that are based on the effort in terms of person-months at various life cycles of the software for a project [1]. In this study, we implement enhanced artificial bee colony method proposed in [2] to establish an algorithmic model for estimating software cost.

The organization of this paper is as follows: Section II explains some algorithmic models, the methodology that has been used is detailed in Section III, in Section IV, the evaluation and results of our model are shown, Section V discusses the conclusions.

II. Background

There are various major models to estimate software cost in the literature [3]-[5]. These models are based on a large number of software projects and applications completed in various organizations. In the formulations of these models, the software size is based on KLOC (Kilo Lines Of Code) that is the number of lines of source codes. The structure of the formulation is as follows:

$$\text{Effort} = a(\text{KLOC})^b \quad (1)$$

It is thought that the software effort can be found by multiplying a power of the number of lines of the software code by a coefficient. Values a and b are constants obtained experimentally. The proposed models in the literature aim to find these constants, a and b, mathematically by using some software projects.

Some of these models are:

- Walston-Felix Model [6]: This model was proposed by Waltson and Felix in 1977. The model was developed from a database of 60 different projects and provides a correlation between effort and source code lines. This formulation can be expressed as:

$$\text{Effort} = 5.2 * (\text{KLOC})^{0.91} \quad (2)$$

- Doty Model: In the same year with Walston-Felix Model, Doty presents the model called with his name [7], to estimate the effort for the number of lines of code as in the following equation:

$$\text{Effort} = 5.288 * (\text{KLOC})^{1.047} \quad (3)$$

- Halstead Model [8]: This model, which predicts the error rate, was proposed in 1977. In this model, in-depth analysis is not required for the programming structure. It provides to recommend the code length and volume metrics of the software. The formulation of this model is:

$$\text{Effort} = 5.2 * (\text{KLOC})^{1.5} \quad (4)$$

- Bailey-Basili Model [9]: Bailey and Basili proposed a meta-model in 1981 to calculate the software effort prediction equations that best fits the given development environment. The resulting model is based on the collection of data such as differences between

projects and their environmental factors. This equation is expressed as follows:

$$\text{Effort} = 0.73 * (\text{KLOC})^{1.16} + 3.5 \quad (5)$$

- Artificial neural networks (ANN) are designed to model the way in which human brain performs. ANNs consist of simple interconnected units and are usually organized as layers. Similar to the information processing method of the brain, after a learning process, ANN has the ability to collect information and the ability to store and generalize this information with the weight of connections between cells. The learning process involves learning algorithms that enable the renewal of ANN weights to achieve the desired goal. Finnie et al. [10] used ANN method to learn parameters in effort estimation of software development in 1977. The prediction model was based on 50 sample cases.
- Genetic algorithm (GA): Genetic algorithm that is inspired by the evolutionary process, is a method used to solve optimization problems. Authors in [11] and [12] use genetic algorithms to achieve improvement on effort estimation.
- ABC algorithm model: This algorithm was developed by Karaboga et al. [13] inspired by the behaviors of biological swarm intelligence. The intelligence in foraging behaviors of honeybees is leveraged to find the optimal solutions for problems. The principal of the algorithm is based on the behaviors of honeybees on finding nectar-rich food sources and information sharing about food sources [14]. The goal of foraging process is to minimize the consumed energy and time while finding nectar-rich sources. The ABC algorithm is very simple and robust optimization algorithm. For cost estimation of a software project, it aims to learn parameters of (1).
- H-ABC [15]: The ABC algorithm can be initialized with a distribution that is either uniform or non-uniform according to the location of the optimum. In this model, Halton points are chosen among the initial points that are generated by ABC algorithm and this model is applied to optimize the parameters given in (1).

III. Methodology

The advantages of ABC algorithm are simplicity, be applicable to many different areas and its outstanding performance. On the other hand, it suffers from slow convergence and tendency to local optima [16]. Researchers have been proposed to overcome these drawbacks [2], [16]-[20]. Abro et al. [2] proposed a novel variant of basic ABC algorithm which is called Enhanced-ABC algorithm (E-ABC). The E-ABC algorithm uses multiple global-best possible-solutions (GBPS) rather than single global-best possible-solution in exploration and exploitation phase. It also has a novel procedure in the scout bee phase to improve the performance.

The bee swarm in basic ABC algorithm consists of three groups to accomplish different tasks: employed bees, onlookers, and scouts. The employed and onlooker bees indicate the number of possible solutions in the population. A scout bee is formed with the transformation of the employed bee when a food source exhausted totally [13]. A scout flies around the hive and produces a food source randomly. In the modeling of the optimization problem, the search space is represented by the hive, each possible solution is considered as a food source and the solution quality is shown by the nectar amount of these food sources.

An employed bee investigates near food sources around the hive. After she finds a food source, she tries to explore the multiple best-found locations of search space instead of the neighborhood of an assigned food source like in basic ABC algorithm.

$$s_{kl} = f_{kl} + \theta_{kl}(f_{kl} - f_{best\ /\ secondBest/\dots/n,l}) \quad (6)$$

where s_{kl} is the produced new source from f_{kl} which represents the old food source, θ_{kl} is randomly selected number between (-1,1) [2]. $f_{best\ /\ secondBest/\dots/n,l}$ represents one of the high-nectar-content food sources.

The fitness value of new-found food source (s_{kl}) is evaluated using the following equation:

$$fit_k = \begin{cases} \frac{1}{1+o_k}, & o_k \geq 0 \\ 1 + abs(o_k), & o_k < 0 \end{cases} \quad (7)$$

where o_k represents objective function value of k_{th} food source.

If the nectar amount of newly found solution is higher, then the bee learns the new one and forgets the previous solution. Otherwise, she continues with the previous position.

All population is divided into different groups, and the number of groups is determined by a user-defined value. The Global best possible solutions (GBPSs) are selected to assign each one of them to a group, regardless of its existence [2]. The aim of this task is to improve the fitness of possible solutions.

In exploitation process, employed bees share information about food sources by dancing. Each onlooker bee chooses a high-quality food source based on the observation of the dance. The quality of the food source is decided by the probability value (p_k):

$$p_k = \frac{fit_k}{\sum_{i=1}^{SN} fit_i} \quad (8)$$

where fit_k is the fitness value (nectar amount of food source) in k_{th} position. If an onlooker bee does not find a better solution than old one, she produces a new solution using the following equation:

$$s_{kl} = f_{kl} + \theta_{kl}(f_{kl} - f_{best,l}) \quad (9)$$

This formula is intended to improve exploitation process and runs if (1) failed to generate better solution than the existing solution and Selection (S), an additional control variable of E-ABC algorithm, is larger than a randomly generated number. Authors in [2] stated that higher value of S accelerates the speed of convergence.

In E-ABC algorithm, a scout bee calculates new food sources, s_{nl}, according to the formula below:

$$s_{nl} = (f_{best,l})\beta_{nl} \quad (10)$$

where $f_{best,l}$ represents one of the high-nectar-content sources. β_{nl} is a random number ranging between 0.9 and 1.1 [2].

There are a few critical parameters to be specified in E-ABC algorithm:

1) SN: the size of the population.

2) Limit value: iteration number to quit processing on a food source.

3) MIN: Maximum iteration number.

4) S value: a decision number to search a new food source.

5) n value: the number of groups in population to be assigned GBPSs.

The values are assigned as shown in Table I.

TABLE I. THE TUNING PARAMETERS FOR THE E-ABC

Parameters	Values
SN	100
Limit	20
MIN	750
Selection (*S*)	0.5
Number of groups (*n*)	5

B. Results

The performance of the predicted model is measured by two mainly used metrics, such as Mean Magnitude Relative Error (MMRE) and Root Mean Square Error (RMSE).

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\frac{|Effort - Estimated\ Effort|}{Effort} \quad (11)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Effort - Estimated\ Effort)^2} \quad (12)$$

IV. RESULTS AND DISCUSSIONS

This section explains the details of the used dataset and obtained results over this dataset using some common metrics.

A. Dataset

The proposed estimation model using E-ABC method in [2] is conducted on a NASA dataset [9] that contains 18 projects consisting of Kilo Line of code (KLOC), Methodology (ME) and the measured effort. The Effort is in terms of person-months. Table II shows the details of this dataset.

To estimate model parameters, the dataset is split into training set that consists of the first 13 projects, and test set that contains the rest of the projects in the dataset. The computed efforts obtained by E-ABC are shown in Table II.

TABLE II. NASA DATA OF EFFORT ESTIMATION

Project No	KLOC	Methodology	Measured Effort
1	90.2	30	115.80
2	46.2	20	96.00
3	46.5	19	79.00
4	54.5	20	90.80
5	31.1	35	39.60
6	67.5	29	98.40
7	12.8	26	18.90
8	10.5	34	10.30
9	21.5	31	28.50
10	3.1	26	7.00
11	4.2	19	9.00
12	7.8	31	7.30
13	2.1	28	5.00
14	5	29	8.40
15	78.6	35	98.70
16	9.7	27	15.60
17	12.5	27	23.90
18	100.8	27	138.30

Table III shows the estimated results by E-ABC algorithm are mostly close to the measured effort except for some projects, such as project 2. Some models known as benchmarks, i.e. Halstead, Doty etc., and previous studies using NN and other evolutionary algorithms, namely GA, H-ABC, are used to compare the performance of E-ABC model in terms of MMRE and RMSE. The results of the model produced by basic ABC algorithm are also provided. As shown in Table IV, basic ABC algorithm outperforms Halstead, Walston-Felix, Doty, GA, and H-ABC in MMRE metric, but not NN. It also gives lower RMSE than all models except GA., E-ABC yields better performance than all other models. It provides 22%, 78%, 81% gains in MMRE metric, and 70%, 54%, 64% gains in RMSE metric according to NN, GA, and H-ABC, respectively.

TABLE III. ESTIMATED EFFORT CALCULATED BY E-ABC

Project No	Measured Effort	E-ABC Estimated Effort
1	115,8	115,8219
2	96	62,1928
3	79	62,568
4	90,8	72,5151
5	39,6	43,052
6	98,4	88,466
7	18,9	18,8653
8	10,3	15,6933
9	28,5	30,5485
10	7	5,05
11	9	6,6968
12	7,3	11,9051
13	5	3,5164
14	8,4	7,8749
15	98,7	101,9125
16	15,6	14,579
17	23,9	18,454
18	138,3	128,4215

TABLE IV. THE PERFORMANCE RESULTS OF EFFORT ESTIMATION

Performance Criteria	Model Used								
	NN System	Halstead	Walston-Felix	Bailey Basili	Doty	GA Estimated Effort	H-ABC Estimated Effort	ABC Estimated Effort	E-ABC Estimated Effort
MMRE	11,7896	175,6550	155,5590	20,2885	302,5020	41,9072	49,0565	23,62405	9,1961
RMSE	17,4475	308,7097	123,4570	25,0224	299,4740	11,4867	14,6136	13,50596	5,2703

V. CONCLUSION

In this study, we propose a novel model to estimate the software effort. The E-ABC algorithm is used as it is faster convergence than basic ABC algorithm. With E-ABC algorithm, an algorithmic effort estimation model based on the number of lines of software has been developed. We tested our model on the NASA software projects dataset. The proposed model is compared with the algorithmic prediction models in the literature using MMRE and RMSE metrics. Results show that our model achieves better results than the previous models. The limitation of this study is that the size of the used NASA dataset is small. As future work, ABC and E-ABC can be employed to model software cost estimation over different datasets.

REFERENCES

[1] Reddy, Ch Satyananda, and Raju KVSVN. "An optimal neural network model for software effort estimation.", Int. J. of Software Engineering, IJSE 3.1 (2010): 63-78.

[2] Abro, Abdul Ghani, and JUNITA MOHAMAD SALEH. "Multiple-global-best guided artificial bee colony algorithm for induction motor parameter estimation.", Turkish Journal of Electrical Engineering & Computer Sciences 22.3 (2014): 620-636.

[3] Kumari, Sweta, and Shashank Pushkar. "Comparison and analysis of different software cost estimation methods.", IJACSA International Journal of Advanced Computer Science and application 4.1 (2013).

[4] Olu-Ajayi, Razak. "An Investigation into the Suitability of k-Nearest Neighbour (k-NN) for Software Effort Estimation.", International Journal Of Advanced Computer Science And Applications 8.6 (2017): 227-233.

[5] PVGD, Prasad Reddy. "Application of Fuzzy Logic Approach to Software Effort Estimation. ", IJACSA Editorial (2011).

[6] Walston, Claude E., and Charles P. Felix. "A method of programming measurement and estimation", IBM Systems Journal 16.1 (1977): 54-73.

[7] Doty Associates, Inc., "Software Cost Estimates Study," vol. 1, (1977): 77-220.

[8] M. H. Halstead, "Elements of Software Science," Elsevier, New York, (1977).

[9] Bailey, John W., and Victor R. Basili. "A meta-model for software development resource expenditures.", Proceedings of the 5th international conference on Software engineering. IEEE Press, 1981.

[10] Finnie, Gavin R., Gerhard E. Wittig, and Jean-Marc Desharnais. "A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models. ", Journal of systems and software 39.3 (1997): 281-289.

[11] Burgess, Colin J., and Martin Lefley. "Can genetic programming improve software effort estimation? A comparative evaluation.", Information and software technology 43.14 (2001): 863-873.

[12] Sheta, Alaa F. "Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects.", Journal of Computer Science 2.2 (2006): 118-123.

[13] Karaboga, Dervis, and Bahriye Basturk. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. ", Journal of global optimization 39.3 (2007): 459-471.

[14] Yigit, Sevgi, Recep Eryigit, and Fatih V. Celebi. "Optical gain model proposed with the use of artificial neural networks optimised by artificial bee colony algorithm. ", Optoelectron Adv Mat 5 (2011): 1026-1029.

[15] Sharma, Tarun Kumar, and Millie Pant. "Halton based initial distribution in artificial bee colony algorithm and its application in software effort estimation.", Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference on. IEEE, 2011.

[16] Zhu, Guopu, and Sam Kwong. "Gbest-guided artificial bee colony algorithm for numerical function optimization.", Applied mathematics and computation 217.7 (2010): 3166-3173.

[17] Karaboga, Dervis, and Beyza Gorkemli. "A quick artificial bee colony (qABC) algorithm and its performance on optimization problems.", Applied Soft Computing 23 (2014): 227-238.

[18] Akbari, Reza, Alireza Mohammadi, and Koorush Ziarati. "A novel bee swarm optimization algorithm for numerical function optimization.", Communications in Nonlinear Science and Numerical Simulation 15.10 (2010): 3142-3155.

[19] Gao, Wei-feng, San-yang Liu, and Ling-ling Huang. "A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. ", IEEE Transactions on Cybernetics43.3 (2013): 1011-1024.

[20] Banharnsakun, Anan, Tiranee Achalakul, and Booncharoen Sirinaovakul. "The best-so-far selection in artificial bee colony algorithm. ", Applied Soft Computing 11.2 (2011): 2888-2901.