International Journal of Advanced Research and Development
ISSN: 2455-4030
Impact Factor: RJIF 5.24
www.advancedjournal.com
Volume 3; Issue 1; January 2018; Page No. 167-169

Software effort estimation using bee colony optimization

Mandeep Kaur
Assistant Professor, Khalsa College for Women, Civil Lines, Ludhiana, Punjab, India

Abstract
Effort estimation is one of the important and challenging tasks in software development. Many algorithmic models have been used to estimate the software development effort which needs to be measured accurately such that there is reduction in the gap between the actual effort and estimated effort. In the recent past many nature inspired optimization techniques have gained popularity as an efficient tool to estimate the software development effort. These optimization techniques are used to optimize the parameters of other models. One such technique is bee colony optimization which is based on the natural behavior of bees. This paper presents a review on the use of bee colony optimization for software development effort estimation.

Keywords: bee colony optimization, software effort estimation, COCOMO, optimization, LOC

1. Introduction
Software effort estimation is the foremost step in the software planning process. Effort estimates help to measure the amount of manpower, cost, hardware, software required to develop software. This estimate should be made accurately and efficiently as overestimation and underestimation both, results in problems related budgeting, scheduling, quality, etc. Since cost estimates depend on effort estimates, it has been observed that inaccurate effort estimates have lead to 89% cost overruns [1]. The accuracy of the different models is decided using fitness function that checks out how close the given solution was to the goal.

Various fitness functions which are used to evaluate the accuracy of estimated effort are [2]: MMRE (Mean Magnitude of Relative Error),

$$\text{Where, MMRE} = \frac{1}{n}\sum_{i=1}^{n} |\text{actual effort} - \text{estimated effort}| * 100 \quad (1.1)$$

$$\text{RMSE (Root Mean Square Error)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{actual effort} - \text{estimated effort})^2} \quad (1.2)$$

$$\text{BRE (Balance Relative Error)} = \frac{|\text{actual effort} - \text{estimated effort}|}{\min(\text{Actual effort}, \text{estimated effort})} \quad (1.3)$$

2. COCOMO Model
Various software metrics that are used to estimate the software effort are Lines of Code (LOC) and Function Points [3]. COCOMO (Constructive Cost Model) was introduced by Dr. Barry Boehm [4]. It uses Kilo Lines of Code (KLOC) as input to estimate effort,

$$\text{Where, Effort} = a\,(KLOC)^b \quad (2.1)$$

COCOMO model has three levels:
a) Basic COCOMO Model [5]
b) Intermediate COCOMO Model [6]
c) Advanced or Detailed COCOMO Model [7]

Dr. Barry Boehm classified software projects into three categories based on their complexities:

Table 1: Types of Software Projects

Model	Effort Equation
Organic	$2.4(KLOC)^{1.05}$
Semi Detached	$3.0(KLOC)^{1.12}$
Embedded	$3.6(KLOC)^{1.20}$

3. Bee Colony Optimization
Bird flocks, school of fish, ant colonies, bee colony, etc. are all swarm intelligent systems [8] where the individuals interact continually with similar individuals and their environment. Bee colony optimization is a meta-heuristic optimization technique which is inspired by the natural behavior of honey bees where the search for food consists of three essential components:

a) **Food Source** [9]: Value of food is dependent upon its richness, closeness to nest.
b) **Employed Foragers** [10]: It consists of bees associated with particular food source who convey the information of the food source to other bees.
c) **Unemployed Foragers** [11]: These are of 2 types:
   i) **Scouts:** They search for food around the nest.
   ii) **Onlookers**: These wait in the nest.

Employed foragers convey the information about the food source with onlookers by dancing which is commonly known as Waggle dance. The longer the duration of dance, the higher is profitability of food searched. Onlookers then decide which food source is to be selected based on the waggle dance of bees. Bee colony optimization uses this concept of bees. It consists of two passes:

a) **Forward Pass:** In this phase the bees explore the search space by applying a certain number of moves, improving the solution in each move.

b) Backward Pass: The information of the solution is communicated back to the onlookers in the hive. The information contains the proximity to objective function. Then the onlookers decide the best solution. This process is repeated till the best solution is found.

The probability of selecting the solution given by bees is selected using the formula:

$$\text{Probability of selection} = \frac{(\max(\text{all solutions}) - \text{current solution})}{\max(\text{all solutions}) - \min(\text{all solutions})} \qquad (3.1)$$

3.1 Bee Colony Optimization Algorithm

Initialize no.of_bees, no.of_forward_passes, stopping_criteria
Do
    Assign empty solution to each bee
    For each forward pass
        For each bee
        For each move
            Evaluate possible moves
            Select best move
        End for
        End for
            For each bee (backward pass)
                Evaluate each solution and find Loyalty solution
            End for
        End for
    Evaluate all solutions and find the best one and make it global solution
End while

4. Related Work

Khuat and Hanh [12] proposed combination of Teaching Learning based optimization and Artificial Bee Colony and used this combination to optimize the parameters of COCOMO II model. The model was tested on NASA software project dataset. It was observed that the proposed model resulted in improvement of COCOMO II parameters and therefore provided better estimation as compared to original COCOMO II.

Sehra, *et al.* [13] implemented Bee colony optimization, Particle swarm optimization and Ant colony optimization to optimize the parameters of COCOMO Model. The proposed model was tested on Interactive Voice Response projects. The model was tested for Mean Magnitude of Relative Error and it was observed that the proposed model provided accurate estimates as compared to other estimation models. The value of Mean Magnitude of Relative Error for Bee Colony Optimization was less than other existing effort estimation algorithmic models.

Khuat and Hanh [14] used a directed artificial bee colony algorithm to tune the parameters of existing models of software effort estimation. The proposed model was tested on NASA software datasets. The results obtained specified that the proposed model improved the performance of the estimations as compared to other estimation models. The proposed model was tested for various evaluation functions like Mean Magnitude of Relative Error, MdMRE, MAR, PRED(25). The efficiency of the proposed model was indicated to be enhanced by 5.5%

Azzeh [15] used Bees optimization to find optimal number of analogies and coefficient values for case based reasoning where case based reasoning is providing solutions for a problem where the solutions are in turned based on solutions of past and similar problems. The proposed method helped to reduce the estimate error and was found to be useful for extension of Case-based effort prediction model.

Sharma and Pant [16] used Halton points for the initial distribution of artificial bee colony optimization algorithm the results obtained were compared with results obtained from rand (0,1) uniform distribution. The model was evaluated on NASA Software project dataset and was used to compute cost model parameters.

Chalotra *et al.* [17] used Bee Colony Optimization on COCOMO model to estimate effort which is capable of making multi-agent system for solving complex optimization problems. The proposed model was evaluated on IVR dataset. Out of many partial solutions, the best solution was selected based on Mean Magnitude of Relative Error and was considered as best global solution. The proposed model was then compared with other exiting models and results obtained for proposed model was evaluated to be better in performance as compared to other models.

5. Conclusion

Software effort estimation is essential activity of software development. Different models are used to accurately measure the software development effort. Bee colony optimization is a swarm intelligence technique which has been widely used for software effort estimation. This paper has reviewed the use of bee colony optimization by many researchers. It has been observed that this nature inspired technique is very effective in optimizing the parameters of other models thereby helping to estimate the software development effort accurately.

6. References

1. Shepperd M, Schofield C. Estimating software project effort using analogies. IEEE Transactions on Software Engineering. 1997; 23(11):736-43.
2. Malathi S, Sridhar S. Analysis of Size Metircs and Effort Performance Criterion in Software Cost Estimation, Indian Journal of Computer Science and Engineering, 2012; 3(1):24-31.
3. Arshad A. A Critical Review of Software Size and Effort Estimation, International Journal of Computer and Electronics Research. 2014; 3(2):96-99.
4. Mall R. Fundamentals of Software Engineering, PHI Learning Private Limited, New Delhi, 2008.
5. Boehm BW. Software engineering economics, Englewood Cliffs, NJ: Prentice-Hall, 1981.
6. Boehm BW. Royce. Ada COCOMO and the Ada Process Model, Proceedings, Fifth COCOMO Users Group Meeting, Software Engineering Institute, Pittsburgh, 1989
7. Clark B, Horowitz E, Madachy R, Shelby R, Westland Ch. The COCOMO 2.0 Software Cost Estimation Model, International Society of Parametric Analysts, 1995.
8. Zhang Y, Agarwal P, Bhatnagar V, Balochian S, Yan J.

Swarm Intelligence and Its Applications, The Scientific World Journal, 2013.
9. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4436220/
10. Davidovic T, Teodorovic D, a Selmic M. Bee Colony Optimization part 1: The Algorithmic Overview, Yugoslav Journal of Operations Research, 2015; 25(1):33-56.
11. https://econ.ubbcluj.ro/~rodica.lung/taco/ArtificialBeeColony.pdf
12. Khuat T, Hanh LTM. Applying Teaching-Learning to Artificial Bee Colony Optimization for Parameter Optimization of Software Effort Estimation Model, Journal of Engineering Science and Technology.
13. Sehra SK, Brar YS, Kaur N. April Evolutionary Computing Techniques for Software Effort Estimation, International Journal of Computer Science & Information Technology, 2017; 9(2).
14. Khuat TT, Hanh ML. Optimizing Parameters of Software Effort Estimation Models using Directed Artificial Bee Colony Algorithm, Informatica. 2016; 40:427-436.
15. Azzeh M. Adjusted Case-Based Software Effort Estimation Using Bees Optimization Algorithm, Springer-Verlag Berlin Heidelberg, 2011, 315-324.
16. Sharma TK, Pant M. Halton Based Initial Distribution in Artificial Bee Colony Algorithm and Its Application in Software Effort Estimation, International Conference on Bio-Inspired Computing: Theories and Applications BIC-TA, 2011.
17. Chalotra1 S, Sehra SK, Brar YS, Kaur N. Tuning of COCOMO Model Parameters by using Bee Colony Optimization, Indian Journal of Science and Technology, 2015; 8(14).