

# Tuning of COCOMO Model Parameters by using Bee Colony Optimization

Sherry Chalotra<sup>1\*</sup>, S. K. Sehra<sup>1</sup>, Y. S. Brar<sup>2</sup> and Navdeep Kaur<sup>3</sup>

<sup>1</sup>CSE Department, Guru Nanak Dev Engineering College, Ludhiana - 141006, Punjab, India; sherry2009in@gmail.com

<sup>2</sup>EE Department, Guru Nanak Dev Engineering College, Ludhiana - 141006, Punjab, India

<sup>3</sup>CSE Department, Sri Guru Granth Sahib World University, Fatehgarh Sahib - 140407, Punjab, India

## Abstract

Constructive Cost Model (COCOMO) used parameters for software effort estimation, which were calculated in 1981 by regression analysis of 63 types of project data; therefore applying these parameters to current project development will not generate accurate results. The objective of current research is applying Bee Colony Optimization (BCO) meta-heuristic approach to optimize the parameters of COCOMO model for improving software cost estimation. The Bee Colony Optimization (BCO) is a new branch of Swarm Intelligence and has been applied successfully to various engineering disciplines. BCO approach is a "bottom-up" approach to modeling where special kinds of artificial agents are created by analogy with bees. These artificial agents or bees are used to solve complex combinatorial optimization problems. The proposed model validation is carried out using Interactive Voice Response software project dataset of a company. The results generated by the proposed model are compared to those obtained by methods proposed in the literature using Walston-Felix, SEL, Bailey-Basil, COCOMO II and Halstead models. The BCO approach generates various partial solutions and best solution is selected based on Mean Magnitude of Relative Error. The results obtained show that the proposed BCO based model is able to improve the accuracy of cost estimation and also outperform other models.

**Keywords:** Bee Colony Optimization, Constructive Cost Model, Optimization, Software Cost Estimation

## 1. Introduction

As software being an essential part of industry, estimation regarding effort and cost becomes important. So it becomes important for cost estimation at earlier stages as it is central to both management and control of project<sup>1,2</sup>. The most important task of software development is to accurately estimate cost and effort required. The inability of industry not providing accurate estimates lead to an average of 89% cost overruns<sup>3</sup>. The accuracy of such predictions depends on MMRE (Mean Magnitude of Relative Error) value<sup>1,2</sup>. MMRE is fairly conservative with a bias against overestimates<sup>1</sup>. In this paper MMRE is used as a fitness function to measure accuracy of COCOMO model by using Bee colony optimization. COCOMO and

LOC is used and implemented on Bee algorithm. BCO is a meta-heuristic technique that mimics honey bee's foraging behavior to compute problem solution. Bees locally search solution space then combine local solutions to get global solution of problem as it optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality<sup>4,5</sup>.

The main objective of this paper is to optimize the parameters of COCOMO model by using Bee Colony Optimization technique and then comparing it with other models. The whole paper is coordinated in different sections. Section 2 discusses COCOMO model and Section 3 explains other existing models. Section 4 discusses Bee Colony Optimization. Proposed model is explained in Section 5 and results are figure out in Section 6.

\* Author for correspondence

## 2. COCOMO Model

For estimation availability of models is not less but COCOMO is well known model<sup>2</sup>. The successor of COCOMO is COCOMO II<sup>6</sup> that was established in 1995 and provides more support for today's modern and demanding projects. The basic formula for COCOMO model is:

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

Where,

KLOC is estimated size of software project expressed in kilo line of code. a, b are constants that depend on project class. Software project are classified into three categories that depends on project's complexity. First, Organic that deals with well-understood applications and team is small, Second semidetached that contain both experienced and inexperienced staff, and last is embedded in which projects are complex<sup>7,8</sup> (Table 1).

**Table 1.** COCOMO Model Project Classes

Model	Effort Equations (in PM)
Organic	Effort = 2.4(KLOC) <sup>1.05</sup>
Semi detached	Effort = 3.0(KLOC) <sup>1.12</sup>
Embedded	Effort = 3.6(KLOC) <sup>1.20</sup>

## 3. Other Models

There are so many models that are used to estimate the effort. In this section few of them are discussed:

### 3.1 COCOMO II Model

COCOMO II model, which is extension of intermediate COCOMO, offers 3 sub-model as: Applications Composition based on object points<sup>9</sup>. Early Design and Post-architecture models. This can be used in many major decision situations like investments, tradeoffs, budgets etc. It is defined as:

$$\text{Effort} = 2.9 * (\text{KLOC})^{1.10} \quad (2)$$

### 3.2 SEL Model

This model is given by Software Engineering Laboratory (SEL) of the University of Maryland for effort estimation<sup>10</sup>.

The formula for this model is:

$$\text{Effort} = 1.4 * (\text{KLOC})^{0.93} \quad (3)$$

### 3.3 Halstead Model

Halstead defined his model as<sup>10</sup>

$$\text{Effort} = 0.7 * (\text{KLOC})^{1.15} \quad (4)$$

### 3.4 Bailey-Basil Model

Bailey-Basil developed model between delivered KLOC and defined as<sup>11</sup>:

$$\text{Effort} = 5.5 * (\text{KLOC})^{1.16} \quad (5)$$

## 4. Bee Colony Optimization

The Bee Colony Optimization (BCO) is a perfect example of swarm intelligence<sup>12,13</sup>. The BCO approach is a "bottom-up" approach to modelling where special kinds of artificial agents are created by analogy with bees. Artificial bee agents collaboratively solve complex combinatorial optimization problem<sup>14</sup>. BCO is a meta-heuristic<sup>15</sup> algorithm which uses the swarm behaviour of bees<sup>16,17</sup> and collective intelligence to deal with combinatorial problems<sup>18</sup>. Bee Algorithm is used for finding the best possible solutions for optimization problems. Artificial bees are responsible for solving problem. Bee algorithm consists of two type of alternating pass that contribute in a single step, forward pass and backward pass<sup>19</sup>. Both passes are problem dependent. During forward pass every bee is assigned with an empty solution. All bees explore search space on individual basis for number of predefined moves. Partial/Complete solutions are computed by every bee. This evaluation depends on individual exploration and past experience. After that bees go back to hive or colony and start the second phase of first step i.e. backward pass. During this all bees participate in decision making process and all evaluated solutions are shared by every bee by performing waggle dance<sup>20</sup> which is in shape of digit '8'. The solutions vary from bee to bee. This is the only time when bees communicate with one other and the best among all solution is considered as partial/complete solution of a problem. Only those solutions will be loyal that will be giving best solutions. Following formula is used for selecting best solution:

$$\text{loyalty} = \frac{\text{max\_sol\_value} - \text{current\_sol\_value}}{\text{max\_sol\_value} - \text{min\_sol\_value}} \quad (6)$$

Where,

max\_sol\_value is maximum value from set of solutions, min\_sol\_value is minimum value from set of solutions and current\_sol\_value is value for current solution.

Loyalty will be checked for every bee solution within a single move then from those set of solutions above mentioned value will be taken.

## 5. Proposed Model

One of the major problems in software development process is to estimate cost. Large number of estimation models is available but only few can reach the level of satisfaction. COCOMO is famous among all that give positive results. But due to increase complexity and over demanding software requirements it becomes less accurate. So we have proposed a model using BCO to optimize COCOMO model's constant parameters and evaluated MMRE for checking its performance. For better results performance should be minimum. For this IVR dataset<sup>21</sup> is used that gave the detail of 48 projects including its size (in KLOC), actual estimation (in PM) and time duration (in months). The equation for COCOMO used is:

$$\text{Effort} = 2.4 * (\text{KLOC})^{1.05} \quad (7)$$

KLOC is physical measure of size of source program that counts source lines and excludes comment and blank space in COCOMO model. The performance is evaluated by:

$$\text{MMRE} = \frac{1}{N} \sum_{i=1}^{i=N} \frac{\text{abs}(\text{actual effort} - \text{predicted effort})}{\text{actual effort}} \quad (8)$$

Where actual effort is taken from data set and predicted effort is from proposed model. The proposed algorithm is:

**Input:** IVR dataset values (project size (in KLOC) and actual estimation (in PM))

**Output:** Optimized values of COCOMO parameters.

**Initialization parameters:** B (Number of bees), NC (number of constructive moves)

**Step 1:** read data from dataset.

**Step 2:** specify number of bees, moves and stopping criteria.

**Step 3:** initialize parameters of COCOMO model.

**Step 4:** call BCO module.

**Step 5:** repeat step 6 to step 9 for every bee.

**Step 6:** calculate effort values for each bee in forward pass by using formula (1).

**Step 7:** generate partial solution for each bee by editing it according to changes required in forward pass by using formula (9).

$$\text{MRE} = \frac{\text{actual effort} - \text{predicted effort}}{\text{actual effort}} \quad (9)$$

**Step 8:** evaluate complete solutions in backward pass and choose best bee according to fitness function by using formula (8). Fitness function is MMRE and minimizing it is our goal.

**Step 9:** check loyalty for each bee by using (6).

**Step 10:** abandon the solutions which are not loyal to their solutions.

**Step 11:** get global best solution from best bee. Here global solution is with minimum MMRE value.

**Step 12:** finish BCO module.

## 6. Result

This section deals with the results obtained from proposed model which is implemented and tested on 48 projects. Results are given for 24 projects here. Table 2 gives the comparison between effort values while Table 3 gives MRE values and performance analysis based on final MMRE values are given in Table 4. MMRE value for proposed model comes out to be 0.11 after optimizing constant parameters of COCOMO. The standard MMRE value for COCOMO model is 0.43 while for BCO is 0.11. The results are then compared with different models (COCOMO II, SEL Model, Bailey Model, and Halstead Model) and proposed model gave the best results as shown in Table 4. The standard MMRE values for COCOMO II, SEL Model, Bailey Model, and Halstead Model are 0.24, 0.73, 0.62, and 0.61 respectively while proposed model gave 0.11. So proposed model is the best and optimal than all. Table 2, Table 3 and Table 4.

**Table 2.** Comparison between Effort Values of proposed model and other models

Project No.	BCO	COC OMO	COC OMO II	SEL	Bailey Basil	Halstead
1.	117.37	44.69	62.07	18.66	139.12	45.64
2.	22.21	13.94	18.31	6.65	38.40	8.64
3.	37.71	20.19	27.00	9.23	57.82	14.67
4.	18.34	12.19	15.91	5.90	33.11	7.13
5.	9.82	7.87	10.07	4.01	20.43	3.82
6.	21.34	13.55	17.78	6.49	37.23	8.30
7.	31.92	17.96	23.89	8.32	50.82	12.41
8.	29.14	16.85	22.35	7.87	47.37	11.33
9.	34.78	19.07	25.44	8.78	54/31	13.52
10.	22.59	14.10	18.54	6.72	38.90	8.78
11.	44.61	22.70	30.53	10.24	65.84	17.35
12.	39.21	20.74	27.78	9.46	59.59	15.25
13.	79.55	34.04	46.67	14.66	102.99	30.94
14.	60.37	28.06	38.12	12.36	83.20	23.48
15.	52.71	25.52	34.51	11.36	74.91	20.50
16.	11.28	8.67	11.14	4.37	22.74	4.39
17.	31.92	17.96	23.89	8.32	50.82	12.41
18.	25.14	15.20	20.05	7.18	42.26	9.78
19.	36.23	19.63	26.22	9.01	56.06	14.09
20.	34.78	19.07	25.44	8.78	54.31	13.52
21.	45.40	22.98	30.93	10.36	66.74	17.65
22.	29.14	16.85	22.35	7.87	47.37	11.33
23.	62.12	28.63	38.93	12.58	85.06	24.16
24.	28.46	16.58	21.96	7.75	46.52	11.07

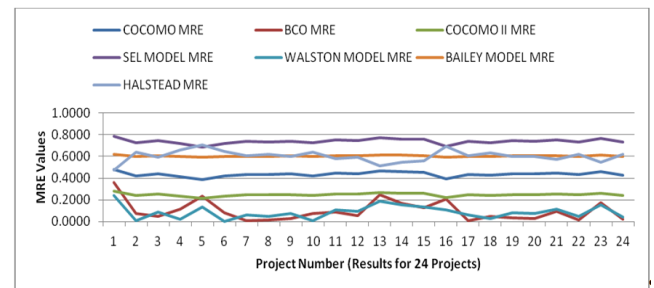
**Table 3.** Comparison between MRE Values of proposed model and other models

Project No.	BCO	COC OMO	COC OMO II	SEL	Bailey Basil	Halstead
1.	0.36	0.48	0.28	0.78	0.62	0.47
2.	0.08	0.42	0.24	0.72	0.60	0.64
3.	0.05	0.44	0.25	0.74	0.60	0.59
4.	0.12	0.41	0.23	0.72	0.60	0.66
5.	0.24	0.39	0.22	0.69	0.59	0.70
6.	0.08	0.42	0.24	0.72	0.60	0.64
7.	0.01	0.43	0.25	0.74	0.60	0.61
8.	0.02	0.43	0.24	0.73	0.60	0.62
9.	0.03	0.44	0.25	0.74	0.60	0.60
10.	0.07	0.42	0.24	0.72	0.60	0.64
11.	0.09	0.45	0.26	0.75	0.61	0.58
12.	0.06	0.44	0.25	0.75	0.60	0.59

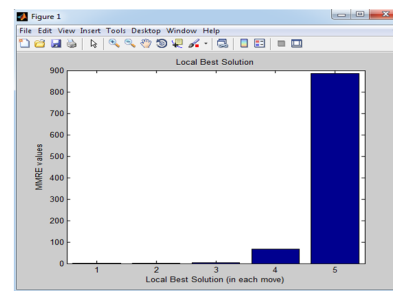
13.	0.24	0.47	0.27	0.77	0.61	0.52
14.	0.17	0.46	0.26	0.76	0.61	0.55
15.	0.13	0.45	0.26	0.76	0.61	0.56
16.	0.21	0.39	0.22	0.69	0.59	0.69
17.	0.01	0.43	0.25	0.74	0.60	0.61
18.	0.05	0.42	0.24	0.73	0.60	0.63
19.	0.04	0.44	0.25	0.74	0.60	0.60
20.	0.03	0.44	0.25	0.74	0.0	0.60
21.	0.09	0.45	0.26	0.75	0.61	0.58
22.	0.02	0.43	0.24	0.73	0.60	0.62
23.	0.18	0.46	0.26	0.76	0.61	0.54
24.	0.02	0.43	0.24	0.73	0.60	0.62

**Table 4.** Performance Analysis of different models base on MMRE values

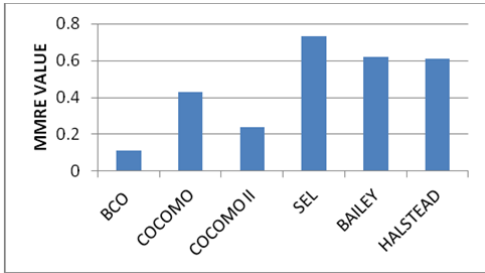
Models	MMRE Value	%MMRE
1. Proposed (BCO)	0.11	11
2. COCOMO	0.43	43
3. COCOMO II	0.24	24
4. SEL	0.73	73
5. Bailey	0.62	62
6. Halstead	0.61	61



**Figure 1.** Comparison of proposed method with different models based on MRE.



**Figure 2.** Solution set computed by proposed method (local best solutions).



**Figure 3.** Comparison of proposed model with COCOMO and other models.

Figure 1 shows comparison between proposed model with different models based on their MRE values for 24 projects and Figure 2 shows local best solutions found out by proposed model in every move among which minimum solution is chosen. Figure 3 shows comparison of MMRE values of proposed model with standard COCOMO and other existing models.

## 7. Conclusion

Software effort estimation is an inevitable and crucial process which is censorious for developers as well as for client. So it becomes very important to make estimates at early stages of development process. In this paper Bee Colony Optimization is used for estimating effort by using bee algorithm that is capable of making multi-agent system for solving complex optimization problems. The proposed model, obtained from COCOMO and KLOC, is validated using IVR dataset. Several partial solutions were evaluated from which the global best solution was carried out which is based on MMRE value. The optimized value of proposed model is then compared with COCOMO itself and other models and it was concluded that the proposed model yields better results in comparison to other models. The proposed model can be validated further by applying on different datasets.

## 8. References

- Gharehchopogh FS, Dizaji ZA. A new approach in Software Cost Estimation with Hybrid of BCO and Chaos Optimization Algorithm. *MAGNT Research Report*. 2014; 2(6):1263–71.
- Molokken K, Jorgensen M. A review of surveys on software effort estimation. *Proceedings of International Symposium on Empirical Software Engineering*; Rome: IEEE Computer Society; 2003. p.223–30.
- Shepperd M, Schofield C. Estimating software project effort using analogies. *IEEE Transactions on Software Engineering*. 1997; 23(11):736–43.
- Kaur M, Sehra SK. Particle Swarm Optimization based effort estimation using Function Point Analysis. *International Conference on Issues and Challenges in Intelligent Computing Techniques*; 2014. p.140–5.
- Sehra SK, Brar YS, Kaur N. Soft computing techniques for software project effort estimation. *International Journal of Advanced Computer and Mathematical Sciences*. 2011; 2(3):160–7.
- Gharehchopogh FS, Ebrahimi L, Maleki I, Gourabi SJ. A Novel PSO based approach with hybrid of Fuzzy C-means and learning automata in software cost estimation. *Indian Journal of Science and Technology*. 2014; 7(6):795–803.
- Chulani SD. Bayesian analysis of software cost and quality models. 1999.
- Kaur A, Goyal S. A survey on the applications of Bee Colony Optimization Techniques. *International Journal on Computer Science and Engineering*. 2011; 3(8):3037–46.
- Teodorovic D. Bee Colony Optimization (BCO). In: Lim CP, et al., editors. *Innovations in Swarm Intelligence*. 2009; 39–60.
- Nikolic M, Teodorovic D. Empirical study of the Bee Colony Optimization (BCO) algorithm. *Expert Syst Appl*. 2013; 40:4609–20.
- Seeley TD. *The wisdom of the hive: The social physiology of honey bee colonies*. Cambridge: Massachusetts Harvard University Press; 1996.
- Pertiwi AP, Suyanto. *Globally Evolved Dynamic Bee Colony Optimization*. Berlin Heidelberg: Springer-Verlag; 2011. p. 52–61.
- Rao KS, Reddy LSS, Devarakonda N. Efficient software cost estimation using partitioning techniques. *Int J Eng Res Tech*. 2013; 2(11):3899–932.
- Khuttan A, Kumar A, Singh A. a survey of effort estimation techniques for the software development. *Int J Eng Res Tech*. 2014; 3(7):234–6.
- Frisch KV. *Bees: Their vision, chemical senses and language*. New York, Ithaca: Cornell University Press; 1976.
- Srivastava DK, Chauhan DS, Singh R. VRS Model: A model for estimation of efforts and time duration in development of IVR software system. *IJSE*. 2012; 5(1):27–40.
- Boehm B, Clark B, Horowitz E, Westland C. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering Special Volume on Software Process and Product Measurement*. 1995; 1(1):57–94.
- Dizaji ZA, Gharehchopogh FS. A hybrid ant colony optimization and chaos optimization algorithms approach for software cost estimation. *Indian Journal of Science and Technology*. 2015; 8(2):128–33.
- Sarvzadeh R, Kashani MH, Zakeri FS, Jameii SM. A novel bee colony approach to distributed systems scheduling. *International Journal of Computer Applications*. 2012; 42(10):1–6.
- Benediktsson O, Dalcher D. Effort estimation in incremental software development. *IEEE Proceedings of Software*. 2013; 150(6):351–7.
- Chong CS, Low MYH, Kumar AS, Gay KL. Using a Bee Colony Algorithm for neighborhood search in job shop scheduling problems. *IEEE International Conference*; 2009.