International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 4 (2017), pp. 525-535

http://www.ripublication.com

Optimization of COCOMO Parameters using TLBO Algorithm

Sumeet Kaur Sehra
Research Scholar, I.K.G. Punjab Technical University, Jalandhar, Punjab, India.

Yadwinder Singh Brar
Professor, I.K.G. Punjab Technical University, Jalandhar, Punjab, India.

Navdeep Kaur
Associate Professor, Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India.

Gagandeep Kaur
Student, Guru Nanak Dev Engineering College, Ludhiana, Punjab, India.

Abstract

Software effort estimation assumes a basic part in software engineering as the success of a project completely relies on it. More recently, various Meta-heuristic methods have been implemented in SEE models. In this paper, a model using Teaching-learning based optimization (TLBO) has been proposed to optimize the parameters of the COCOMO model for providing better effort estimates. The proposed model validation is carried out using IVR dataset of a company. The results obtained show that proposed TLBO based model is able to improve the accuracy of cost estimation and also outperforms other models.

Keywords: Software effort estimation, TLBO, COCOMO, MMRE

INTRODUCTION

Despite the fact that interest of software framework is expanding in all parts of human life, development of software undertakings is hysterically noted with delays, high cost, and errors. Inaccurate estimation of assets is central point of failure in software [1]. Software estimation draws colossal consideration from academicians and authorities. Software estimation is the instrument of anticipating cost, effort, and duration that are required to create software. Estimator frequently relies on upon different pragmatisms to create software estimations. Surpassed budgets, incomplete functions, low quality and partial consummation of the venture are a portion of the main considerations that results in underestimation or overestimation of the software effort [2, 3]. The process of software development effort estimation in which estimator foresee the measure of effort to keep up software are based on questionable, noisy input and deficient undertaking arrangement, budgets, pricing processes and investment [4]. Estimated effort acts as the input to project request and proposal, project planning, control, budget, progress monitoring and staff allocation. The software engineering group puts tremendous effort while structuring models with a specific end goal to solace estimators to give accurate cost appraisals to software projects.

In recent years, tremendous increase has been witnessed in the SEE models based upon meta-heuristic algorithms [5]. Evolutionary algorithms including differential evolution [6], Genetic algorithms [7] have been applied in SEE for tuning parameters of the estimation models. Swarm intelligence based algorithms have also found use in SEE models including particle swarm optimization [8, 9], bee colony optimization [10] and ant colony optimization [11]. In this paper, Teacher-learner based algorithm (TLBO) has been applied to optimize the parameters of COCOMO model. TLBO, an iterative learning algorithm has similar characteristics with contemporary evolutionary computation (EC) algorithms. Rao and Patel [12] have suggested that TLBO has been found to produce better results as compared to other EC techniques including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Artificial Bee Colony (ABC).The competition strategy characterizes the range of the main competition methods selected by the management of the enterprise: the price or non-price ones within the behaviour strategy. The selection of each of these strategies is associated with a certain degree of risk, arising, in particular, due to the inadequately evaluated capacity of competitors and own capacity, the competitive situation, the unpredictable changes in the economy, legal regulation, technology, ecology, society and politics [11].

The remaining paper is structured in different sections. Second section discusses different existing SEE models. In third section, related work on meta-heuristic algorithms is presented. Fourth section explains TLBO in detail. The proposed model is explained in fifth section and results are figured out in sixth section. In seventh section, conclusion and future scope are presented.

SOFTWARE EFFORT ESTIMATION MODELS

Algorithmic effort estimation techniques involve the use of mathematical formulas based on statistical data analysis. Some of the algorithmic estimation models include:

COCOMO-II Model: This model is an extension of COCOMO Intermediate model and effort is calculated as in equation 1.

$$\text{Effort} = 2.9 * (\text{KLOC})^{1.10} \tag{1}$$

Bailey-Basili Model: The relationship between effort and lines of code is suggested by Bailey-Basili and is represented by equation 2.

$$\text{Effort} = 5.5 * (\text{KLOC})^{1.16} \tag{2}$$

Software Engineering Laboratory (SEL) Model: This model has been suggested by Software Engineering Laboratory (SEL) of the Maryland University. According to SEL model, the effort estimate is evaluated as in equation 3.

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

Halstead Model: In Halstead model, relationship between effort and KLOC is represented as in equation 4.

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

RELATED WORK

Toka and Turetkan [13] have discussed the accuracy comparison of different parametric software estimation models. They have utilized 51 genuine projects for evaluating the accuracy of estimates in terms of MMRE and suggested for a significant scope for improvement in effort estimates. Sarro [14] has discussed the use of meta-heuristic techniques for SEE and has concluded that these are able to enhance the effectiveness of estimation methods. Genetic algorithms have been implemented for developing effort estimation models by [15, 16, 17]. Ferrucci et al [18] have reported the use of Tabu Search on Desharnais dataset for SEE and concluded that Tabu Search provides better estimates. Aljahdali and Sheta [6] have developed a model based upon differential evolution COCOMO-DE and the validation of the performance of the developed model was performed using NASA dataset. The proposed model was capable to provide more accurate estimates. Chalotra et al [10] have implemented Bee Colony Optimization (BCO), a metaheuristic way to optimize the parameters of COCOMO. The results obtained show that the proposed BCO based model is able to improve the accuracy of cost estimation and also outperform other models. Chen and Zhang [11] have proposed an approach using ant colony optimization (ACO) algorithm for an event-based scheduler (EBS), which enables resource conflict modelling and task pre-emption. Dewan and Sehra [19] have utilized ACO to optimize the parameters of the organic mode of Basic COCOMO model using ACO technique and have concluded that proposed model performed better than the COCOMO model. PSO has been implemented by [8, 9, 20] for modification of the COCOMO parameters and it has been concluded that the resultant techniques significantly increase the accuracy of the COCOMO approach and also incorporate the much needed flexibility related to the software.

Teaching-Learning Based optimization (TLBO) Algorithm

Rao et al [21] have proposed a Teaching-Learning Based Optimization (TLBO) inspired by the teaching-learning process in the classroom. It is a population-based efficient algorithm that models the concept of transferring knowledge in the classroom. TLBO is called parameter-less optimization algorithm as there is no need of any specific parameters [12]. The algorithm simulates two vital means of learning: (a) through the teacher (teacher phase) and (b) interacting with different learners (learner phase). In TLBO algorithm, group of students (i.e. learner) is viewed as the population and the diverse subjects offered to the learners are related to unique design variables of the optimization problem. Fitness value of the optimization problem is equivalent to the results obtained by the learner.

Teacher Phase: In this phase, learners are created by using a randomly ordered state of points from population or search space. From these learners, a point is assumed as a teacher who transfers knowledge to the other learners in the population. The mean of the class depends on the capability of the teacher. If a teacher is a good teacher, then there is an increase
e in the mean of class from M_A to M_B. But this does not solely depend on upon the capability of the teacher, it depends upon the capability of class as well.

Let T_i be the point considered as teacher and M_i is the corresponding mean. M_{new} will be the new mean when Ti will try to move mean Mi towards its own level. The solution is generated as the difference between the two mean values is calculated as in equation 5.

$$Difference\_mean = r_i\,(M_{new} - TF * M_i) \tag{5}$$

where TF (teaching factor) is the deciding factor and ri is a random number in the range of [0,1]. The value of TF can be either 1 or 2, decided randomly with equal probability [22] and is represented as in equation 6.

$$TF = round[1 + rand(0,1)2 - 1] \tag{6}$$

New Solution is evaluated as shown in equation 7.

$$X_{new} = X_{old,i} + Difference\_mean_i \tag{7}$$

Learner phase: In this phase, interaction among learners is the key to knowledge improvement. A learner interacts randomly with other learners in the group by different modes of communication. A learner will improve his knowledge if and only if another learner knows something better than him or her. Learner knowledge improvement is represented by equations 8 and 9.

$$X_{new,i} = X_{old,i} + r_i(X_i - X_j), iff(X_i) < f(X_j) \tag{8}$$

$$X_{new,i} = X_{old,i} + r_i(X_j - X_i), iff(X_j) < f(X_i) \tag{9}$$

The steps of implementing the TLBO algorithm can be summarized as suggested by [21, 23].

Step 1. Problem is outlined and required parameters are initialized.

Step 2. Population or search space is initialized.

Step 3. Learners enhance their knowledge by interacting with the teacher in teacher phase.

Step 4. Learners further enhance their knowledge through the interaction with other learners.

Step 5. Stop criterion is evaluated and the algorithm is terminated when stop criteria is justified otherwise return to step 3.

PROPOSED METHODOLOGY

In this work, TLBO algorithm is implemented to improve effort estimation of COCOMO model. To improve effort estimation of COCOMO model, the output of COCOMO model in terms of MRE value can be given as input to TLBO algorithm. The TLBO algorithm will calculate the best value of MRE by executing various iteration using teaching rules and iteration at which MRE value is minimum can be considered as best MRE value of the project. The TLBO algorithm is the algorithm which is used for the optimization. In this research TLBO algorithm is applied to reduce MRE value of the COCOMO model by estimating predicted efforts more accurately. The steps of the proposed algorithm are as follows. Figure 1 depicts the flowchart of proposed methodology.

1. Input the value of MRE of each project which is calculated using COCOMO model.
2. Calculate the best value of MRE for each project for the learner phase.
3. Calculate the best teacher phase value of each project.
4. Select the best MRE value of each phase by comparing it with the last phase.
5. Steps 2, 3 and 4 are repeated until MRE value of each project is calculated.

Evaluation Criteria

Magnitude of Relative Error (MRE): MRE is used as a performance measure to evaluate the performance of the proposed model and is evaluated by using equation 10.

$$\text{MRE} = \frac{\text{actual effort - predicted effort}}{\text{actual effort}} \tag{10}$$

Mean Magnitude of Relative Error (MMRE): The performance of the model is evaluated using MMRE which is calculated by using equation 11.

$$MMRE = \frac{1}{N}\sum_{i=1}^{N}\frac{\text{actual effort- predicted effort}}{\text{actual effort}} \quad (11)$$

where actual effort is taken from the data set and predicted effort is from proposed model.

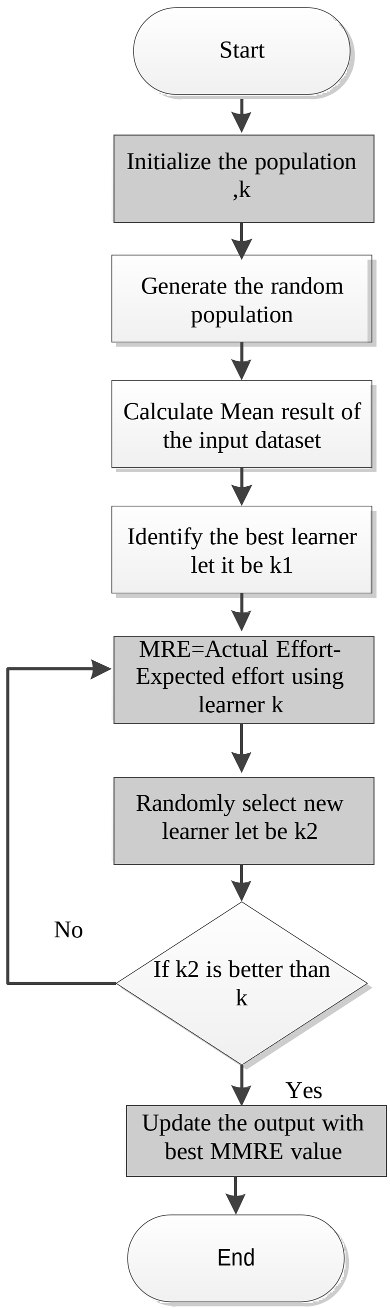

Figure 1: Flowchart of proposed methodology

RESULTS

The validation of proposed model is performed by using a dataset of Interactive Voice Response (IVR) applications from software industry [24] as shown in Table 1. Various algorithmic models are implemented on the data set using equations 1-4. The performance of these algorithmic models including COCOMO-II, Bailey model, Halstead Model, BCO, and SEL is compared with proposed technique of TLBO algorithm for effort estimation. The MMRE of proposed model using TLBO is comparatively less than existing SEE models as shown in Figure 2. The values of MRE for different models is discussed in Table 1.

Table 1: MRE evaluation for different models

Project No.	Size(KLOC)	Bailey-Basili	COCOMO -II	Halstead	SEL	BCO	TLBO
1	16.2	0.8987	0.8860	0.8967	0.8405	0.1681	0.1933
2	5.34	0.8940	0.8689	0.8940	0.8346	0.1669	0.1920
3	7.6	0.8845	0.8583	0.8845	0.8228	0.1646	0.1892
4	4.7	0.8700	0.8423	0.8700	0.8052	0.1610	0.1852
5	3.1	0.8553	0.8262	0.8553	0.7877	0 .1575	0.1812
6	5.2	0.8100	0.7776	0.8100	0.7359	0.1472	0.1693
7	6.8	0.7997	0.7667	0.7997	0.7245	0.1449	0.1666
8	6.4	0.7946	0.7613	0.7946	0.7188	0.1438	0.1653
9	7.2	0.7946	0.7613	0.7946	0.7188	0.1438	0.1653
10	5.4	0.7894	0.7558	0.7894	0.7132	0.1426	0.1640
11	8.5	0.7842	0.7504	0.7842	0.7075	0.1415	0.1627
12	7.8	0.7632	0.7285	0.7632	0.6849	0.1370	0.1575
13	12.5	0.7558	0.7208	0.7558	0.6770	0.1354	0.1557
14	10.4	0.7526	0.7175	0.7526	0.6736	0.1347	0.1549
15	9.5	0.7472	0.7120	0.7472	0.6680	0.1336	0.1536
16	3.4	0.7365	0.7010	0.7365	0.6568	0.1314	0.1511
17	6.8	0.7312	0.6955	0.7312	0.6512	0.1302	0.1498
18	5.8	0.7096	0.6734	0.7096	0.6289	0.1258	0.1446
19	7.4	0.7096	0.6734	0.7096	0.6289	0.1258	0.1446
20	7.2	0.7041	0.6679	0.7041	0.6233	0.1247	0.1434
21	8.6	0.6987	0.6624	0.6987	0.6178	0.1236	0.1421
22	6.4	0.6987	0.6624	0.6987	0.6178	0.1236	0.1421
23	10.6	0.6987	0.6624	0.6987	0.6178	0.1236	0.1421
24	6.3	0.6877	0.6513	0.6877	0.6067	0.1213	0.1395
25	4.5	0.6822	0.6457	0.6822	0.6011	0.1202	0.1383
26	9.7	0.6767	0.6402	0.6767	0.5956	0.1191	0.1370
27	8.4	0.6767	0.6402	0.6767	0.5956	0.1191	0.1370

28	6.2	0.6545	0.6179	0.6545	0.5735	0.1147	0.1319
29	8.5	0.6545	0.6179	0.6545	0.5735	0.1147	0.1319
30	6.2	0.6434	0.6068	0.6434	0.5625	0.1125	0.1294
31	2.6	0.6434	0.6068	0.6434	0.5625	0.1125	0.1294
32	2.5	0.6322	0.5956	0.6322	0.5515	0.1103	0.1268
33	4.3	0.6209	0.5844	0.6209	0.5405	0.1081	0.1243
34	4.6	0.5869	0.5508	0.5869	0.5078	0.1016	0.1168
35	6.6	0.5869	0.5508	0.5869	0.5078	0.1016	0.1168
36	7.4	0.5812	0.5452	0.5812	0.5023	0.1005	0.1155
37	4.6	0.5812	0.5452	0.5812	0.5023	0.1005	0.1155
38	8.6	0.5755	0.5395	0.5755	0.4969	0.0994	0.1143
39	5.5	0.5755	0.5395	0.5755	0.4969	0.0994	0.1143
40	4.8	0.5467	0.5114	0.5467	0.4697	0.0939	0.1080
41	18	0.5235	0.4888	0.5235	0.4481	0.0896	0.1031
42	12.5	0.5119	0.4775	0.5119	0.4373	0.0875	0.1006
43	6.7	0.4708	0.4379	0.4708	0.3996	0.0799	0.0919
44	8.4	0.4589	0.4265	0.4589	0.3889	0.0778	0.0894
45	5.7	0.3449	0.3181	0.3449	0.2876	0.0575	0.0661
46	2.8	0.3449	0.3181	0.3449	0.2876	0.0575	0.0661
47	6.4	0.1150	0.1047	0.1150	0.0933	0.0187	0.0215
48	9.1	0.7467	0.7129	0.7472	0.6680	0.1336	0.1526

From the table, it is evident that value of MRE for proposed model is less than existing algorithmic models. Also from Table 2 the value of MMRE for proposed model is 0.23 and it has been widely accepted that if the MMRE is at most 0.25 then a software effort estimation model is considered acceptable. Figure 2 illustrates the comparison for MMRE between the proposed and existing algorithmic models.

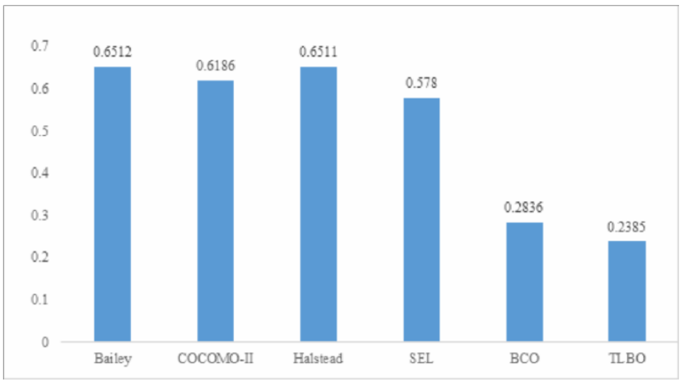

Figure 2: MMRE comparison of proposed model with existing models

Table 2: Comparison of MMRE of different models

Model	MMRE
Bailey-Basili	.6512
COCOMO-II	.6186
Halstead	.6511
SEL	.578
BCO	.2836
TLBO	.2385

CONCLUSION

Accurately estimating the effort required to develop a software has always been a concern for industry and academia. TLBO algorithm has proved to be an efficient optimization algorithm that can be implemented on very few parameters. In this paper, a model for SEE has been proposed using TLBO algorithm to provide better estimates. The performance of the developed model was validated using 48 projects of IVR dataset presented in [24]. The comparison of the performance of different models based on MRE and MMRE resulted in the better performance of proposed model. The value of MMRE for proposed model was 0.23 comparatively less than existing state of the art SEE algorithmic models. The proposed model was able to outperform the existing models in estimating the effort more accurately. In future, TLBO can be used with other meta-heuristic techniques to develop a hybrid model for SEE and more datasets can be used to validate the model.

REFERENCES

[1] R. N. Charette, "Why software fails [software failure]," *IEEE Spectrum*, vol. 42, pp. 42–49, Sept. 2005.

[2] S. Basha and D. P, "Analysis of empirical software effort estimation models," *International Journal of Computer Science & Information Security*, vol. 7, no. 3, pp. 68–77, 2010.

[3] S. Chulani, "Bayesian analysis of software cost and quality models," in *Proc. IEEE Int Software Maintenance Conf*, pp. 565–568, 2001.

[4] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Transactions on Software Engineering*, vol. 33, pp. 33– 53, Jan. 2007.

[5] S. K. Sehra, Y. S. Brar, and N. Kaur, "Soft Computing Techniques for Software Project Effort Estimation," *International Journal of Advanced Computer and Mathematical Sciences*, pp. 160–167, 2011.

[6] S. Aljahdali and A. F. Sheta, "Software effort estimation by tuning COOCMO model parameters using differential evolution," in *ACS/IEEE International*

Conference on Computer Systems and Applications - AICCSA 2010, pp. 1–6, May 2010.

[7] A. F. Sheta, "Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects," *Journal of Computer Science*, vol. 2, no. 2, pp. 118–123, 2006.

[8] M. Kaur and S. K. Sehra, "Particle swarm optimization based effort estimation using Function Point analysis," in *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*, pp. 140–145, Feb. 2014.

[9] Prasad Reddy P.V.G.D, Hari CH.V.M.K., and Srinivasa Rao T., "Multi Objective Particle Swarm Optimization for Software Cost Estimation," *International Journal of Computer Applications (0975 – 8887)*, vol. 32, pp. 13–17, Oct. 2011.

[10] S. Chalotra, S. Sehra, Y. Brar, and N. Kaur, "Tuning of COCOMO model parameters by using bee colony optimization," *Indian Journal of Science and Technology*, vol. 8, no. 14, 2015.

[11] W.-N. Chen and J. Zhang, "Ant Colony Optimization for Software Project Scheduling and Staffing with an EventBased Scheduler," *IEEE Transactions on Software Engineering*, vol. 39, no. 1, pp. 1–17, 2013.

[12] R. V. Rao and V. Patel, "An improved teachinglearning-based optimization algorithm for solving unconstrained optimization problems," *Scientia Iranica*, vol. 20, no. 3, pp 710 – 720, 2013.

[13] D. Toka and O. Turetken, *Accuracy of Contemporary Parametric Software Estimation Models: A Comparative Analysis*.

[14] F. Sarro, "Search-based Approaches for Software Development Effort Estimation," in *Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement*, Profes '11, (New York, NY, USA), pp. 38–43, ACM, 2011.

[15] A. L. I. Oliveira, P. L. Braga, R. M. F. Lima, and M. L. Corn´elio, "GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation," *Information and Software Technology*, vol. 52, no. 11, pp. 1155 – 1166, 2010. <ce:title>Special Section on Best Papers {PROMISE} 2009</ce:title>.

[16] S.-J. Huang and N.-H. Chiu, "Optimization of analogy weights by genetic algorithm for software effort estimation," *Information and Software Technology*, vol. 48, no. 11, pp. 1034 – 1045, 2006.

[17] C. J. Burgess and M. Lefley, "Can genetic programming improve software effort estimation? A comparative evaluation," *Information and Software Technology*, vol. 43, no. 14, pp. 863 – 873, 2001.

[18] F. Ferrucci, C. Gravino, R. Oliveto, and F. Sarro, "Using Tabu Search to Estimate Software Development Effort," in *Software Process and Product Measurement: International Conferences IWSM 2009 and Mensura 2009 Amsterdam, The Netherlands, November 4-6, 2009.*

[19] N. Dewan and S. K. Sehra, "Ant Colony Optimization Based Software Effort Estimation," *International Journal of Computer Science And Technology*, vol. 5, pp. 53– 56, July 2014.

[20] P. Bhattacharya, P. R. Srivastava, and B. Prasad, "Software Test Effort Estimation Using Particle Swarm Optimization," in *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012) held in Visakhapatnam, India, January 2012* pp. 827–835,

[21] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Computer-Aided Design*, vol. 43, no. 3, pp. 303 – 315, 2011.

[22] S. C. Satapathy and A. Naik, "Modified Teaching–Learning-Based Optimization algorithm for global numerical optimization—A comparative study," Swarm and Evolutionary Computation, vol. 16, pp. 28 – 37, 2014.

[23] F. Zou, L. Wang, X. Hei, D. Chen, and B. Wang, "Multiobjective optimization using teaching-learning-based optimization algorithm," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1291 – 1300, 2013.

[24] D. K. Srivastava, D. S. Chauhan, and R. Singh, "VRS Model: A Model for Estimation of Efforts and Time Duration in Development of IVR Software System," *Int. J. of Software Engineering, IJSE*, vol. 5, no. 1, 2012.