A Novel Technique of Software Cost Estimation Using Flower Pollination Algorithm

Aman Ullah, Bin Wang
School of Computer Science and Engineering,
Central South University, China
dr.aman@csu.edu.cn,
wb_csut@cus.edu.cn

Jinfan Sheng, Jun Long
School of Computer Science and Engineering,
Central South University, China
jfsheng@csu.edu.cn,
jlong@cus.edu.cn

Muhammad Asim, Faiza Riaz
School of Computer Science and Engineering,
Central South University, China
asimpk@csu.edu.cn,
faiza123@cus.edu.cn

Abstract— **Estimating Cost of Software (ECS) is a very essential aspect of a software development life cycle. Accurate estimations are needed in terms of person month and development time for software projects. Several project estimation techniques have been developed such as parametric and non-parametric. Estimation by parametric is one of the convenient methods in the field of software engineering. However, parametric approaches used in the development of software cost estimation is not able to handle the definite data in an explicit and accurate way. Different computational intelligence algorithms like machine learning, evolutionary algorithms, and swarm intelligence algorithms have been applied to optimize the parameters of various ECS model. But however, accurate cost estimation is still a big issue in ECS. In this work, a Flower Pollination Algorithm (FPA) is proposed to optimize the parameters of a Constructive Cost Model II (COCOMO-II) via using a standard Turkish industry dataset. Experimental results demonstrate that the proposed algorithm gives a better estimation as compared to existing approaches such as the Bat algorithm and original COCOMO-II in terms of Manhattan distance (MD) and mean magnitude of relative errors (MMRE).**

Keywords— Estimating Cost of Software (ECS), COCOMO-II, Computational Intelligence Algorithms, Turkish Industry software project.

I. INTRODUCTION

Estimating Cost of Software is an extremely significant activity in the development of any software project. The failure and success of a software projects is based on reliable cost estimation [1, 2]. The estimation of the cost of any software might be the difficult phase of the whole process due to the software indeterminate nature. Also, the software product may change while the development process can have unexpected alterations in the estimations as well. Underestimating and overestimating the cost have a great influence on the development of the project and can inherit critical issues in the development process. Therefore, it is required to find out the better cost estimation technique for the organization to develop a better software product [3].

Consequently, there have been a lot of research efforts made in the estimation concern of software and numerous methods have been introduced such as non-parametric or non-algorithmic and algorithmic. Some common non-algorithmic techniques are an expert judgment, top-down approach, price–to-win, bottom-up and so on. Some used techniques of the or algorithmic models are Constructive Cost Model (COCOMO), Walston-Felix, Doty, Software Engineering Laboratory (SEL), and Halsted. The prevalent and extensively used model for the estimation of cost is a Constrictive Cost Model-II (COCOMO-II). COCOMO-II uses cost or effort multipliers, scale factors and software size i.e. line of code to predict the effort and time development for any software project. However, precise estimation in terms of man month and time development is still a big challenging issue in COCOMO-II. Therefore, researchers started applying different computational intelligence techniques like genetic algorithm, particle swarm, ant-colony optimization and firefly algorithm [4-8] to optimize the coefficients of the COCOMO-II for better estimation of projects.

In this study, we propose the utilization of applying the Flower Pollination Algorithm (FPA) for optimizing the current coefficients of COCOMO-II technique using a stranded Turkish industry software project dataset.

The reminder organization of this paper is as follows: Section II re-presents the related work of cost estimation using various optimization algorithms. Section III describes the COCOMO-II Model. In section IV we discusses the proposed methodology. Section V presents the experimental work and analysis. Finally,in section VI, the conclusion along with future recommendation is presented.

II. RELATED WORK

Many related studies are devoted to an optimization of cost estimation models, such as the optimization of COCOMO-81 and COCMO-II model parameters. For example, in [1, 2], PSO and Bat algorithms were proposed for the optimization of COCOMO-II parameter's via using the Turkish industry dataset. According to the authors the proposed methods have better estimation efforts as compared to the original COCOMO-II.

In [9], Sheta et al. proposed Soft-Computing(SC) approaches like COCOMO tuned by PSO and fuzzy-logic and compared with conventional techniques such as Constructive Cost Model (COCOMO), Walston-Felix, Doty, Bailey-Basili and Halsted models. Experimental analysis show that the proposed method outperforms traditional cost estimation technique for NASA software projects. Also, Saif et al. [10] presented an intelligent water drops algorithm for the optimization of the COCOMO-II parameters. This technique produces more precise effort estimation in the term of time and effort for the development of software project on testing NASA-93 dataset. In [11], the cuckoo search algorithm was used to find the best parameters for the software cost

0-7695-6728-2/19/$31.00 ©2019 IEEE
DOI 10.1109/ICICAS48597.2019.00142

estimation model on testing the NASA dataset. Experimental analysis demonstrate that the suggested model improves the performance effort in term of Prediction (PRED) and MMRE. Moreover, the hybrid technique of Artificial Neural Network(ANN) and Cuckoo Search(CS) was proposed in [12], for optimizing the parameters selection COCOMO-II. Numerical results reveal that the proposed hybrid technique performs better than cuckoo search and ANN.

Khuat et al. [13], proposed a hybrid algorithm by combining bat algorithm with Gravitation Search Algorithm (GSA) to optimize the parameters COCOMO model on testing NASA dataset projects. Also, in [14], the authors proposed teaching learning algorithm for the optimization of COCOCMO-II parameters. The simulation result achieved using proposed method is better than the original COCOCMO-II model results.

Our study mainly focuses on the optimization of the COCOMO-II parameters by implementing FPA on standard datasets taken from Turkish Industries software project. The proposed technique is compared with COCOMO-II and bat algorithm. Simulation results show that the proposed technique outperforms COCOMO-II and bat in terms of MMRE and Manhattan Distance (MD).

III. Proposed Methodlogy

In this section, we present a Flower Pollination Algorithm (FPA) for optimization of the COCOMO-II parameters. The COCOMO-II and FPA are explained in subsections A & B respectively.

A. COCOMO-II Model

COCOMO-II model [15] was introduced in 1995 for the software project estimation cost or effort and time needed for the development of any software project. It takes software size in term of Thousand Source Line of Code (SLOC) as an input and produces an output i.e. effort which is represented in the form of Man Month (MM). "*A man month(MM) is actually the amount of time one man spend working on the development of software project for one month* "[16].

COCOMO-II consists of further three sub models namely, application composition, post-architecture (PA) and early design [16]. This work only studies the PA method. According to this method the software project development effort MM is computed using the following (1).

$$MM = A \times Size^{E} \times \prod_{i=1}^{17} EMi \quad (1)$$

Where

$$E = B + 0.01 \times \sum_{j=1}^{5} SFj \quad (2)$$

Where A=2.94 and B=0.91 are multiplicative constants. Size represents the actual size of software in terms of KSLOC, EM means effort multipliers and SF represents scale factor. There are seventeen EM rating from very-low to extra- high based on COCOMO-II post architecture method as shown in Table I. The five scales factor rating level ranges from low to extra-high as show in Table II.

Table I. Effort Multipliers For COCOMO-II Model [15]

Drivers	Very-Low(VL)	Low (L)	Nominal (N)	High (H)	Very-High(VH)	Extra-High(X)
RELY	0.820	0.920	1.0000	1.100	1.260	---
DATA	----	0.900	1.0000	1.140	1.280	---
CPLX	0.730	0.870	1.000	1.170	1.340	1.740
RUSE	----	0.950	1.000	1.070	1.150	1.240
DOCU	0.810	0.910	1.000	1.110	1.230	--
TIME	----	----	1.000	1.110	1.290	1.630
STOR	----	----	1.000	1.050	1.170	1.460
PVOL	----	0.870	1.000	1.150	1.300	---
ACAP	1.420	1.190	1.000	1.850	0.710	---
PCAP	1.340	1.150	1.000	0.880	0.760	---
PCON	1.290	1.120	1.000	0.900	0.810	---
APEX	1.220	1.100	1.000	0.880	0.810	---
PLEX	1.190	1.090	1.000	0.910	0.850	---
LTEX	1.200	1.090	1.000	0.910	0.840	---
TOOL	1.170	1.090	1.000	0.900	0.780	---
SITE	1.220	1.090	1.000	0.930	0.860	0.800
SCED	1.430	1.140	1.000	1.000	1.000	---

Table II. Scale Factors For COCOMO-II Model [15]

Scale-Factors	Very-Low (VL)	Low (L)	Nominal(N)	High (H)	Very – High (VH)	Extra-High (XH)
PREC	6.200	4.960	3.720	2.480	1.240	0.000
FLEX	5.070	4.050	3.040	2.030	1.010	0.000
RESL	7.070	5.650	4.240	2.830	1.410	0.000
TEAM	5.480	4.380	3.290	2.190	1.100	0.000
PMAT	7.800	6.240	4.680	3.120	1.560	0.000

For software cost estimation using COCOMO-II, there are some uncertainties like the current parameters A and B of COCOMO-II. These parameters are not yet tuned in an efficient way, resulting an inaccurate cost estimation for projects. Therefore, researchers tried to tune these parameters for better cost estimation. In this research work, we propose an algorithm named a flower pollination algorithm to optimize the existing parameters of COCOMO-II model.

B. Proposed Flower Pollination Agorithm (FPA)

Flower pollination algorithm (FPA) [17, 18] was introduced in 2012. It is a recent nature-inspired algorithm, inspired by insemination (pollination) flower process. The idealized rules that FPA follows are summarized as [19]:

1) Global pollination phase is accomplished by conducting cross-pollination among pollinators like birds. The pollinators act based on the behavior of levy flight.

2) Self-pollination taking place on nearby flowers is termed as a local pollination process.

3) The reproductive rate is defined in terms of the consistency of the flower and has a direct ratio with a similar level of two involved flowers.

4) Both global and local pollinations are executed based on a switch probability.

Global and local pollination are two operators used in FPA. Each pollen item is measured as a solution X_i in FPA. The solutions are generated based on random vectors in the feasible search space.

Algorithm 1 Pseudo Code of FPA

```
Fitness function f(X); X = x_1, x_2, , , , , x_d
Initialize population randomly with n flowers
Evaluate initial population to find best solution G_*
Choose pϵ [0, 1]; where p is a switch probability
while (t < MaxGeneration) do
  for i=1:n (all n flowers in the population) do
    if rand < p, then
      Compute vector L through Levy flight
      Update global pollination
    else
      Compute εϵ[0, 1] through uniform distribution
      Select two solutions j and k randomly
      Invoke local pollination
    end if
    Evaluate new solutions in population through Fitness function f(X)
    If new solutions are better based on fitness value, update solutions
  end for
  Find the current best solution G_*
end while
```

The global pollination phase (rule-i) and flower constancy (rule-iii) can be mathematically formulated as:

$$X_i^{t+1} = X_i^t + \gamma L_{(\lambda)}(G_* - X_i^t) \quad (3)$$

Where X_i^t represents the pollen i at the iteration t and G_* denotes the recent global best solution. γ is the scale factor which control a size of step. $L_{(\gamma)}$ is a flight feature of bird that matches to the strength of the pollination. The levy distribution when $L > O$ can be denoted as follows:

$$L \sim \frac{\gamma\tau(\lambda)\sin(\frac{\pi\lambda}{2})}{\pi} \frac{1}{S^{1+\lambda}} \quad (S \gg S_0 > 0). \quad (4)$$

Where $\tau(\lambda)$ is a standard gamma function. For the local pollination, both (rule ii) and the constancy of flower can be described as follows:

$$X_i^{t+1} = X_i^t + \epsilon(X_k^t - X_k^t) \quad (5)$$

Here X_j^t and X_k^t are pollens, which are selected randomly from different flowers of the similar plant classes. Here j and $k \in \{1, \dots \dots \dots . NP\}$, $\in$ is the dimensional random vector [0,1]. In addition, from rule iv, the two pollination classes occur randomly and are determined by a probability P. For example if a random value rand [0,1] in is lower than P, then global pollination is accompanied, otherwise, vice versa. The pseudo code of FPA is given in algorithm 1.

IV. Experiments and Resluts Analysis

A. Evalution Criteria and Dataset Decription

For evaluation purpose, we use MRE, MMRE and MD as the objective function for the proposed technique. MRE can be calculated as:

$$MRE = \left|\frac{Actual\ Effort - Estimated\ Effort}{Actual\ Effirt}\right| \times 100 \quad (6)$$

The MMRE is the absolute average value of the MRE, divided by a total number of projects (N) and can be computed in (7).

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \quad (7)$$

Where 'N' is number of total projects. MD, which determines the absolute difference of actual effort and predicted effort and can be computed in (8).

$$MD = (\sum_{i=1}^{N} |\ actual\ Effort - Estimated\ Effort\ |) \quad (8)$$

The proposed (FPA) model is tasted on standard dataset taken from Turkish Industry (software projects), which consists of 12 software project, size in term of KSLOC, actual effort, five scale factor and seventeen effort multipliers [20].

B. Exerimentation and Results Analysis

The main goal of the proposed work is to optimizing the current coefficients A & B of the COCOMO-II model via using the FPA for batter cost estimation results. The experiments have been conducted on Turkish Industry software projects using MATLAB environment under the Windows 10 PC. The parameters setting of the FPA are set as: Number of independent run = 50, the maximum no. of iteration=100, the size of population=50 and probability switch=0.8.

The proposed optimized COCOMO-II using FPA constants A and B depend on the parameters that we set. The experimental work is based on trial and error. Sometimes it does not give batter results due to the searching behavior/characteristic of meta-heuristic algorithm for finding the optimal solutions. If the result obtained by FPA is better than the previous one we accept it otherwise we ignore it.

TABLE III. Comparison Among The COCOMO-II, Bat Algorithm And Proposed Model

Model	MMRE(Mean Magnitude Relative Error)	MD (Manhattan Distance)
COCOMO II	733.1436	585.9424
Bat Algorithm	34.2667	44.18909
Proposed FPA	**34.1957**	**43.2506**

In experiments based on trial and errors, we get A = 4.4051 and B = -0.1842 using the FPA, while in original COCOMO-II parameters values are A=2.94 & B=0.91. Table III shows comparison among the proposed method, original COCOMO-II, optimized COCOMO-II by Bat algorithm [1]. Figure 1. Illustrates the actual effort, COCOMO II, bat algorithm and FPA comparisons. In this graph it is clearly shown that using proposed FPA is batter and much closer to the actual cost when compared to cost estimated by original COCMO-II parameters and the Bat algorithm.

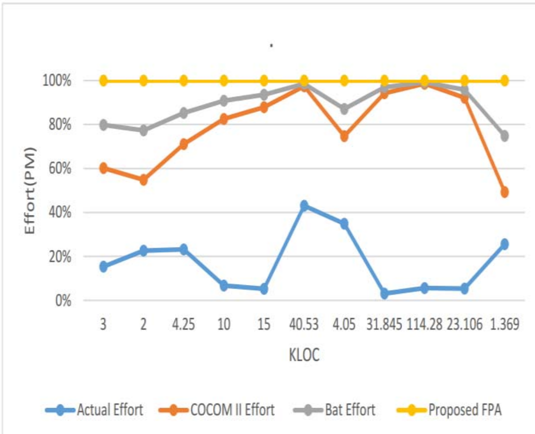

Fig 1. Comparison among the Actual Effort, COCOMO-II, & COCOMO-II optimized by Bat algorithm and Proposed FPA.

Based on these analysis, it observed that using the FPA optimization technique for COCOMO-II gave the higher estimated cost results as compared to conventional COCOMO-II in terms of MMRE and MD as mention in equations 6 and 7. The mean MRE of each technique shows the overall of precise measurements. The mean MRE results for COCOMO- II, Bat algorithm and FPA are 733.1436%, 34.2667%, and 34.1957% respectively, means that (FPA) proposed model are able to minimize errors efficiently.

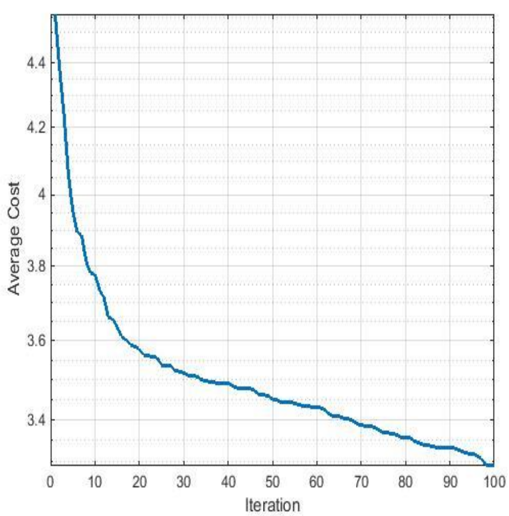

Fig 2. The average MMRE convergence of the Propose FPA model.

The comparison graph of MD and MMRE indicate that cost estimation provided by the proposed FPA model is better than the original COCOMO-II model and Bat algorithms as shown in figure 4. Figure 2 illustrates the average convergence of the proposed model. The graph of optimized coefficients (using FPA) A and B are shown in figure 3.

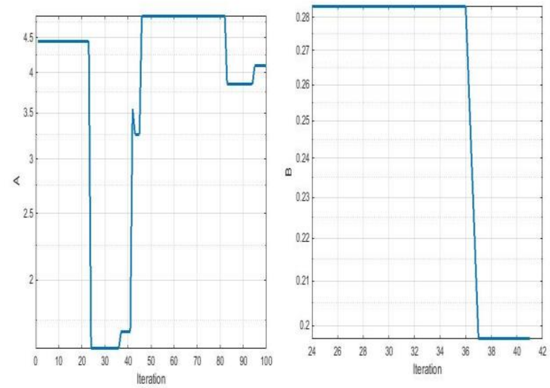

Fig 3. The best convergence of proposed FPA Parameters A & B.

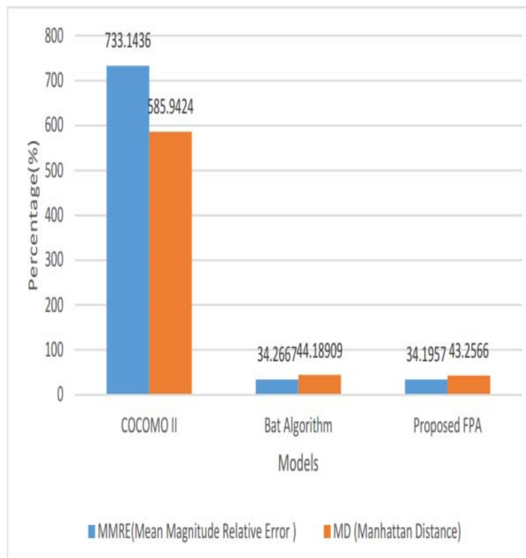

Fig 4. MMRE and MD comparison of proposed FPA, COCOMO-II and COCOMO-II optimized by Bat algorithm.

V. CONCLUSION AND FUTURE WORK

Eestimation of Software project cost is a significant part in field of software engineering. For software projects, accurate estimation can help project developer or managers in planning before starting the developments of any software projects.

In this study, we proposed FPA for optimization of COCOMO-II model parameters using standard dataset (Turkish Industry software project). The proposed method efficiently optimizes the coefficient of COCOMO-II like A & B for better cost estimation by considering MMRE and MD as a fitness function. Thus, A and B produced by FPA can be assigned in the equation used for cost estimation. By experimental results analysis show that the proposed (FPA) technique outperforms the original COCOMO-II and COCOMO-II optimized by Bat algorithm in terms of MMRE and MD.

In the future, we plan to apply the FPA for Software quality estimation model. Along with this, we also plan to propose some other nature-inspired algorithms for the optimization of various cost estimation models.

ACKNOWLEDGMENT

The research was jointly supported by the National Science and Technology Major Project of China (2017ZX06002005).

REFERENCES

[1] Y. A. Effendi, R. Sarno, and J. Prasetyo, "Implementation of Bat Algorithm for COCOMO II Optimization," in 2018 International Seminar on Application for Technology of Information and Communication, 2018, pp. 441-446.

[2] K. Langsari and R. Sarno, "Optimizing COCOMO II parameters using particle swarm method," in 2017 3rd International Conference on Science in Information Technology (ICSITech), 2017, pp. 29-34.

[3] A. Salam, A. Khan, and S. Baseer, "A comparative study for software cost estimation using COCOMO-II and Walston-Felix models," in The 1st International Conference on Innovations in Computer Science & Software Engineering,(ICONICS 2016), 2016, pp. 15-16.

[4] Z. Dan, "Improving the accuracy in software effort estimation: Using artificial neural network model based on particle swarm optimization," in Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics, 2013, pp. 180-185.

[5] A. Kumar, A. Sinhal, and B. Verma, "A Novel Technique of Optimization for Software Metric Using PSO," International Journal of Soft Computing And Software Engineering, vol. 3, pp. 2251-7545, 2013.

[6] C. Hari and P. P. Reddy, "A fine parameter tuning for COCOMO 81 software effort estimation using particle swarm optimization," Journal of Software Engineering, vol. 5, pp. 38-48, 2011.

[7] C. S. Yadav and R. Singh, "Tuning of COCOMO II model parameters for estimating software development effort using GA for PROMISE project data set," International Journal of Computer Applications, vol. 90, 2014.

[8] R. Puri and I. Kaur, "Novel Meta-Heuristic Algorithmic Approach for Software Cost Estimation," International Journal of Innovations in Engineering and Technology (IJIET), vol. 5, pp. 456-463, 2015.

[9] A. Sheta, D. Rine, and A. Ayesh, "Development of software effort and schedule estimation models using soft computing techniques," in 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), 2008, pp. 1283-1289.

[10] A. G. Saif, S. Abbas, and Z. Fayed, "Intelligent Water Dros (IWD) Algorithm for COCOMO II and COQUAMO Optimization " presented at the Proceeding of the World COngress on Egineering and Computer Science 2015.

[11] S. Kumari and S. Pushkar, "Cuckoo search based hybrid models for improving the accuracy of software effort estimation," Microsystem Technologies, vol. 24, pp. 4767-4774, 2018.

[12] A. Puspaningrum and R. Sarno, "A hybrid cuckoo optimization and harmony search algorithm for software cost estimation," Procedia Computer Science, vol. 124, pp. 461-469, 2017.

[13] D. Nandal and O. P. Sangwan, "Software cost estimation by optimizing cocomo model using hybrid batgsa algorithm," International Journal of Intelligent Engineering and Systems, vol. 11, pp. 250-263, 2018.

[14] T. T. Khuat and M. H. Le, "A Novel Technique of Optimization for the COCOMO II Model Parameters using Teaching-Learning-Based Optimization Algorithm," Journal of Telecommunications and Information Technology, 2016.

[15] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby, "Cost models for future software life cycle processes: COCOMO 2.0," Annals of software engineering, vol. 1, pp. 57-94, 1995.

[16] B. Boehm, C. Abts, B. Clark, S. Deuhani-Chulani, and I. COCOMO, "Model Definition Manual," University of Southern California, Los Angeles, California, 1998.

[17] X.-S. Yang, M. Karamanoglu, and X. He, "Multi-objective flower algorithm for optimization," Procedia Computer Science, vol. 18, pp. 861-868, 2013.

[18] K. Balasubramani and K. Marcus, "A study on flower pollination algorithm and its applications," International Journal of Application or Innovation in Engineering & Management (IJAIEM), vol. 3, pp. 230-235, 2014.

[19] X.-S. Yang, "Flower pollination algorithm for global optimization," in International conference on unconventional computing and natural computation, 2012, pp. 240-249.

[20] E. Manalif, "Fuzzy Expert-COCOMO risk assessment and effort contingency model in software project management," 2013.