# Software Cost Estimation Using Flower Pollination Algorithm

Bilal Khan[1], Rashid Naseem[2], Muhammad Binsawad[3], Muzammil Khan[1,4], Arshad Ahmad[5]
[1] Department of Computer Science, City University of Science and Information Technology, Pakistan
[2] Department of IT and Computer Science, Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Pakistan
[3] Faculty of Computer Information Systems, King Abdulaziz University Jeddah, Saudi Arabia
[4] Department of Computer and Software Technology, University of Swat, Pakistan
[5] Department of Computer Science, University of Swabi, Pakistan
bilalsoft63@gmail.com, rashid@cusit.edu.pk, mbinsawad@kau.edu.sa, muzammilkhan86@gmail.com, yaarshad@gmail.com

## Abstract

Software Development Organizations (SDO) develop a massive number of projects per year. One of the elementary and significant features of any SDO is to use a tool that can precisely estimate the software cost. It directly affects nearly all management activities including resource allocation, project planning, and project bidding. Imprecise estimation causes troubles e.g. dropping the worth of the project, waste the company's budgets and can outcome in the disaster of the project. During the last few decades' researchers have developed a large number of models for software cost estimation (SCE). However, SCE is still a challenging task. Algorithmic and non-algorithmic approaches were firstly used to achieve the goal. Each of them has their own merits and demerits but still, these are considered as primary tools for SCE. This study proposes Flower Pollination Algorithm (FPA) for SCE. Mean Magnitude of Relative Error (MMRE) is used as an evaluation metric for benchmarking the proposed model with the existing model. All the results of FPA are compared with the COCOMO model. Experimental results show a better performance of FPA as compare to COCOMO. Three datasets from NASA software projects are selected, NASA93, NASA63, and NASA60. On NASA93 dataset the improvement is 10.17%, on NASA63 the improvement is 77.38% and on NASA60 it is 22.96%.

**Keywords:** SCE, FPA, COCOMO model

## 1 Introduction

In the recent era, SDO give a lot of prominence to efficient and effective SCE techniques for their achievement [1] as it is an imperative problem in software project management (SPM). SCE aims to assess the time and cost essential for software development in early stages [2], which is the elementary concern of project management [3]. SCE models support project managers to estimate the cost, delivery time and manpower that were essential for software development [4]. On the other hand, in the life cycle of project development, the expense and time are significant [5]. In SCE the financial capitals required to implement the project to predict [6].

Abundant researches are available on SCE methods which are categorized into three main types listed below:

### 1.1 Expert Judgment

The discussions of one or more experts are required to originate the cost estimation. The estimation is obtained by uncertain and personal reasoning processes on the bases of data taken from past projects for understanding new projects and with experiences [7].

### 1.2 Parametric

Analytical and statistical equations concerning software project cost, are utilized to the numeral of project features. Constructive Cost Model (COCOMO), COCOMO II and Putnam's Software Life Cycle Model (SLIM) [8] are the very familiar parametric (algorithmic) models, used as a tool for SCE by software project managers.

### 1.3 Data Mining

The data mining (DM) models contain a minimum modeling method that takes project features to produce cost prediction. However, DM models can offer grander estimation abilities to resolve hard problems [7]. Lately, DM models have been expected as an alternative or collected with the expert judgment and parametric models [9].

The first methods were algorithmic methods that were developed in the past for SCE. They estimate the software cost using the COCOMO family [10]. The main issue has been rise massive research over the previous decades that estimating the necessary costs

*Corresponding Author: Muhammad Binsawad; E-mail: mbinsawad@kau.edu.sa
DOI: 10.3966/160792642020092105002

software projects is going to be complex as project complexities are rising day by day. That is the reason there is a requirement for non-algorithmic models that are focused on artificial intelligence and machine learning techniques that work dependent on heuristic data [10].

However, this paper introduces the Flower Pollination Algorithm (FPA) for SCE. This algorithm is applied to NASA software project datasets (NASA93, NASA63 and NASA 60). For evaluation of FPA on these datasets, MMRE is used as an evaluation metric. The obtained results are compared with the results, taken from COCOMO model applied to the same datasets. FPA, COCOMO model and evaluation metrics are described in the subsequent sections of this paper.

The respite study is planned as follows: Section 2 refers to the related work of SCE. COCOMO model and FPA are discussed in section 3 and 4. Section 5 illuminates the technique used in this exertion. Experiment and benchmarking are defined in section 6 and 7. The conclusion and future work are described in the last sections.

## 2 Related Work

As we have discussed that our focus is on data mining techniques, many researchers have performed significant research in SCE using data mining techniques. They have proposed different methods with their strengths and weaknesses. The subsequent paragraphs concise the related work in sequential order.

Maleki et al., presented a hybrid model based on GA and FA algorithms are presented for SCE [11]. For evaluation of performance three criteria were used namely MMRE, PRED and EF. MMRE of the proposed model shows that they have reduced it from 58.80% to 22.53% as compared to the COCOMO model. They have performed their experiments on the NASA93 dataset on which the hybrid model had improved efficiency rendering to the performance of criteria.

Gharehchopogh et al, introduced PSO based hybrid models for SCE [12]. Results of the PSO-FCM and PSO-LA hybrid model generates higher performance for SCE as compare with the COCOMO model. Permitting to the results of this model it can be analyzed that hybrid algorithms based on PSO can expressively increase the accuracy of SCE.

Lazarova, has used Bee Colony and hybrid algorithm for the improvement of SCE [13]. The results were compared with the COCOMO model and MMRE was selected as evaluation criteria. On the test dataset COCOMO results for MMRE is produced 0.2952% and on the proposed model it is 0.07%. Therefore, the proposed model is optimum than the COCOMO model.

Sajadfar and Ma, presents a framework for SCE with feature-based experiential data regression approach [14]. The speculative importance of the projected model is the novel feature-based hybrid method with data mining and linear regression. Nonetheless, there are certain boundaries due to that the pursuer should be attentive. The early description of industrialized structures is the main drawback so that the production procedure data can be dependably examined and managed. Such dependable element definitions need one-time semantic modeling quality in any tolerating activity. Though salvaging the definitions with irritating applications is greatly proposed and therefore the carrying out effort obstacle would be condensed expressively. Instead, data mining requires to be sustained with accessible and complete chronological data. In the present presentation case, ERP data was interpreted into a dissimilar data structure so as to care for data mining.

Gharehchopogh and Pourali, selected NASA60 dataset for training the proposed model [15]. They used genetic algorithm in their model and compare their results with the COCMO II model. The results achieved for both models illustrate that the projected algorithm is capable to improve the accuracy and reduce MMRE for SCE as compare with COCOMO II model.

Ebrahimpour et al, has gained the result that non-algorithmic methods perform well as compared with algorithmic models for SCE [10]. Persistent amounts of SCE are not quantified amounts in algorithmic models but they could be considered in the usual way. They have found that algorithmic models do not have sufficient accuracy as compare to MLP ANN and ACO. These non-algorithmic models have used a tool for SCE in their paper. The results of the proposed model show that estimation is more than 80%of the cases provide optimistic and vastly accurate magnitudes in comparison to algorithmic model on tested data. The result shows that MMRE of the proposed model is less than the COCOMO model.

Maindoab has proposed a hybrid model of two algorithms of Cuckoo optimization and KNN [6]. The proposed model was evaluated on six different datasets that are: MAXWELL, KEMERER, MIYAZAKI 1, NASA 93, NASA 63, NASA60 and assessment standards include MMRE, MMER, RMSE, MDMRE, MAPE, MSE, MAE, PRED (N). The results conclude that the proposed model performs better results on KEMERER, MIYAZAKI1, NASA60, NASA93 on all evaluation criteria as compare with COCOMO. But on NASA63 and MAXWELL using MDMRE, MSE and MAE has not good performance compared to COCOMO.

Kumari and Pushkar, proposed a novel model for SCE that is centered on the Cuckoo Searching Optimization algorithm, which has been recycled to determine the optimum bounds of SCE model [1]. The investigational result achieved on a standard dataset has exposed the preeminence of the proposed model over the presently used SCE models. The upcoming direction of the exploration can be to include numerous assessment criteria to enhance the parameters and to examine the correctness of the technique for the precise SCE. Table 1 shows the summary of related work.

**Table 1.** Summary of related work

| Year | Paper | Technique | Dataset | Compare with | Evaluation Criteria |
|---|---|---|---|---|---|
| 2014 | Maleki et al., [11] | Firefly and Genetic Algorithm | NASA 93 | COCOMO | MMRE, PRED |
| 2014 | Gharehghopogh et al., [12] | PSO, FCM, LA | NASA | COCOMO | MRE, MMRE |
| 2015 | Ebrahimpour et al., [3] | BCO, ABC, Hybrid of BCO and COA | NASA | COCOMO II | MARE, MMARE |
| 2015 | Lazarova, [13] | Linear Regression | Historical Datasets from different Companies | COCOMO | MRE, MMRE |
| 2015 | Gharehphopogh et al., [14] | COCOMO II Model using Genetic Algorithm | NASA 60 | COCOMO II | MRE, MMRE, PRED |
| 2016 | Ebrahimpour et al., [10] | MLP, ANN, ACO | NASA | COCOMO | MRE, MMRE |
| 2016 | Maindoab, [6] | Combination of COA-Cuckoo and KNN | NASA 60, NASA 63, NASA 93, MAXWEL, KEMERER, MIYAZAKI | COCOMO II, KNN, Cuckoo | MMER, MMRE, MDMRE, RMSE, MAPE, MAE, PRED, MSE |
| 2017 | Kumari & Pushkar, [1] | Cuckoo Search | NASA | COCOMO | MMRE, PRED |

## 3 COCOMO Model

This is the record widely used algorithmic model for SCE [16]. It has been usually utilized to project costs for the assortment of project and business progressions. This model is essentially grounded on assessing the project size or lines of code, and some other attributes that apply to estimates, including project attributes, personal attributes, hardware attributes, and general attributes. COCOMO model affords the SDOs with the development time in months, the effort in person-months and the team size in persons.

**The basic COCOMO model-** It calculates the software expansion exertion (and cost) as a purpose of program size which is articulated in assessed thousands of Source Lines of Code (SLOC). COCOMO put on to three sessions of software projects [17]:

*Organic mode*- It is used for minor projects that are KLOC is up to 2-50 within an accustomed environment and skilled developers.

*Semidetached mode*- It used for average projects that's KLOC is up to 50-300 on related projects with normal preceding knowledge.

*Embedded mode*- This mode is used for complex and large projects usually above than 300 KLOC with fewer experience originators.

**Intermediate Model-** Basic COCOMO does not revenue interpretation of the software development environs. So far Intermediate COCOMO Boehm presented a set of 15 attributes listed in Table 2, which match correctness to the basic COCOMO.

**Table 2.** List of attributed selected by boehm [18]

| NO | Attribute | Description | NO | Attribute | Description |
|---|---|---|---|---|---|
| 1 | RELY | Requires software reliability | 9 | AEXP | Application Experience |
| 2 | DATA | Size of the application database | 10 | PCAP | Software engineer capability |
| 3 | CPLX | Complexity of the product | 11 | VEXP | Virtual machine experience |
| 4 | TIME | Run-time performance constraints | 12 | LEXP | Programming language experience |
| 5 | STOR | Memory constraints | 13 | MODP | Application of software engineering methods |
| 6 | VIRT | Volatility of the virtual machine environment | 14 | TOOL | Use of software tools |
| 7 | TURN | Required turnabout time | 15 | SCED | Required development schedule |
| 8 | ACAP | Analyst capability | | | |

These cost drivers are assembled into four classes that are: Personal Attributes, Product Attributes, Hardware Attributes, and Project Attributes. Each Cost driver has a capacity of six ranks of rating: Very Low, Low, Nominal, High, Very High, and Extra High [16].

Each rating has an equivalent real number called Effort Multiplier (EM) Shown in Table 3 [18], grounded on the influence and the degree to which the factor can affect output. For the Intermediate COCOMO model the estimated effort in person-month is calculated as [19]:

$$\text{Effort} = a \times [\text{SIZE}]^{b} \times \prod_{i=1}^{15} EM_i \qquad \textbf{(1)}$$

here "a" is productivity coefficient and "b" is the scale factor. These coefficients depend on dissimilar project modes given in Table 4 [19].

**Table 3.** Intermediate COCOMO model cost drivers and effort multipliers

| Cost Driver | Rating | | | | | |
|---|---|---|---|---|---|---|
| | Very-Low | Low | Nominal | High | Very-High | Extra-High |
| Product Attributes | | | | | | |
| rely | 0.75 | 0.88 | 1.00 | 1.15 | 1.40 | |
| data | | 0.94 | 1.00 | 1.08 | 1.16 | |
| cplx | 0.70 | 0.85 | 1.00 | 1.15 | 1.30 | 1.65 |
| Hardware Attributes | | | | | | |
| time | | | 1.00 | 1.11 | 1.30 | 1.66 |
| stor | | | 1.00 | 1.06 | 1.21 | 1.56 |
| virt | | 0.87 | 1.00 | 1.15 | 1.30 | |
| turn | | 0.87 | 1.00 | 1.07 | 1.15 | |
| Personal Attributes | | | | | | |
| acap | 1.46 | 1.19 | 1.00 | 0.86 | 0.71 | |
| aexp | 1.29 | 1.13 | 1.00 | 0.91 | 0.82 | |
| pcap | 1.42 | 1.17 | 1.00 | 0.86 | 0.70 | |
| vexp | 1.21 | 1.10 | 1.00 | 0.90 | | |
| lexp | 1.14 | 1.07 | 1.00 | 0.95 | | |
| Project Attributes | | | | | | |
| modp | 1.24 | 1.10 | 1.00 | 0.91 | 0.82 | |
| tool | 1.24 | 1.10 | 1.00 | 0.91 | 0.83 | |
| sced | 1.23 | 1.08 | 1.00 | 1.04 | 1.10 | |

**Table 4.** Coefficient for Intermediate COCOMO

| MODE | a | b |
|---|---|---|
| Organic | 3.2 | 1.05 |
| Semi-Detached | 3 | 1.12 |
| Embedded | 2.8 | 1.20 |

This research focuses on the intermediate COCOMO model as its approximation accuracy is better than the basic version.

## 4 Flower Pollination Algorithm

Flower Pollination Algorithm (FPA) was developed by Yang, [19] encouraged by the flow pollination procedure of flowering plants [20]. It is a nature motivated algorithm that inductees the pollination performance of flowering plants [21]. FPA can be romanticizing the features of pollination procedure, flower devotion and pollination behavior as the rules given below [22]:

(1) Biotic and cross-pollination is pondered as global pollination procedure with pollen resonant pollinators acting Levy flights.

(2) Abiotic and self-pollination are pondered as local pollination.

(3) Flower dependability can be pondered as the reproduction probability is proportional to the resemblance of two flowers involved.

(4) Global pollination and local pollination is organized by a switch probability $p \in [0, 1]$. Because of the corporeal immediacy and other influences such as insects and wind, local pollination can have a momentous fraction p in the complete pollination processes the quasi code of FPA is presented in Algorithm 1.

**Algorithm 1.** Flower Pollination Algorithm

- Initialize the population (Objective function) minimum iteration and maximum iteration f(x), x = (x1, x2, x3, ……… xd)
- Initialize the population of N flowers with a random solution
- Identify the best and worst solutions in the population
- Define the switch probability $p \in [0, 1]$
  - If random <p switch
  - Global pollination (Accept and replace the previous solution)
  - Else Do local pollination (Keep the previous pollination)
  - End if
- Evaluate the best solution (output of generating minimum MMRE)
- Check new solution
  - If new solution is better
  - Then update them in the population
  - End if
  - Find the current best solution

## 5 The Proposed Model

The proposed model tries to deliver good accuracy in the SCE process by applying FPA. The core objective of applying FPA is to lessen the Mean of Magnitude Relative Error (MMRE) midst of the actual and estimated efforts. This is a mutual assessment standard used to measure the enactment of the model in the SCE practice. The MMRE is calculated as:

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Actual\ Effort - Estimated\ Effort}{Actual\ Effort}\right| \quad \textbf{(2)}$$

here, n is the total number of projects.

Figure 1 shows the flowchart of the proposed model.

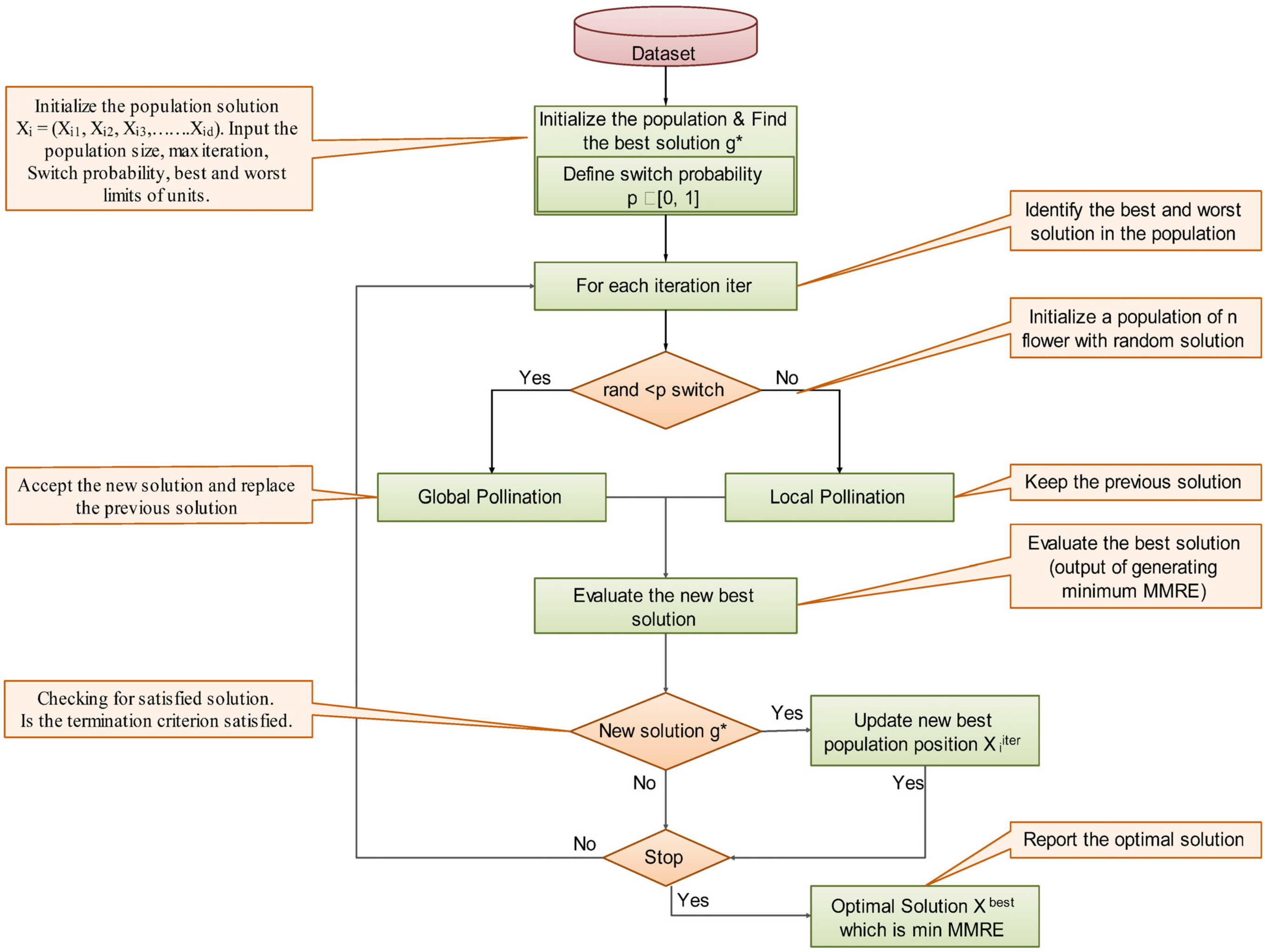



**Figure 1.** Flowchart of proposed model

The following are the steps of the proposed model.

**Step1:** The algorithm starts by initializing the primary values of the supreme significant bounds, e.g. the population size n, switch probability p ∈ [0, 1] and maximum numbers of iterations. In our case, the population size is different according to each dataset. For NASA93 it is 93, for NASA63 it is 63 and for NASA60 it is 60. The maximum number of iterations is 100.

**Step2:** The initial population xi, i = 1,…,n is produced randomly and the fitness function of every solution f(xi) in the population is assessed by computing its consistent objective function.

**Step3:** The subsequent steps are recurring until the finishing criterion fulfilled, which is to reach the preferred number of iterations.

(a) The global pollination process is started to accept and replace the previous solution. This is done by producing a random number r, where r ∈ [0, 1], for each solution xi. xi is the initial population solution.

(b) If the value of the random solution is less than the switch probability (r < p) than a new solution is generated. The New solution is generated by using a Levy distribution as follow:

$$X_i^{t+1} = X_i^{t+1} + L(X_i^t - g^*) \quad \textbf{(3)}$$

Here $X_i^t$ are the solutions from the dissimilar sinks, g* is the best solution and L is a Levy flight which is greater than 0 (L > 0) and calculated as follow:

$$L \sim \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi}\frac{1}{s^{1+\lambda}}, s \gg s_0 > 0 \quad \textbf{(4)}$$

Here Γ(λ) is the ordinary gamma function and this dispersal is effective for large steps s > 0.

(c) If global pollination is not started then the local pollination process is started. Local pollination keeps the previous best solution by producing a random number ∈, ∈ in [0, 1] as follow:

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

Where $X_i^t$, $X_j^t$ are the solutions from the dissimilar lowers. If $X_i^t$, $X_j^t$ selected from a similar population, this becomes a local random walk.

(d) All the generated solutions $X_i^{t+1}$ we'll be evaluated in the population and modernize the results in the population according to their objective values which is the optimal solution g*.

## 6 Experimental Setup

FPA and COCOMO are evaluated using the MMRE evaluation metric and tested on software projects of NASA datasets including NASA93, NASA63, and NASA60. Among all these datasets seventeen attributes that are listed in Table 2 are selected. As in meta-heuristic algorithms, the alteration of parameters is very important to reach an optimal solution and can have a momentous impact on function and efficiency. Some parameters of FPA include initial population and iteration which are 93 and 100 for NASA93, 63 and 100 for NASA63 and 60 and 100 for NASA60 dataset respectively. The experiments are also taken over the 20, 100 and 400 iterations, and the results obtained from these iterations are similar, so we selected 100 iterations for this study. The performance of meta-heuristic algorithm is better than the algorithmic model. The reason is that the algorithmic model used statistical linear regression equation and a set of nonlinear regression equations which are sensitivity for outliers and meant to describe linear relationship between variables. So if there is a nonlinear relationship than the outcomes will be a bad model. With the rising complexities of the projects, the parametric model does not estimate the constant amount for SCE. These are the reasons that we are using FPA for SCE.

## 7 Results

The evaluated results of FPA and COCOMO are listed in Table 5 and Table 6. In the tables, the first column shows the dataset utilized and the second column represents the outcomes of each algorithm. While each row represent the assessments of individual dataset. The overall outcomes show that the MMRE values of FPA are less than COCOMO. NASA93, NASA63, and NASA60 are firstly tested on the intermediate version of COCOMO. For the Intermediate COCOMO model, the estimated effort in person-month is calculated using Equation 1.

Belong to Equation 1, "a" is productivity coefficient and "b" is the scale factors which are listed below:

For organic projects, the value of "a" is 3.2 and the value of "b" is 1.05.

For semidetached projects, the value of "a" is 3 and the value of "b" is 1.12.

For embedded projects, the value of "a" is 2.8 and the value of "b" is 1.20.

The MMRE evaluation results of COCOMO are listed in Table 5.

**Table 5.** COCOMO MMRE results

| Dataset | COCOMO Results |
|---|---|
| NASA 93 | 59.4939 |
| NASA 63 | 36.3027 |
| NASA 60 | 25.3922 |

After that, all these attributes are tested on FPA which reduces the error rate of SCE. The MMRE evaluation results of FPA for all the datasets are listed in Table 6.

**Table 6.** FPA MMRE results

| Dataset | FPA Results |
|---|---|
| NASA 93 | 53.7338 |
| NASA 63 | 16.0480 |
| NASA 60 | 20.1620 |

## 8 Comparison with COCOMO Model

Table 7 shows the comparative analysis of FPA and COCOMO, where first column represents the datasets, second and third column represents the algorithms and the fourth column represents the percentage difference of FPA and COCOMO, while each row shows the assessment of individual dataset. When compare FPA with COCOMO model, these results are achieved. On NASA93 dataset the improvement is 10.17%, on NASA63 the improvement is 77.38% and on NASA60 it is 22.96% which are shown in Table 7. The percentage difference is computed using the formula given below:

$$Percentage\ Difference = \frac{|v_i - v_j|}{\frac{v_i + v_j}{2}} * 100 \quad \textbf{(6)}$$

here vi and vj are the value which difference is required.

**Table 7.** Comparative results of COCOMO and FPA with percentage difference

| Model \ Dataset | COCOMO | FPA | Percent Difference |
|---|---|---|---|
| NASA 93 | 59.4939 | 53.7338 | 10.17 % |
| NASA 63 | 36.3027 | 16.0480 | 77.38 % |
| NASA 60 | 25.3922 | 20.1620 | 22.96 % |

Figure 2 shows the comparison MMRE results of the COCOMO model and FPA and Figure 3 shows the percentage difference of each dataset.

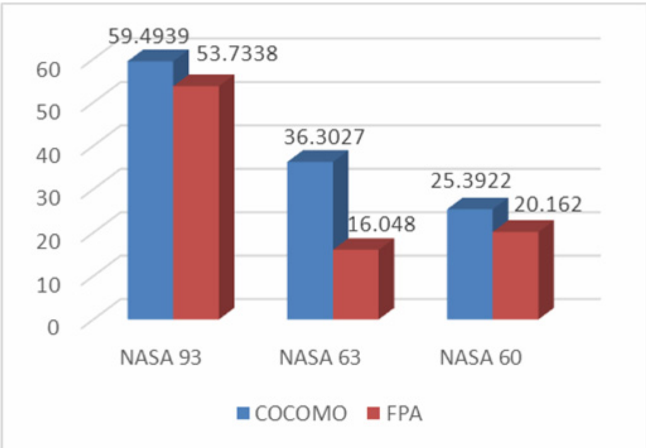



**Figure 2.** The results of MMRE evaluation using COCOMO and FPA

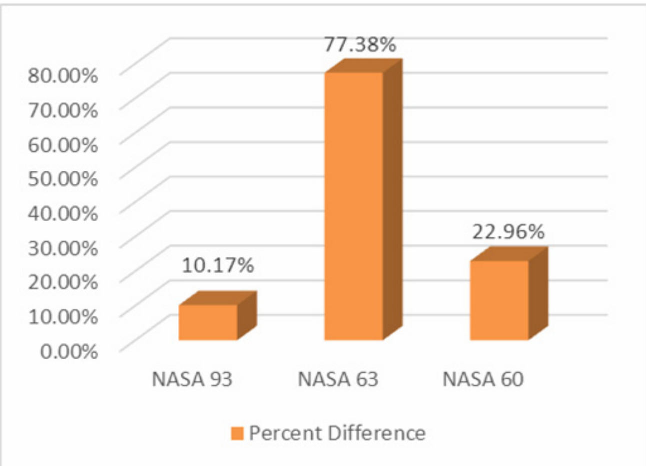



**Figure 3.** Percentage differences of result between COCOMO model and FPA using different datasets

## 9 Conclusion

It is observed from the literature that researches have developed different models for SCE, but it is still a challenging task for SDOs. SCE is one of the important and complex aspects of project management. One of the main anxieties of project and design managers is planning, budgeting, and controlling the cost. However, the common problems in most of the projects are increasing the cost of projects. Despite the wide research done in this field, the main reasons for this problem have not been completely found and an effort has not been made to solve them. In software projects, costs are directly or indirectly associated with the project milieu and these influences have a relative effect on the total function of the costs. Although direct costs are often fixed costs, they may include part of variant costs. The focus of this research is to improve the accuracy rate of SCE in terms of reducing the error rate of the evaluation metric using FPA. To achieve this goal first and foremost approaches are discussed in this research, which is Algorithmic (COCOMO), Non-Algorithmic (Experts judgment) and Data Mining (FPA) approaches. The datasets from NASA data repository were selected that are NASA93, NASA63 and NASA60. MMRE was used as an evaluation metric. The MMRE results taken from the proposed model were compared with the results of the COCOMO model.

## 10 Future Work

The ultimate goal of this research is to reduce the error rate of SCE. The proposed model has the potential for further development because of its simplicity and encouraging results. Improvement in results can be performed by using the latest algorithm or enhancement in the latest algorithms. Enhancement can be performed either by [23]:

(1) Modifying the existing algorithm.

(2) Merging the strength of different algorithms to enhance the proficiency and performance which is known is hybridization.

FPA can also be hybridized with Artificial Neural Network (ANN), Support Vector Machine (SVM) or any other classification algorithms to improve the accuracy of SCE in terms of reducing the error rate of SCE.

## References

[1] S. Kumari, S. Pushkar, Software Cost Estimation Using Cuckoo Search, in: S. Sahana, S. Saha (Eds.), *Advances in Computational Intelligence*, Vol. 509, Springer, Singapore, 2017, pp. 167-175.

[2] J. Hakkarainen, P. Laamanen, R. Rask, Neural Networks in Specification Level Software Size Estimation, *Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences*, Wailea, HI, USA, 1993, pp. 626-634,.

[3] Z. Li, X. Y. Jing, X. Zhu, Progress on Approaches to Software Defect Prediction, *IET Software*, Vol. 12, No. 3, pp. 161-175, June, 2018.

[4] A. K. Marandi, D. A. Khan, An Impact of Linear Regression Models for Improving the Software Quality with Estimated Cost, *Procedia Computer Science*, Vol. 54, pp. 335-342, January, 2015.

[5] M. Ibrahim, A New Model for Software Cost Estimation Using Bat Algorithm, *International Journal of Academic Research in Computer Engineering*. Vol. 1, No. 1, pp. 53-60, September, 2016.

[6] E. E. Miandoab, F. S. Gharehchopogh, A Novel Hybrid Algorithm for Software Cost Estimation Based on Cuckoo Optimization and K-Nearest Neighbors Algorithms, *Engineering, Technology & Applied Science Research*, Vol. 6, No. 3, pp. 1018-1022, June, 2016.

[7] J. Huang, Y.-F. Li, M. Xie, An Empirical Analysis of Data Preprocessing for Machine Learning-based Software Cost Estimation, *Information and Software Technology*, Vol. 67, No. C, pp. 108-127, November, 2015.

[8] J. Wen, S. Li, Z. Lin, Y. Hu, C. Huang, Systematic Literature Review of Machine Learning Based Software Development Effort Estimation Models, *Information and Software Technology*, Vol. 54, No. 1, pp. 41-59, January, 2012.

[9] A. B. Nassif, D. Ho, L. F. Capretz, Towards an Early Software Estimation Using Log-linear Regression and a Multilayer Perceptron Model, *The Journal of Systems and Software*, Vol. 86, No. 1, pp. 144-160, January, 2013.

[10] N. Ebrahimpour, F. S. Gharehchopogh, Z. A. Khalifehlou, New Approach with Hybrid of Artificial Neural Network and Ant Colony Optimization in Software Cost Estimation, *Journal of Advances in Computer Research*, Vol. 7, No. 4, pp. 1-12, November, 2016.

[11] I. Maleki, L. Ebrahimi, F. S. Gharehchopogh, A Hybrid Approach of Firefly and Genetic Algorithms in Software Cost Estimation, *MAGNT Research Report*, Vol. 2, No. 6, pp. 372-388, September, 2014.

[12] F. S. Gharehchopogh, L. Ebrahimi, I. Maleki, S. J. Gourabi, A Novel PSO based Approach with Hybrid of Fuzzy C-Means and Learning Automata in Software Cost Estimation, *Indian Journal of Science and Technology*, Vol. 7, No. 6, pp. 795-803, June, 2014.

[13] M. Lazarova, Parallel Artificial Bee Colony Optimisation for Solving Curricula Time-Tabling Problem, *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 4, No. 1, pp. 1-8, January, 2016. http://www.ijircce.com/volume-4-issue-1.html#

[14] N. Sajadfar, Y. Ma, A Hybrid Cost Estimation Framework Based on Feature-oriented Data Mining Approach, *Advanced Engineering Informatics*, Vol. 29, No. 3, pp. 633-647, August, 2015.

[15] F. S. Gharehchopogh, A. Pourali, A New Approach Based on Continuous Genetic Algorithm in Software Cost Estimation, *Journal of Scientific Research and Development*, Vol. 2, No. 4, pp. 87-94, April, 2015.

[16] B. Boehm, Software Engineering Economics, *IEEE Transactions on Software Engineering*, Vol. 10, No. 1, pp. 4-21, Junuary, 1984.

[17] Z. Zia, A. Rashid, K. Uz Zaman, Software Cost Estimation for Component-based Fourth-generation-language Software Applications, *IET Software*, Vol. 5, No. 1, pp. 103-110, Fereruary, 2011.

[18] M. J. Basavaraj, K. C. Shet, Empirical Validation of Software Development Effort Multipliers of Intermediate COCOMO Model, *Joutnal of Software*, Vol. 3, No. 5, pp. 65-71, May, 2008.

[19] M. Auer, A. Trendowicz, B. Graser, E. Haunschmid, S. Biffl, Optimal Project Feature Weights in Analogy-based Cost Estimation: Improvement and Limitations, *IEEE Transactions on Software Engineering*, Vol. 32, No. 2, pp. 83-92, Feberuary, 2006.

[20] X. Yang, Flower Pollination Algorithm for Global Optimization, in: J. Durand-Lose, N. Jonoska (Eds.), *International Conference on Unconventional Computing and natural Computation*, Berlin, Heidelberg, 2012, pp. 240-249.

[21] X. Yang, M. Karamanoglu, X. He, Flower Pollination Algorithm: A Novel Approach for Multiobjective Optimization, *Engineering Optimization*, Vol. 46, No. 9, pp. 1-16, October, 2014.

[22] Z. A. A. Alyasseri, A. T. Khader, M. A. Al-betar, M. A. Awadallah, X. Yang, Variants of the Flower Pollination Algorithm: A Review, in: X. S. Yang (Ed.), *Nature-Inspired Algorithms and Applied Optimization*, Springer, 2018, pp. 91-118.

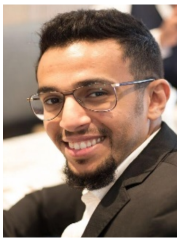
[23] R. V. Rao, V. J. Savsani, *Mechanical Design Optimization Using Advanced Optimization Techniques*, Springer, 2012, pp. 5-34.


## Biographies

**Bilal Khan** received the MCS degrees in from Northern University, Nowshera Pakistan, in 2013, and the MSCS degree from City University of Science and Information Technology, Peshawar, Pakistan in 2018. He is currently working as Lecturer in the Department of Computer Science at City University of Sciences and Information Technology, Peshawar. His research interests are Machine Learning, Machine Learning in Big Data, Data Mining, and Bio-Informatics.

**Rashid Naseem** belongs to Landikotal, Khyber, KPK, Pakistan. He received the BCS degree in computer science from the University of Peshawar, Pakistan, in 2008 and the MPhil degree in computer science from the Quaid-i-Azam University, Pakistan, in 2011. He obtained PhD in Information Technology from the Universiti Tun Hussein Onn Malaysia in February 2017. He was with the software industry from 2007 to 2008. He is currently Assistant Professor of Software Engineering at Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Mang Khanpur Road Haripur, Pakistan. Before, he was Lecturer at Department of Computer Science, City University of Science and Information Technology, Peshawar, Pakistan, since 2012, and was promoted to Assistant Professor in November 2017. He is member of

National Computing Education Accreditation Council, Pakistan. He has published a number of peer reviewed journals and conferences. He is serving as reviewer of well reputed journals and conferences. His research interests include software modularization, architecture recovery, search based software engineering, search based techniques, datamining and clustering techniques.



**Muhammad Binsawad** is a teaching assistant at the Department of Computer Information Systems, Faculty of Computing and Information Technology King Abdulaziz university. He received his master's degree (Applied Information Technology) and successfully completed his post baccalaureate certificate in Information Systems Management from Towson University in US. The Doctoral Degree (PhD) in Information Systems was awarded to him from the University of Technology Sydney (UTS) in 2019. He has professional experiences in Information Systems Design, Digital Transformation, Business Analysis, IT Project Management and Solid knowledge in the SDLC approach. He has taught several subjects in the area of information systems and software engineering at King Abdulaziz university and UTS. His research interests' areas include and not limited to Information Systems Modeling-Services, Human-Computer Interaction (HCI) and Digital Transformation. Also, he actively involved in international research and events activities, and contributing to international conferences and journals.

**Muzammil Khan** received Ph.D. Computer Science degree from Preston University Islamabad Campus, Pakistan in 2018. Currently, he is serving Department of Computer & Software Technology, University of Swat as an Assistant Professor in Computer Science. His interests include Data Science, Digital News Preservation, information retrieval from digital archives, Data Analytics, Text Processing, Named Entity Recognition and classification in Urdu manuscripts, Text/Data Mining, Machine Learning.

**Arshad Ahmad** received Ph.D. degree in Computer Science & Technology from Beijing Institute of Technology, China in 2018. He is currently working as an assistant professor of Computer Science at Department of Computer Science, University of Swabi, Pakistan. His research interests include requirements engineering, text mining, sentiment analysis and machine learning.