Flower Pollination Algorithm for Software Effort Coefficients Optimization to Improve Effort Estimation Accuracy

Alifia Puspaningrum[1], Fachrul Pralienka Bani Muhammad[2], Esti Mulyani[3]

[1,2,3] *Informatics Department, Politeknik Negeri Indramayu, Indonesia*

[1]alifia.puspaningrum@polindra.ac.id, [2]fachrul.pbm@polindra.ac.id

[3]esti.mulyani@polindra.ac.id

Abstract - Software effort estimation is one of important area in project management which used to predict effort for each person to develop an application. Besides, Constructive Cost Model (COCOMO) II is a common model used to estimate effort estimation. There are two coefficients in estimating effort of COCOMO II which highly affect the estimation accuracy. Several methods have been conducted to estimate those coefficients which can predict a closer value between actual effort and predicted value. In this paper, a new metaheuristic algorithm which is known as Flower Pollination Algorithm (FPA) is proposed in several scenario of iteration. Besides, FPA is also compared to several metaheuristic algorithm, namely Cuckoo Search Algorithm and Particle Swarm Optimization. After evaluated by using Mean Magnitude of Relative Error (MMRE), experimental results show that FPA obtains the best result in estimating effort compared to other algorithms by reached 52.48% of MMRE in 500 iterations.

Keywords: software effort estimation, flower pollination algorithm, metaheuristic algorithm

I. INTRODUCTION

Software effort estimation has an important role in project management [1]. In addition, software effort estimation is used to predict time, effort, people, and finance to develop an application [2]. Wrong estimation can bring an overestimating or underestimating effort which has a big consequence for the project [3]. The important issue in effort estimation is less accuracy of estimation caused by unclear requirements, inconsistency, and complexity of software projects.

Constructive Cost Model (COCOMO) II developed by Boehm has been a common model used to estimate effort [4]. There are two coefficients in estimating effort of COCOMO II which highly affect the estimation accuracy. There are several methods that have been conducted to optimize those coefficients. The goal of the methods is estimating effort to find an accurate effort value compared to actual effort.

Beside that, there has been so many problems are solved using optimization algorithms to find the best solution [5], one of them is in estimating effort by using COCOMO model. There are several alghorithms that have been conducted to optimize COCOMO II model parameters, such as genetic algorithm [6], bat algorithm [7], differential evolution [8], firefly algorithm [9], deep neural network [10], particle swarm optimization, dolphin algorithm [1], ant colony optimization [11], cuckoo search algorithm [12], etc.

Current trend is to use nature-inspired metaheuristic algorithms to tackle such difficult problems, and it has been shown that metaheuristics are surprisingly very efficient. In 2012, Xin She Yang developed flower pollination algorithm which inspired from pollination process of flowers, which is the way plants maintain their generations [13]. Beside that, FPA (Flower Pollination Algorithm) has only one key parameter p (switch probability) which makes the algorithm easier to implement and faster to reach optimum solution. Moreover, this transferring switch between local and global pollination can guarantee escaping from local minimum solution. In addition, Flower Pollination Algorithm has witnessed explored in several domains [5], such as an optimum capacitor placement in radial distribution systems [14], solar PV parameter estimation [15-16], optimizing layouts of nodes in wireless sensor network [17].

According to the adventages of FPA, this paper proposes flower pollination algorithm to estimate the optimal coefficients of software effort by using NASA 93 dataset. The paper is written as follows: Section 2 contains the proposed method. Section 3 result and discussion of the experimental results. The last, Section 4 consists of conclusion and future work.

II. METHOD

A. COCOMO II

In this reaserch, research uses NASA 93 as one of COCOMO II model dataset. COCOMO II has 17 effort multipliers, 5 scale factors, and software size. The atribut of effort multipliers and scale factor of COCOMO is written in Table I.

TABLE I
EFFORT MULTIPLIERS AND SCALE FACTOR OF COCOMO II

Effort Multipliers				Scale Factor
Product Attributes	**Computer Attributes**	**Personnel Attributes**	**Project Atributes**	
Required Software Reliability Size of Database Complexity of Product Reusability Documentation describe what life cycle needs	Constraint of Time Execution Constraint of Main Storage Volatility of Platform	Ability of Analyst Ability of Programmer Continuity of Personnel Experience of Application Experience of Platform Language and Tool Experience	Software Tool Multisite Development Schedule of Required Development	Precedentedness Development Flexibility Risk Resolution Team Cohesion Process Maturity

Table I is describe effort multipliers and scale factor used in COCOMO II. There are four categories of effort multipliers, namely: product attributes, computer attributes, personnel attributes, and project attributes. However, there is no categories of scale factor, only several attributes within. By using these attributes, effort value can be estimated. Effort is also called as Person-Months (PM) which is calculated how long software is developed by one person in a month, as shown in (1).

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

In (1), as default, A = 2,94 is initiated by COCOMO II. In addition, and EM is effort multipliers for each attribute in several categories explained in Table 1. Project line of code is written in E which is calculated by using (2):

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

with B = 0,91 is initiated by COCOMO II and SF is scale factor for each attribute explained in Table 1.

B. Flower Pollination Algorithm

Flower pollination algorithm is a new metaheuristic algorithm developed by [13]. This algorithm is inspired by the pollination process of flowers, which is a way for plants to maintain their generations. There are two mechanisms in this pollen transfer, namely cross pollination and self-pollination. The results of pollination are tested with an objective function and the results are better, the value will be maintained. Flowchart of flower pollination algorithm is illustrated is Fig. 1.

Yang [13] developed FPA in four steps as listed as follows:

- The global pollination processes are divided into two kind of process, namely biotic and cross pollination which the pollen transports pollinators perform the levy flight.
- Local pollination is represented as abiotic and self pollination as a process which does not need any pollinators.
- Reproduction probability is proportional to the similarity of two flowers involved
- A switch probability p ∈ [0, 1] controls the interaction between local pollination and global pollination, lightly biased toward local pollination.

Besides flowchart system of flower pollination algorithm, in estimating software effort, there are several steps which implemented in the proposed method.

- Step 1: Create population which each N population represents a possible combination of two coefficients, namely A and B
- Step 2: Calculate the effort value by using those two coefficient
- Step 3: Calculate the deviation between the actual effort and the predicted effort value
- Step 4: The first – third step will be iterated as many solution initialized.

The deviation value is used to be the fitness function for each solution. The smallest value of the fitness is declared as the best solution. Table II describes flower pollination algorithm parameters which is used in the experiment.

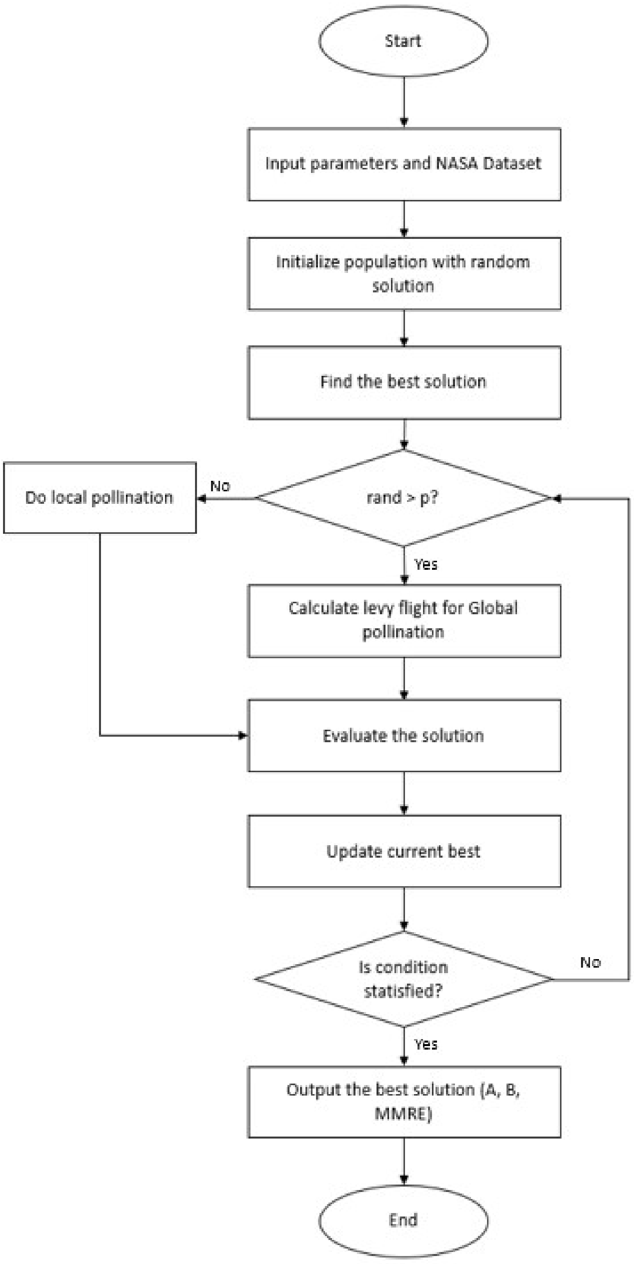

Fig. 1 Flower pollination algorithm flowchart

TABLE II
FLOWER POLLINATION ALGORITHM PARAMETER

Parameter	Value
Probability Switch	0.25
Iteration	500, 1000, 1500, 2000, 2500
Dimension	2
Lower Bound	-1
Upper bound	5

C. Evaluation Criteria and Dataset

Magnitude of Relative Error (MRE) and Mean Magnitude of Relative Error (MMRE) are calculated to evaluate the estimation between the actual and the predicted value. MMRE is also defined as fitness function for flower pollination algorithm. The formula of MRE is calculated by using (3). Then, Mean Magnitude of Relative Error (MMRE) is calculated by using the value of MRE as shown in (4).

The best performance is stated from the minimum value of MMRE. In addition, NASA 93 is used for the dataset. Beside that, Matlab is used to process the proposed method.

III. RESULTS AND DISCUSSION

In this paper, flower polllination algorithm is applied for optimizing effort coefficients of COCOMO II, namely A and B. By used these two coefficients the effort value is calculated. To obtain a good result, this paper analyze several kind scenario. The first scenario is finding the best iteration by assesed some iteration, namely, 500, 1000, 1500, 2000, and 2500 which is illustrated in Fig.2. The second condition is comparing the proposed method to some metaheuristic algorithm, namely, cuckoo search algorithm and particle swarm optimization.

$$MRE = |actual\ effort_i - predicted\ effort_i|/actual\ effort_i \quad (3)$$

$$\frac{1}{n}\sum_{i=1}^{n}|actual\ effort_i - predicted\ effort_i|/actual\ effort_i \quad (4)$$

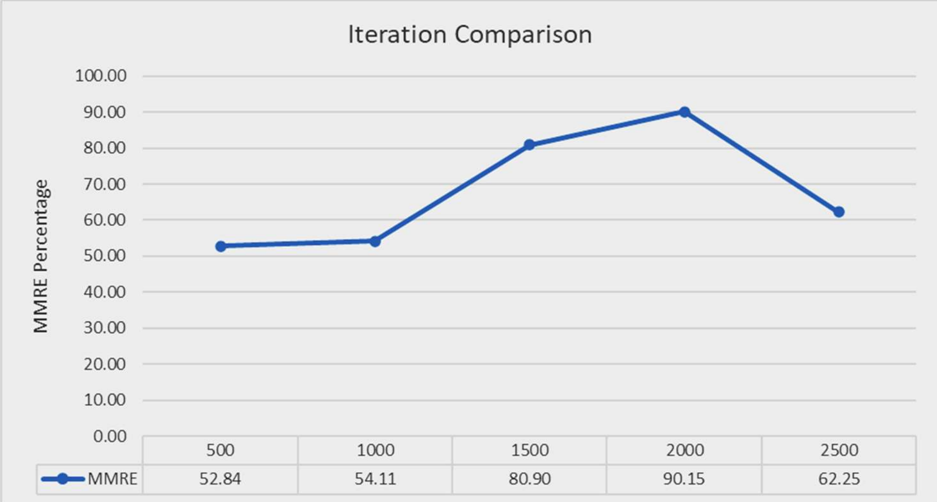

Fig. 2 Iteration comparison of effort estimation

In this experiment, two coefficients is searched for the optimal value. MMRE becomes objective value or fitness value of the optimization algorithm. To calculate effort estimation, A and B coefficients are being balancing parameters. In addition, effort multiplier, line of code, and scale factor for each projects are also parameters which influence value of effort. Fig 2 shows the MMRE value of effort estimation performance according to five kind of scenario. The experiment shows that the best value is reached in 500 iteration. So, it means that predicted value of the proposed method is closer to the actual effort.

Table III considers five kind of iteration scenario to confirm the ability of flower pollination algorithm in finding an optimum solution. For each kind of iteration scenario, the experiment is done five times. The best value is in 500 iteration by reached 52.48% of MMRE. In finding the optimal value of effort coefficients, FPA is performed iteratively based on objective function which has been specified before. The system converges when the criteria's fixed for the stoppage is reached [5]. However, there is no guarantee that the best solution is going to be reachable always. So that, as shown in Table III, eventhough the best MMRE is reached in 500 iteration, the other experiments in 500 iteration even show the worst value of MMRE.

By using the value of 500 iteration, the proposed method is compared to fixed value of COCOMO and several algorithm such as cuckoo search algorithm and particle swarm optimization. This experiment is done to confirm the ability of flower pollination algorithm over other metaheuristic algorithm to find optimum value of effort estimation. Fig 3 shows that the proposed method is outperform by reaching a lower value of MMRE than other algorithms.

TABLE III
COMPARISON OF MMRE ACCORDING TO FLOWER POLLINATION ALGORITHM ITERATION

Iteration	Coefficients		MMRE
	A	B	
500	3.31	1.00	55.39
	4.22	1.25	184.46
	2.45	1.77	1921.86
	4.62	**1.00**	**52.84**
	1.82	1.76	1327.22
1000	4.58	0.00	94.29
	2.08	1.37	150.93
	4.13	0.44	87.64
	1.00	1.00	80.03
	5.00	**1.00**	**54.11**
1500	4.45	-0.46	97.19
	2.54	0.62	86.41
	4.05	0.02	94.57
	0.76	0.68	94.15
	3.20	**0.67**	**80.90**
2000	4.22	0.00	94.57
	1.05	0.17	97.92
	4.25	0.01	94.48
	5.00	**0.28**	**90.15**
	2.16	0.26	94.73
2500	-0.12	1.20	106.37
	4.94	-0.08	94.59
	5.00	**0.78**	**62.25**
	0.00	1.67	100.00
	4.36	0.00	94.46

As result of MMRE performance in 93 project is showed in Fig 3, Table IV shows the optimized coefficients for each algorithms. By using flower pollination algorithm, value of A and B can be optimized well. NASA 93 consists of 93 projects, to analyze the performance, Table V shows the performance of effort estimation for some project.

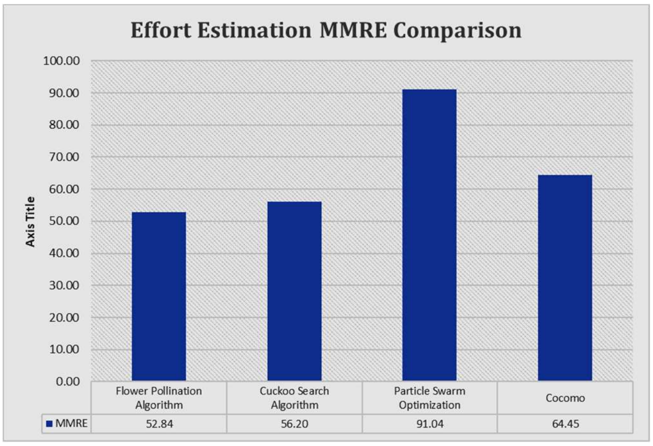

Fig. 3 An example of a graphic figure

TABLE IV
MMRE COMPARISON BETWEEN PROPOSED METHOD AND OTHER ALGORITHMS

Algorithm	Coefficients		MMRE
	a	b	
Flower Pollination Algorithm	**4.62**	**1.00**	**52.84**
Cuckoo Search Algorithm	3.00	1.02	56.20
Particle Swarm Optimization	4.39	0.28	91.04
COCOMO II	2.94	0.91	64.45

TABLE V
EFFORT ESTIMATION PERFORMANCE

Project No.	Actual Effort	Estimated Effort				MRE			
		COCOMO	FPA	CSA	PSO	COCOMO	FPA	CSA	PSO
9	72	27.56	51.91	35.23	11.62	61.73	27.90	51.07	83.86
13	36	15.57	30.43	20.84	5.08	56.75	15.47	42.11	85.90
14	215	167.41	398.18	285.99	13.91	22.13	85.20	33.02	93.53
15	48	19.99	41.13	28.52	4.56	58.36	14.32	40.59	90.51
24	90	33.21	66.60	45.89	9.07	63.10	26.00	49.01	89.92
36	42	20.54	38.21	25.85	9.48	51.09	9.03	38.46	77.44
39	42	21.58	40.89	27.79	8.74	48.62	2.65	33.84	79.19
40	114	45.01	90.77	62.63	11.80	60.52	20.37	45.06	89.65
55	370	119.68	267.44	189.20	15.36	67.65	27.72	48.87	95.85
64	8.4	42.41	92.89	65.39	6.26	71.73	38.08	56.41	95.83
86	458	343.53	786.00	559.24	37.40	80.62	55.66	68.45	97.89
92	12	52.71	97.60	65.96	25.04	339.22	713.37	449.69	108.69

Furthermore, not 93 projects are getting the optimal value, there are also some projects which is worse than other methods, such as project 14 and 92.

IV. CONCLUSION

In this paper, flower pollination algorithm is used to optimize COCOMO II in order to estimate effort precisely. The proposed method has been tested in NASA 93 dataset. From the result of the test, it is shown that flower pollination algorithm can achieve the best result compared to other algorithms by reached 52.48% of MMRE in 500 iterations. But the parameters used in this algorithm is still static. So that, tuning parameter of the algorithm should be conducted for the next research in order to obtain the best result.

ACKNOWLEDGMENT

Authors completely would like to give thank to Politeknik Negeri Indramayu for encouragement in providing facilities and financial support in publishing this paper.

REFERENCES

[1] A. A. Fadhil, R. G. H. Alsarraj, and A. M. Altaie, “Software Cost Estimation Based on Dolphin Algorithm,” *IEEE Access*, vol. 8, pp. 75279–75287, 2020.

[2] R. De Lemos *et al.*, “Software Engineering for Self-Adpaptive Systems A second Research Roadmap,” *Softw. Eng. Self-Adaptive Syst.*, no. 10431, pp. 1–32, 2011.

[3] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, “Software development effort estimation using regression fuzzy models,” *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–17, 2019.

[4] B. W. Boehm, "Software Engineering Economics," *IEEE Trans. Softw. Eng.*, vol. SE-10, no. 1, pp. 4–21, 1984.

[5] H. Chiroma, N. L. M. Shuib, S. A. Muaz, A. I. Abubakar, L. B. Ila, and J. Z. Maitama, "A review of the applications of bio-inspired Flower Pollination Algorithm," *Procedia Comput. Sci.*, vol. 62, no. Scse, pp. 435–441, 2015.

[6] A. F. Sheta, "Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects," *J. Comput. Sci.*, vol. 2, no. 2, pp. 118–123, Feb. 2006.

[7] M. Ibrahim, "A New Model for Software Cost Estimation Using Bat Algorithm," *Int. J. Acad. Res. Comput. Eng.*, vol. 1, no. 1, pp. 53–60, 2016.

[8] B. W. Boehm, "Software cost estimation meets software diversity," *Proc. - 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Companion, ICSE-C 2017*, pp. 495–496, 2017.

[9] N. Ghatasheh, H. Faris, I. Aljarah, and R. M. H. Al-Sayyed, "Optimizing software effort estimation models using firefly algorithm," *J. Softw. Eng. Appl.*, vol. 8, no. March, pp. 133–142, 2015.

[10] M. S. Khan, F. Jabeen, S. Ghouzali, Z. Rehman, S. Naz, and W. Abdul, "Metaheuristic Algorithms in Optimizing Deep Neural Network Model for Software Effort Estimation," *IEEE Access*, vol. 9, pp. 60309–60327, 2021.

[11] V. Venkataiah, R. Mohanty, J. S. Pahariya, and M. Nagaratna, "Application of Ant Colony Optimization Techniques to Predict Software Cost Estimation," *Comput. Commun. Netw. Internet Secur.*, pp. 315–325, 2017.

[12] A. Puspaningrum and R. Sarno, "A Hybrid Cuckoo Optimization and Harmony Search Algorithm for Software Cost Estimation," *Procedia Comput. Sci.*, vol. 124, pp. 461–469, 2017.

[13] X. S. Yang, "Flower pollination algorithm for global optimization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7445 LNCS, pp. 240–249, 2012.

[14] V. Tamilselvan, T. Jayabarathi, T. Raghunathan, and X. S. Yang, "Optimal capacitor placement in radial distribution systems using flower pollination algorithm," *Alexandria Eng. J.*, vol. 57, no. 4, pp. 2775–2786, 2018.

[15] D. F. Alam, D. A. Yousri, and M. B. Eteiba, "Flower Pollination Algorithm based solar PV parameter estimation," *Energy Convers. Manag.*, vol. 101, pp. 410–422, 2015.

[16] J. P. Ram, T. S. Babu, T. Dragicevic, and N. Rajasekar, "A new hybrid bee pollinator flower pollination algorithm for solar PV parameter estimation," *Energy Convers. Manag.*, vol. 135, pp. 463–476, 2017.

[17] T. T. Nguyen, J. S. Pan, and T. K. Dao, "An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network," *IEEE Access*, vol. 7, pp. 75985–75998, 2019.