JURNAL INFOTEL
VOL. 17, NO. 1, FEBRUARY 2025, PP. 122–135. DOI:10.20895/INFOTEL.V17I1.1250

RESEARCH ARTICLE

Software Effort Coefficient Optimization Using Hybrid Bat Algorithm and Whale Optimization Algorithm

Alifia Puspaningrum[1,*], Muhamad Mustamiin[2], Fauziah Herdiyanti[3], and Kamaludin Noviyanto[4]

[1,2,3,4]Politeknik Negeri Indramayu, Jawa Barat 45252, Indonesia

*Corresponding email: alifia.puspaningrum@polindra.ac.id

Received: October 10, 2024; Revised: November 21, 2024; Accepted: February 13, 2025.

Abstract: Software effort estimation is crucial in software engineering, especially in project management. It defines a person's effort to develop an application in a certain amount of time. One of the models which is widely used for this purpose is the Constructive Cost Model (COCOMO) II. In COCOMO II, two coefficients play a significant role in determining the accuracy of the effort estimation. Various methods have been conducted to estimate these coefficients to closely match the actual effort with the predicted values, such as particle swarm optimization, cuckoo search algorithm, *etc*. However, several metaheuristics have a limit in exploration and exploitation to find an optimal value. To overcome this problem, a hybrid metaheuristic combining the Bat Algorithm and the Whale Optimization Algorithm (BAWOA) is proposed. This approach aims to optimize the two COCOMO II coefficients for better estimation accuracy. The proposed method is also compared to other metaheuristic algorithms using the NASA 93 datasets. This research used two evaluation criteria: Magnitude of Relative Error (MRE) and Mean Magnitude of Relative Error (MMRE). With the optimal score among the comparison methods, the proposed method achieves superior effort estimation, with an MMRE of 51.657%. It also shows that hybrid BAWOA can estimate predicted effort close to the actual effort value.

Keywords: BAWOA, COCOMO II, hybrid algorithm, metaheuristic algorithm, software effort estimation

---

 ISSN: 2085-3688; e-ISSN: 2460-0997

1 Introduction

Software effort estimation is critical in project management, especially during the initial phases of software development, which include planning and analysis. Accurate cost estimation is essential to produce high-quality software in the allocated time and budget [1]. The challenge of achieving precise and stable cost estimation in software engineering is still ongoing, as incorrect estimations can lead to significant consequences due to overestimation or underestimation of effort [2–5]. Various approaches are available for software project cost estimation; one widely used model is the Constructive Cost Model (COCOMO). Developed by Boehm [6] in 1981, COCOMO has become a standard in the field.

To enhance the accuracy of COCOMO II parameters, optimization algorithms have been employed. The current trend favors nature-inspired metaheuristic algorithms for solving complex problems efficiently, including effort estimation using the COCOMO model. Algorithms such as the fuzzy model [7], Grey Wolf Optimizer [8], particle swarm optimization [9,10], biogeography-based optimization [11], strawberry algorithm [12], dolphin algorithm [13], flower pollination algorithm [14], genetic algorithm [15], and bat algorithm [16] have been applied to optimize COCOMO II parameters.

In 2010, inspired by bats lifecycle, Yang [17] developed the Bat Algorithm (BA). The algorithm has been successfully applied in various sectors, including health [18, 19], agriculture [20, 21], transportation [22], wireless sensor network [23]. While BA has shown better results than other techniques, its local search in the exploration phase indicates room for improvement [24]. To address this, the Whale Optimization algorithm (WOA) [25] can be combined with BA to enhance both exploration and exploitation capabilities. WOA has also been successfully implemented in sectors like health [26, 27], agriculture [28, 29], transportation [30, 31], *etc*.

The hybrid Bat and Whale Optimization Algorithm (BAWOA) leverages the strengths of both BA and WOA, leading to better convergence towards the global optimum. In addition, the NASA 93 dataset is being used to evaluate the proposed model's performance.

The paper is organized as follows: the proposed method, including a preliminary study and method, is described in Section 2, Section 3 declares the experimental result. In contrast, the experimental results are discussed in Section 4. Lastly, the conclusion and future work are described in Section 5.

2 Research Method

2.1 COCOMO II

The Constructive Cost Model (COCOMO) is a commonly used parametric model for estimating the effort required for collocated projects. It provides calculations for various parameters, such as person-months, to estimate the effort needed. This research utilizes the NASA 93 dataset, which is part of the COCOMO II model. COCOMO II features 22 cost drivers, divided into 17 effort multipliers (EM) and five scale factors (SF), and software size. Table 1 lists the attributes of these effort multipliers and scale factors.

Effort multipliers consist of four categories, namely: product attributes, computer attributes, personnel attributes, and project attributes. Unlike effort multipliers, scale factors do not have categories but consist of various attributes. These attributes are used to estimate the effort value, also known as Person-Months (PM), which indicates the time re-

Table 1: Effort multipliers and scale factor of COCOMO II

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

quired for one person to develop the software in a month. This relationship is shown in (1).

$$PM = A \times \text{Size}^E \times \prod_{i=1}^{17} \text{EM}_i \tag{1}$$

In (1), the default value for A is set to 2.94, as established by COCOMO II. Furthermore, EM represents the value for each attribute within the various categories described in Table 1. The project kilo line code is defined as size, while the variable E is calculated using (2).

$$E = B + 0.01 \times \Sigma_{j=1}^{5} \text{SF}_j \tag{2}$$

In (2), the default value for B is set to 0.91, as specified by COCOMO II. SF represents the scale factor for each attribute described in Table 1.

2.2 Bat Algorithm

Yang [17] constructed a metaheuristic optimization algorithm called the Bat Algorithm (BA), inspired by the echolocation phenomena observed in nature, such as dolphins, ants, swarms and whales. The BA follows several rules:

- Each bat uses echolocation to determine its distance from prey.
- Bats fly with velocity v_i towards a position x_i and emit signals within a given frequency interval $(f_{\text{min}}, f_{\text{max}})$ by varying wavelengths (λ) and loudness (A) to detect prey.

- Bats can adjust the wavelength and pulse rate of their signal to calculate the distance to their target,
- It is assumed that A decreases from an initial maximum value (A_0) to a constant minimum value (A_{min}).

The results of iteration are evaluated using an objective function and, when the results are better, the value will be maintained. The BA pseudocode is given in **Algorithm 1**.

Algorithm 1: Pseudocode of Bat Algorithm

```
Initialize the bat population x_i (i = 1, 2, ..., n) and velocities v_i
Define pulse frequency f_i at x_i
Initialize pulse rates r_i and loudness A_i

while (t < MaxIter) do
  for each bat i in the population do
    Generate new solutions by adjusting frequency, velocity, and location

    if rand() > r_i then
      Select a solution among the best solutions
      Generate a local solution around the selected best solution
    end if

    Generate a new solution by flying randomly

    if rand() < A_i and f(x_i) < f(x*) then
      Accept the new solutions
      Increase r_i and reduce A_i
    end if

  Rank the bats and find the current best x*
  t = t + 1
end while

Output the best solution found
```

2.3 Whale Optimization Algorithm (WOA)

In 2016, Mirjalili and Lewis [25] developed a metaheuristic optimization method named the Whale Optimization Algorithm (WOA), which is inspired by the life cycle of whales, including the bubble-net hunting behavior of humpback whales, where whales encircle and spiral toward their prey, forming a unique hunting pattern. WOA emulates this natural strategy through three primary components:

- Encircling Prey: This operator places whale agents around the optimal solution.
- Bubble-Net Attacking (exploitation phase): This phase leverages the best solutions by either shrinking the encirclement or using a spiral movement towards the prey.
- Search for Prey (exploration phase): This phase improves exploration by randomizing direction and position updates to locate global optima and avoid local minima.

The pseudocode of the WOA algorithm is given in **Algorithm 2**.

Algorithm 2: Pseudocode of Whale Optimization Algorithm (WOA)

Initialize the whale population X_i ($i = 1, 2, ..., n$)
Initialize the parameters a, A, C, l, and p
Evaluate the fitness function
$X^* =$ the best search agent
 	
while ($t <$ MaxIter) **do**
  for each whale i in the population **do**
    Update the parameters a, A, C, l, and p
 	
    if ($p < 0.5$) **then**
      if ($\|A\| < 1$) **then**
        Update the position using the best whale X^*
      else if ($\|A\| \geq 1$) **then**
        Select a random whale (X_{rand})
        Update the position using X_{rand}
      end if
 	
    else if ($p \geq 0.5$) **then**
      Update the position using spiral-shaped path
    end if
 	
  end for
Check if any solution goes beyond the search space and amend it
Recalculate the fitness of each search agent
Update X^* if there is a better solution
$t = t + 1$
end while
 	
Return X^* as the best solution

2.4 Hybrid Bat and Whale Optimization Algorithm (BAWOA)

This proposed method aims to expand and improve the optimization performance by combining the BA search capability with the WOA exploration strengths. There are three main steps, namely: `initialization`, `main iteration`, and `termination`.

`Initialization` starts with initial populations for both algorithms and sets their respective parameters. In `main iteration` step, the main loop iterates over the process where both algorithms operate in phases, and their interactions enhance overall performance. In the main loop, each single algorithm runs their phase, bats move and perform local searches based on their echolocation characteristics. At the same time, whales update their positions using encircling prey and bubble-net attacking mechanisms. For hybrid interaction, periodic exchange of information between bat and whale populations is used to leverage their combined strengths.

In `termination`, the loop continues until a specified condition is met, such as a maximum number of iterations or convergence criteria. BAWOA algorithm pseudocode is given in **Algorithm 3**.

Algorithm 3: Pseudocode of BAWOA

```
Initialize the whale population X_i (i = 1, 2, ..., n)
Initialize the parameters a, A, C, l, and p
Evaluate the fitness function
X* = the best search agent

while (t < MaxIter) do
  for each whale i in the population do
    Update the parameters a, A, C, l, and p

    if (p < 0.5) then
      if (|A| < 1) then
        Update the position using the best whale X*
      else if (|A| ≥ 1) then
        Select a random whale (X_rand)
        Update the position using X_rand
      end if

    else if (p ≥ 0.5) then
      Update the position using spiral-shaped path
    end if

  end for
Check if any solution goes beyond the search space and amend it
Recalculate the fitness of each search agent
Update X* if there is a better solution
t = t + 1
end while

Return X* as the best solution
```

2.5 Evaluation Criteria and Dataset

Two evaluation criteria were used for this research. The first criterion is Magnitude of Relative Error (MRE), which is used to calculate Mean Magnitude of Relative Error (MMRE) as the second criterion. These criteria compared predicted to actual value. MMRE is used as a fitness function for each algorithm. Eq. (3) defines the formula of MRE.

$$\textbf{MRE} = \frac{|\textbf{actual effort}_i - \textbf{predicted effort}_i|}{\textbf{actual effort}_i} \tag{3}$$

In addition, MRE is used to calculate MMRE as shown in (4).

$$\textbf{MMRE} = \frac{1}{n}\sum\nolimits_{i=1}^{n}\frac{|\textbf{actual effort}_i - \textbf{predicted effort}_i|}{\textbf{actual effort}_i} \tag{4}$$

The minimum value of MMRE is defined as the optimal value. Furthermore, dataset used in this experiment is NASA 93. In addition, the experiment process is constructed in Google Colaboratory using Python.

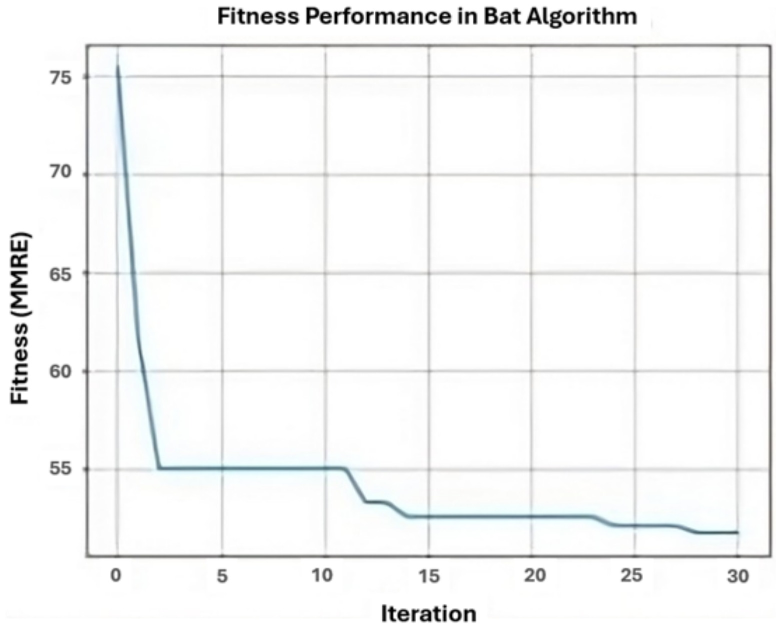

Figure 1: Iteration performance of BA.

3 Results

This paper proposes a hybrid metaheuristic BAWOA to optimize two effort COCOMO II coefficients. There are two scenarios to evaluate the performance of the model. The first scenario is to iterate through some scenarios of model parameters to find the best performance. To ensure the performance of the model, comparing the proposed method with other metaheuristic algorithms becomes the second scenario. The second scenario compares single and hybrid metaheuristic algorithms with the proposed method.

In the first scenario, there are two model parameters, namely iteration and population, to find the best performance between a single algorithm of the BA, the WOA, and Hybrid BAWOA. Each model is tested with 60 independent runs, the number of populations is in the range of 5 to 50, and the number of iterations is in the range of 10 to 31.

Figure 1 shows the iteration performance of BA. Figure 2 shows the iteration performance of WOA. Compared to BA, WOA reaches faster convergence. However, it is outperformed by BA after the 5th iteration.

Figure 3 shows the performance of the hybrid BAWOA. WOA has the best convergence rate to find the global minimum compared to other models. The proposed method explores and exploits increased capabilities.

To verify the efficiency of the proposed method, 60 scenarios are evaluated using three model algorithms to calculate the best, worst, mean, median, and standard deviation values as shown in Table 2. BAWOA yields a smaller value for the best and mean values than the worst results obtained by the other algorithms. BAWOA also obtained the best std value.

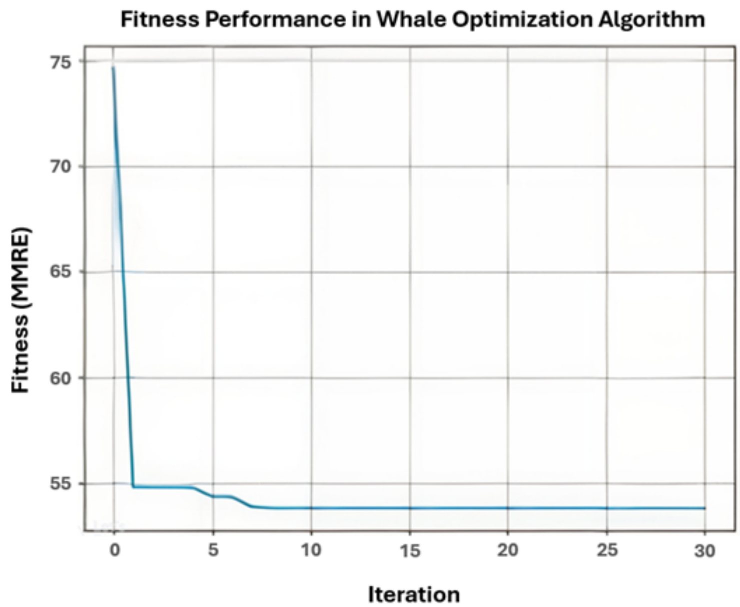

Figure 2: Iteration performance of WOA.

Table 2: Result comparison of BA, WOA, and BAWOA

Algorithm	Best	Median	Worst	Mean	Std
Bat Algorithm (BA)	51,704	52,090	73,395	53,992	4,453
Whale Optimization Algorithm (WOA)	53,827	54,625	69,487	56,721	4,084
Hybrid BA and WOA (BAWOA)	51,657	52,359	75,270	53,903	3,699

In finding the standard value, the results of the three algorithms are not ideal, indicating that the standard value is quite high. It shows that the variance of MMRE is quite high. However, BAWOA is quite stable for solving COCOMO II problems compared to a single algorithm. Table 3 shows each algorithm's five best MMRE values of effort estimation.

The five best MMRE performances for each algorithm and optimal coefficients are shown in Table 3. Using BAWOA, the value of the effort coefficient can be optimized well. The best coefficient for each algorithm is compared with several algorithms in the second scenario. It consists of a single algorithm, COCOMO-II coefficients, and compares the proposed method to another hybrid algorithm, namely the hybrid cuckoo search and harmony search Algorithm.

Table 4 shows the MMRE of the estimation performance. For single algorithm comparison, BA yields the best performance in 51.704%. In addition, BAWOA yields the best performance compared to CSHS for hybrid algorithm comparison. In addition, BAWOA has a lower MMRE compared to BA. It shows that the proposed method can find optimal parameters which can generate predicted value which is closer to the actual effort value. After having the best parameters, SoEEst for the software interface is constructed. The

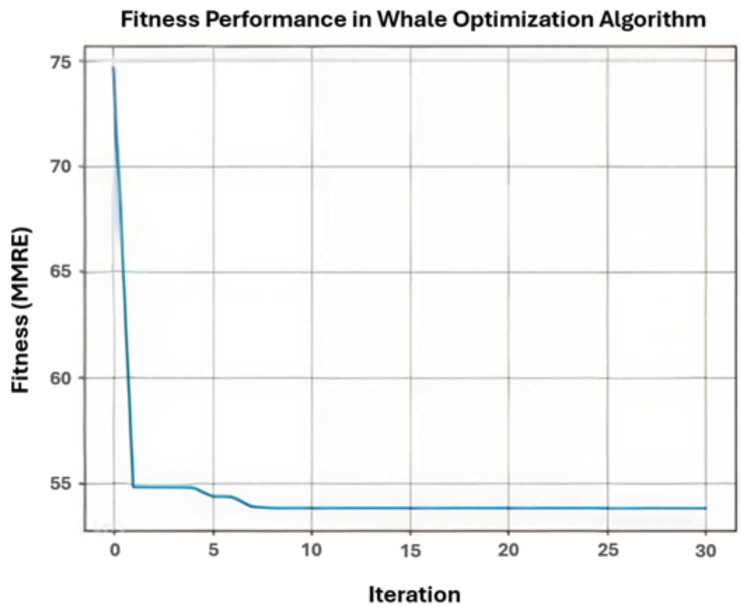

Figure 3: Iteration performance of hybrid BAWOA.

Table 3: Sample performance evaluation of first scenario

Algorithm	Population	Iteration	Coefficients		MMRE(%)
			A	B	
Bat Algorithm (BA)	27	31	5.234	0.941	51.704
	26	31	4.999	0.955	51.711
	38	31	5.617	0.933	51.712
	46	31	5.000	0.957	51.715
	41	31	5.608	0.921	51.725
Whale Optimization Algorithm (WOA)	27	31	5.000	0.799	53.827
	26	31	5.000	0.799	53.827
	38	31	5.000	0.799	53.827
	46	31	5.000	0.799	53.827
	41	31	5.000	0.799	53.827
Hybrid BA and WOA (BAWOA)	27	31	5.307	0.941	51.657
	26	31	5.474	0.930	51.677
	38	31	5.214	0.944	51.682
	46	31	5.601	0.930	51.682
	41	31	5.515	0.927	51.689

model is implemented in website-based software named SoEEst as shown in Figure 4 and Figure 5. The result of SoEEst is a person-month or effort estimate to build the software.

After having the best parameters, the SoEEst is constructed for the software interface. The result of SoEEst is a person-month or effort estimate to build the software.

Table 4: MMRE comparison of effort estimation

Algorithm	Coefficients		MMRE (%)
	A	B	
COCOMO II	2.94	0.91	64.45
Bat Algorithm (BA)	5.234	0.941	51.704
Whale Optimization Algorithm (WOA)	5.000	0.799	53.827
Flower Pollination Algorithm (FPA)	4.62	1.00	52.84
Cuckoo Search Algorithm (CSA)	3.00	1.02	56.20
Particle Swarm Optimization (PSO)	4.39	0.28	91.04
Hybrid Cuckoo Search and Harmony Search (CSHS)	5.16	0.88	54.11
Hybrid Bat and Whale Optimization (BAWOA)	5.307	0.941	51.657

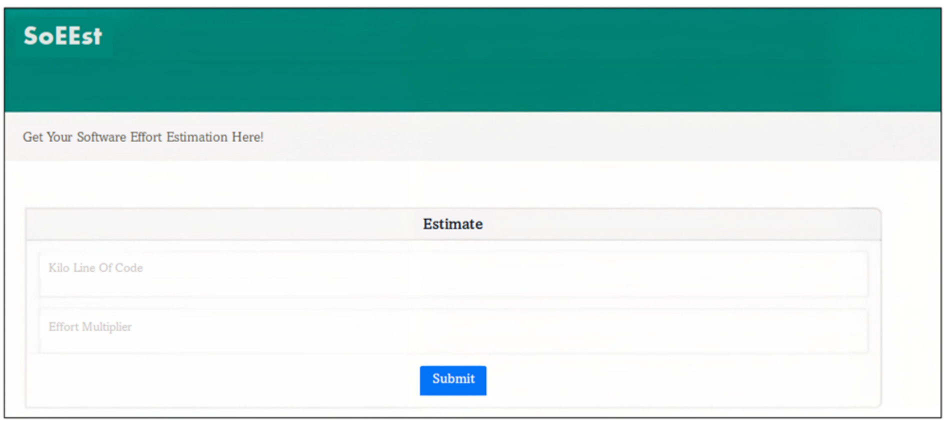

Figure 4: Iteration performance of hybrid BAWOA.

4 Discussion

This research uses a hybrid model of bat and whale optimization algorithms to predict the optimal value of the COCOMO II effort parameter. The aim is to achieve the lowest MMRE value, which illustrates the least deviation of the predicted value from the actual value. So, this research uses MMRE as a fitness value. To find optimal parameters, each model is run iteratively to achieve optimal fitness values. However, in several conditions, the best value MMRE might be the worst value because of limitations in exploration and exploitation.

In the first scenario, there is a population and an iteration used to confirm the model's ability to find an optimal solution. In Figure 2, WOA produces a faster convergence rate compared to BA, but BA finally converges to the value of WOA, which approaches BA-WOA. Finally, BAWOA reaches the optimal solution significantly superior to other algorithms. Apparently, BAWOA shows good exploitation at finding optimal value, outperforms all other approaches. Good exploitation can help algorithms search for a promising wide range of solutions but may initially slow down convergence because it takes time to explore. In addition, exploitation focuses on exploiting information from the exploitation result to improve the current solution.

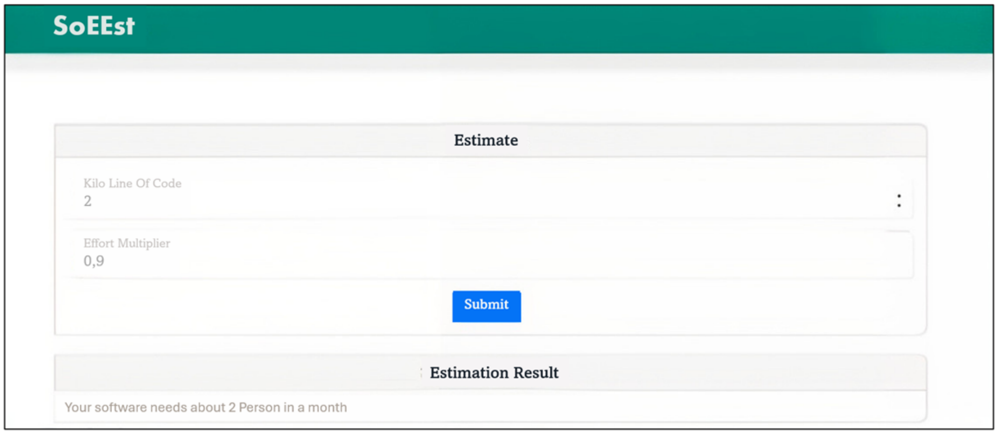

Figure 5: Iteration performance of hybrid BAWOA.

In the second scenario, the fixed value of COCOMO and several single algorithms are evaluated and compared with the proposed method. This scenario aims to ensure that the ability of the proposed method is better than that of other metaheuristic algorithms to find the optimal value of the effort estimation.

From the above analyses about Table 5, we can conclude that the proposed method BAWOA outperforms the other six algorithms. In general, BA, WOA, FPA, and CSA are inferior to PSO and the fixed value of COCOMO. In addition, BA, WOA, and FPA perform better than CSA and CSHS, respectively. However, BAWOA demonstrated good performance compared to the single algorithm and CSHS. It also shows that the proposed method is more efficient in optimizing the COCOMO parameter than other metaheuristic optimization algorithms.

NASA 93 consists of 93 projects for which each algorithm can correctly predict. Experimental results show that some projects have worse MMRE than COCOMO II, such as projects 17 and 18. Furthermore, BA is inferior in several projects, such as project 37, and WOA is inferior in projects 6 and 7. Several parameters besides the A and B coefficients are used to calculate effort estimation, namely kilo lines of code (KLOC), effort multiplier, and scale factors. In dataset NASA 93, the proposed method can optimize the value of A and B well according to the MMRE comparison in Table 5. So, hybrid BA and WOA promise better exploration and exploitation to estimate the effort.

5 Conclusion

In this paper, a hybrid metaheuristic BAWOA is proposed to optimize two coefficients precisely in the estimation of the effort. The NASA 93 dataset is used to evaluate the performance of the proposed method. The experimental result shows that the proposed method can deliver the best performance, outperforming other algorithms in the MMRE value by reaching 51.657%. However, estimating effort parameters is static. So, for future research, the cost driver coefficient tuning can be implemented to find an optimal value.

Table 5: Effort estimation MRE comparison

Project No.	Actual Effort	Effort				MRE (%)			
		COCOMO	**BA**	**WOA**	**BAWOA**	**COCOMO**	**BA**	**WOA**	**BAWOA**
6	8.4	5.63	9.29	7.93	9.42	33.02	10.59	5.55	12.13
7	10.8	9.10	14.38	11.50	14.58	15.71	33.14	6.46	35.00
15	48	30.59	39.04	24.38	39.59	36.27	18.66	49.22	17.52
17	324	323.59	330.15	154.83	334.76	0.13	1.90	52.21	3.32
18	60	54.75	66.44	38.89	67.37	8.75	10.73	35.19	12.28
27	70	48.50	66.50	43.08	67.43	30.72	5.00	38.45	3.68
37	60	48.33	64.70	42.94	65.61	19.45	7.84	28.43	9.34
55	370	221.80	240.54	131.84	243.89	40.06	34.99	64.37	34.08
58	8.4	2.23	4.01	3.88	4.06	73.48	52.31	53.76	51.65
61	458	233.48	277.03	144.76	280.89	49.02	39.51	68.39	38.67
64	150	71.64	84.65	47.89	85.83	52.24	43.57	68.07	42.78
79	409	196.59	236.72	126.44	240.02	51.93	42.12	69.09	41.31
81	1350	733.35	888.34	518.78	900.73	45.68	34.20	61.57	33.28
86	1772.5	582.50	696.06	367.58	705.77	67.14	60.73	79.26	60.18
91	480	188.85	246.57	158.47	250.01	60.66	48.63	66.99	47.91

Acknowledgments

The authors would like to thank Politeknik Negeri Indramayu for providing encouragement and financial support in publishing this paper.

References

[1] S. P. Singh, "Cost estimation model using enhance-based differential evolution algorithm," *Iran J. Comput. Sci.*, vol. 3, pp. 115–126, June 2020.

[2] B. Khan, W. Khan, M. Arshad, and N. Jan, "Software cost estimation: Algorithmic and non-algorithmic approaches," *ijdsaa*, vol. 2, pp. 1–5, Mar. 2020.

[3] M. N. Alatawi, "Forecasting the software engineering model's effort estimation using constructive cost estimation models," *Iran J. Comput. Sci.*, vol. 7, pp. 735–754, 2024.

[4] S. M. R. Chirra and H. Reza, "A survey on software cost estimation techniques," *J. Softw. Eng. Appl.*, vol. 12, no. 06, pp. 226–248, 2019.

[5] M. A. Saleem, R. Ahmad, T. Alyas, M. Idrees, Asfandayar, A. Farooq, A. Shahid, and K. Ali, "Systematic literature review of identifying issues in software cost estimation techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 8, 2019.

[6] B. W. Boehm, "Software engineering economics," in *Pioneers and Their Contributions to Software Engineering*, pp. 99–150, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.

[7] D. Manikavelan and R. Ponnusamy, "Software quality analysis based on cost and error using fuzzy combined COCOMO model," *J. Ambient Intell. Humaniz. Comput.*, Mar. 2020.

[8] R. R. Putri, D. O. Siahaan, and C. Fatichah, "Improve the accuracy of software project effort and cost estimates in cocomo ii using gwo," *2021 5th International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 128–133, 2021.

[9] S. Chhabra and H. Singh, "Optimizing design of fuzzy model for software cost estimation using particle swarm optimization algorithm," *International Journal of Computational Intelligence and Applications*, vol. 19, no. 1, p. 2050005, 2020.

[10] A. B. Nassif, M. Azzeh, A. Idri, and A. Abran, "Software development effort estimation using regression fuzzy models," *Comput. Intell. Neurosci.*, vol. 2019, p. 8367214, Feb. 2019.

[11] A. Ullah, B. Wang, J. Sheng, J. Long, M. Asim, and Z. Sun, "Optimization of software cost estimation model based on biogeography-based optimization algorithm," *Intell. Decis. Technol.*, vol. 14, pp. 441–448, Jan. 2021.

[12] M. S. Khan, C. A. ul Hassan, M. A. Shah, and A. Shamim, "Software cost and effort estimation using a new optimization algorithm inspired by strawberry plant," in *2018 24th International Conference on Automation and Computing (ICAC)*, IEEE, Sept. 2018.

[13] 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.

[14] A. Puspaningrum, F. P. B. Muhammad, and E. Mulyani, "Flower pollination algorithm for software effort coefficients optimization to improve effort estimation accuracy," *JUITA J. Inform.*, vol. 9, p. 139, Nov. 2021.

[15] A. F. Sheta, "Estimation of the COCOMO model parameters using genetic algorithms for NASA software projects," *J. Comput. Sci.*, vol. 2, pp. 118–123, Feb. 2006.

[16] "A new model for software cost estimation using bat algorithm,"

[17] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Studies in Computational Intelligence, pp. 65–74, Berlin: Springer, Berlin, Heidelberg, 2010.

[18] D. Gupta, J. Arora, U. Agrawal, A. Khanna, and V. H. C. de Albuquerque, "Optimized binary bat algorithm for classification of white blood cells," *Measurement (Lond.)*, vol. 143, pp. 180–190, Sept. 2019.

[19] A. S. Abdullah, J. Rahebi, Y. E. Özok, and M. Aljanabi, "A new and effective method for human retina optic disc segmentation with fuzzy clustering method based on active contour model," *Med. Biol. Eng. Comput.*, vol. 58, pp. 25–37, Jan. 2020.

[20] Y. Han, J. Wu, B. Zhai, Y. Pan, G. Huang, L. Wu, and W. Zeng, "Coupling a bat algorithm with XGBoost to estimate reference evapotranspiration in the arid and semiarid regions of china," *Adv. Meteorol.*, vol. 2019, pp. 1–16, Oct. 2019.

[21] A. Zarei, S.-F. Mousavi, M. Eshaghi Gordji, and H. Karami, "Optimal reservoir operation using bat and particle swarm algorithm and game theory based on optimal water allocation among consumers," *Water Resour. Manage.*, vol. 33, pp. 3071–3093, July 2019.

[22] M. He, L. Sun, X. Zeng, W. Liu, and S. Tao, "Node layout plans for urban underground logistics systems based on heuristic bat algorithm," *Comput. Commun.*, vol. 154, pp. 465–480, Mar. 2020.

[23] Trong-The Nguyen, J.-S. Pan, and T.-K. Dao, "A compact bat algorithm for unequal clustering in wireless sensor networks," *Appl. Sci. (Basel)*, vol. 9, p. 1973, May 2019.

[24] D. Nandal, Guru Jambheshwar University of Science and Technology, O. Sangwan, and Guru Jambheshwar University of Science and Technology, "Software cost estimation by optimizing COCOMO model using hybrid BATGSA algorithm," *Int. J. Intell. Eng. Syst.*, vol. 11, pp. 250–263, Aug. 2018.

[25] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, May 2016.

[26] R. K. Saidala and N. Devarakonda, "Improved whale optimization algorithm case study: Clinical data of anaemic pregnant woman," in *Advances in Intelligent Systems and Computing*, Advances in intelligent systems and computing, pp. 271–281, Singapore: Springer Singapore, 2018.

[27] I. Sudha and R. Nedunchelian, "A secure data protection technique for healthcare data in the cloud using homomorphic encryption and Jaya–Whale optimization algorithm," *Adv. Complex Syst.*, vol. 10, p. 1950040, Dec. 2019.

[28] W. Guo, T. Liu, F. Dai, and P. Xu, "An improved whale optimization algorithm for forecasting water resources demand," *Appl. Soft Comput.*, vol. 86, p. 105925, Jan. 2020.

[29] Y. Tikhamarine, A. Malik, K. Pandey, S. S. Sammen, D. Souag-Gamane, S. Heddam, and O. Kisi, "Monthly evapotranspiration estimation using optimal climatic parameters: efficacy of hybrid support vector regression integrated with whale optimization algorithm," *Environ. Monit. Assess.*, vol. 192, p. 696, Oct. 2020.

[30] H. Zhang, L. Tang, C. Yang, and S. Lan, "Locating electric vehicle charging stations with service capacity using the improved whale optimization algorithm," *Adv. Eng. Inform.*, vol. 41, p. 100901, Aug. 2019.

[31] Y. Wang, C. Yang, and H. Hou, "Risk management in perishable food distribution operations," *Ind. Manag. Data Syst.*, vol. ahead-of-print, Oct. 2019.