ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

A Hybrid Artificial Bee Colony and Artificial Fish Swarm Algorithms for Software Cost Estimation

Hawar Othman Sharif[1], Mazen Ismaeel Ghareb[2], Hoshmen Murad Mohamedyusf[3]

[1]Department of Computer Science, College of Science, University of Sulaimani, Iraq, [2]Department of Computer Science, College of Science and Technology, University of Human Development, Kurdistan Region, Iraq, [3]Department of Plastic Arts, Halabja Fine Arts Institute, Halabja, Iraq

ABSTRACT

Software cost estimation (SCE), estimating the cost and time required for software development, plays a highly significant role in managing software projects. A somewhat accurate SCE is necessary for a software project to be successful. It allows effective control of construction time and cost. In the past few decades, various models have been presented to evaluate software projects, including mathematical models and machine learning algorithms. In this paper, a new model based on the hybrid of the artificial fish swarm algorithm (AFSA) and the artificial bee colony (ABC) algorithm is presented for SCE. The initial population of AFSA, which includes the values of the effort factors, is generated using the ABC algorithm. ABC algorithm is used to solve the problems of the AFSA algorithm such as population diversity and getting stuck in a local optimum. ABC algorithm achieves the best solutions using observer and scout bees. The evaluation of the combined method has been implemented on eight different data sets and evaluated based on eight different criteria such as mean magnitude of relative error and PRED (0.25). The proposed method is more error-free than current SCE methods, according to the results. The error value of the proposed method is lower on NASA60, NASA63, and NASA93 datasets.

Index Terms: Artificial Fish Swarm Algorithm, Software Cost Estimation, Artificial Bee Colony Algorithm, Constructive Cost Model

1. INTRODUCTION

Software is the most expensive part of computer systems. Thus, Software cost estimation (SCE) is vital in the profit and loss of the company, and the smallest error in SCE can cause a lot of financial and time loss to the software development company [1]. One of the most challenging challenges for software engineers is SCE. Project failure might result from poor estimating. The estimating error is the primary cause of this issue. Estimating cost and time at the beginning of the software production and development cycle is the biggest challenge for software projects. Knowing the cost of a software project is of particular importance for companies and software production companies. The cost of creating and producing the project is not known relatively precisely at the beginning of the software production, the company may have problems and the project may fail [2]. Among the problems that exist in SCE, we can mention the wrong use of the SCE process, not using the correct methods, or problems that may not allow accurate SCE. It is not accurately calculated because many variables influence the calculation of SCE, such as people, environment, and technology politics. The software engineering community frequently discusses estimation methods for software effort estimation (SEE) and SCE development. Correct forecasts may significantly improve a company's success in terms of SCE and resource allocation. Conversely, inaccurate estimates

Access this article online

DOI:10.21928/uhdjst.v8n1y2024.pp129-141 E-ISSN: 2521-4217 P-ISSN: 2521-4209

Corresponding author's e-mail: Hawar Othman Sharif, Department of Computer Science, College of Science, University of Sulaimani, Iraq. hawar.sharif@univsul.edu.iq

Received: 12-01-2024 Accepted: 05-04-2024 Published: 07-05-2024

can lead to significant financial losses and project failure [3]. Therefore, it is crucial to find novel models with improved estimate skills.

Over the past few decades, methods such as machine learning and meta-heuristic evolution have been employed to predict the time and cost of software projects. These models have shown their effectiveness compared to other approaches. Therefore, an appropriate model that can precisely forecast the SEE and ECE in software development is required. The production of software and its development is one of the inevitable problems of today's world, and these problems are not possible without accurate SCE. The success or failure of software depends on determining the required resources (including human, financial, and time resources) [4]. Algorithmic and non-algorithmic techniques are generally used in software project SCE, with algorithmic models being more significant due to their simplicity [5]. Among the algorithmic models, constructive cost model (COCOMO) SCE can be mentioned. This model is presented by Boehm in three basic, intermediate, and advanced types. It serves as a model for calculating the time, money, and effort required to complete software projects [6].

Despite the strengths of this model, the estimated cost is not sufficiently accurate the possibility of error is very high, and the dimensions of the problem expand with the increase in the number of factors and response variables. Software engineering has continuously sought ways to improve and increase the predictability of software development efforts and costs. Boehm and his colleagues pioneered this field by introducing COCOMO models [7]. These models serve as the foundation for SEE and SCE, which are necessary for every software project. The SCE of a software project strongly influences whether it is successful or not. The cost of the project should be estimated based on the basic information from the software.

Usually, the task of SCE in companies and programming teams is the responsibility of the project manager, who of course must be an experienced project manager and have previous knowledge in these fields [8]. Estimating resources, cost, and schedule for a software engineering activity depends on things like experience, and access to relevant statistical data. Inherently, software projects have risk, and this risk leads to ambiguity, which itself depends on things such as the ability to convert estimates based on size, human labor, time, software capabilities, and the stability of product requirements and the environment of software engineering activities. SCE is a key tool used by developers and project managers to estimate the time and financial investment required for software projects. Although linear and gradient algorithms can nearly always discover the best solution, they are expensive and inefficient for difficult optimization tasks. Approximate algorithms are used to quickly solve complicated problems because they have a high possibility of finding solutions that are near to optimum. Heuristic and meta-heuristic algorithms are the two broad categories into which approximate algorithms are separated. The best solution is found using meta-heuristic algorithms, a class of stochastic algorithms.

In this paper, the hybrid model of artificial fish swarm algorithm (AFSA) [9] and artificial bee colony (ABC) algorithm [10] is used to manage software projects and the error of these algorithms is also investigated. Two measures, the effort adjustment factor and the number of lines of code (KLOC), play a fundamental role in SCE. The former indicates the size of the project, while the latter takes into account project-specific conditions and requirements such as hardware limitations, personnel capabilities, and project complexity. To acquire a reasonable estimate of the time and cost required for software development, both of these criteria are essential. This paper's primary contributions are as follows:

- Increase productivity by providing a hybrid method for accurate SCE projects
- Improvement of AFSA algorithm using ABC algorithm. AFSA algorithm is not able to find the best solutions due to the population diversity problem. Therefore, the ABC algorithm is used to generate the initial population of AFSA
- Evaluation of the proposed method on eight main data sets of cost estimation and comparison of the proposed method with other models.

The general structure of this paper is as follows: In Section 2, the previous studies in the field of SCE are reviewed. ABC algorithm is explained in Section 3. In Section 4, the proposed method and its steps are explained. In Section 5, the evaluation and results of the proposed method and its comparison with other models are done. And finally, in the ion 6, conclusions and future works are stated.

2. RELATED WORKS

Correct cost estimation makes the project manager a strong support for making different decisions during the software lifecycle. The software development team's project manager, analyst, designer, programmer, and other members should

be aware of how much time and effort will be needed to create a quality result. Software development employing AI techniques has been the subject of extensive research in recent years.

The development of large industrial software systems with the highest reliability and availability requirements results in a great cost. That's why different companies started to develop such costly systems by reusing already developed components. Sethy and Rani [11] stated that SCE is always done before project implementation. To measure the cost of a software project, accurately several estimation models are present in the literature. In this research, the author presents a hybrid model of SCE based on COCOMO and function points. Both the model's function point and COCOMO are considered accurate for SCE. This model uses a hybrid formula to calculate effort and project size. The IVR dataset was used to develop the proposed method in MATLAB. From the result, it was concluded that the hybrid model of COCOMO and function point.

SCE was discovered when the COCOMO II dataset was used to train a multilayered feed forward Artificial Neural Network (ANN) using the back-propagation approach [12]. MSE and mean magnitude of relative error (MMRE) were used to validate the findings of this study. However, providing a precise and accurate estimation for a software project is still considered the most challenging task. The major reason for this failure in a software project is inaccurate software development norms; the rapid change in the technology becomes more puzzling for the software development industry. ANN is to adjust different dependent and independent variables among complex sets of bonds. The data set of COCOMO is utilized to train and test the network. Performance measurement measures include mean square error (MSE) and MMRE. From the result, it was concluded that the presented model delivers superior and accurate predictions for software development efforts.

An ANN with two independent activation functions that are based on the Taguchi approach was proposed by Rankovic *et al*. [13]. Based on a procedural approach, six different datasets were employed in this study. The clustering technique was used to apply the input values. The validation of the results was based on the MMRE. This work reduces the execution time by requiring fewer repeats. Polynomial Analogy estimation was suggested by Shahpar *et al*. [14] to increase the SCE's accuracy of prediction. To determine the similar characteristics of the supplied project, an analogy approach is applied.

The Whale-crow optimization (WCO) method [15] combines the Crow Search Algorithm (CSA) with the Whale Optimization Algorithm (WOA). By identifying the ideal regression coefficients for regression models like the linear regression model and the kernel logistic regression model, the WCO technique aims to develop an optimal regression model for SCE. The experiment uses four datasets from the Promise software engineering repository to conduct the practical performance analysis. The provided Kernel Regression model achieves the average MMRE at a rate of 0.2692, whereas the recommended linear regression model does so at a rate of 0.2442, proving the usefulness of the suggested strategy of SCE. To optimize four COCOMO-II coefficients and get optimal estimation, the author of Puspaningrum and Sarno [16] presented a hybrid model combining harmony search algorithm and the cuckoo search algorithm (CSA). Utilizing the MRE and MMRE, the suggested methodology is tested on the NASA 93 dataset. The suggested technique outperforms COCOMO-II and the CSA, according to experimental data, in evaluating the work and time required to construct a software project.

A fuzzy inference system has been developed in [17] to determine the relevant effort multiplier for each cost driver. It offers guidance on how to improve fuzzy logic-based COCOMO utilizing the particle swarm optimization algorithm employing evolutionary-based optimization strategies. Utilizing assessment measures such as mean and amount of the relative error, which were calculated using COCOMO NASA2 and COCOMONASA datasets, it is verified. The model outperforms other optimization techniques like the genetic algorithm.

The authors of Singh *et al*. [18] suggest a brand-new multi-objective differential evolution (MODE) algorithm. It is validated in two phases during the validation process. First, the MODE algorithm is included in the novel homeostatic factor-based mutation operator. The Pareto optimality concept is applied. They use a MODE to increase candidate solution variety and convergence rates, resulting in improved solutions that support evolution. The efficiency of the suggested strategy is assessed using eight bi- and tri-objective test function benchmarks. In comparison to the most recent iterations of MOEAs, the proposed technique fared well. Second, using it for SCE, the suggested approach is put to use for an application-based test. Multiobjective parameters, such as two and three objectives-based SCE, are also included in this technique. In terms of lowering work and minimizing mistakes, the suggested strategy produces superior outcomes in the majority of software projects.

Turkish Industry software projects and NASA-93 are the two typical data sets used in the experiments. The Manhattan distance and the MMRE are two performance measures used to assess the performance of the proposed algorithm biogeography-based optimization (BBO)-COCOMO-II. The suggested BBO technique, according to simulation findings, enhances the COCOMO-II coefficients now in use for a more accurate prediction of software project cost or effort [19].

In Gouda and Mehta [20], the authors discuss the value of the meta-heuristic algorithms in tackling numerous optimization problems that arise in software applications and mathematical models. The novel evolutionism-based self-adaptive mutation operator is used in the proposed approach to address multi-objective optimization issues. The problems with MODE algorithms are addressed by this method. The Pareto optimality principle and the evolutionism-based self-adaptive mutation operator are integrated in a MODE technique to improve the diversity of potential solutions. They have used the non-dominated sorting method to lessen Pareto dominance's temporal complexity. Eight benchmark test functions were used to gauge the effectiveness of the proposed method, and it outperformed the most recent MOEAs. Further, research is conducted on the suggested method, which accurately predicts SCEs by improving the tuning parameters of the multi-objective COCOMO. For all objective tasks, the proposed approach outperforms the other traditional benchmark algorithms in mean absolute error (MAE), root mean square error, and terms of prediction.

3. ABC ALGORITHM

One of the population-based algorithms that was introduced in 2005 [10] is the ABC algorithm. The three groups into which the bees are divided using the ABC algorithm Bees that work, watch, and scout. About half of the colony is made up of worker bees, and the other half is made up of observation bees. Worker bees convey food information to observer bees as they look for food near the food source stored in their memory. Worker bees frequently discover an appropriate food supply, and observer bees frequently do the same. Scout bees are worker bees that leave their existing food sources to seek for new ones. Like other population-based algorithms, the ABC algorithm is an iterative process. In the ABC algorithm, the process of searching for food is started by the worker bees. Each worker bee performs a special dance upon finding food, and the observer bees inside the hive look at the dance of the worker bees to use it to know the location of the food source. Scout bees randomly look for food in the surrounding environment. The ABC algorithm's initialization procedures for worker bees, spectator bees, and scout bees are carried out as follows:

3.1. Initial Population

Eq. (1) defines the starting population of solutions.

$$x_{ij} = x_{min,j} + \text{rand}[0,1]\left(x_{max,j} - x_{minj}\right)$$
$$i \in 1,2,\ldots,SN; j \in 1,2,\ldots,D \quad (1)$$

3.2. Worker Bees

At this step, artificial bees explore the area surrounding the food source at point x_i in search of new, better food sources v_i. Eq. (2) is used to pinpoint the new position of the food supply.

$$v_{ij} = x_{ij} + \varphi_{ij}\left(x_{ij} - x_{kj}\right) j = 1,2,\ldots,D; k = 1,2,\ldots,SN \quad (2)$$

In Eq. (2), $x_i = [x_{i1}, x_{i2},\ldots,x_{in}]$ is the position vector of the i^{th} bee, D is equal to the dimensions of the solutions. $v_i = [v_{i1}, v_{i2},\ldots,v_{in}]$ is the new position vector of the bees, an integer random number in the interval [1, SN] is that SN is equal l to the number of artificial bees. A uniformly distributed random number in the range [−1, 1] makes up the parameter ϕ_{ij}. Using Eq. (3), the random number x_i is chosen at random from the problem's domain.

$$x_{ij} = L_i + rand(0,1) \times (U_i - L_i) \quad (3)$$

In Eq. (3), U_i and L_i are the upper and lower limits of variable x_{ij}, respectively, and the procedure rand() uses random values between 0 and 1. After determining the location of the new food source, I have to calculate its optimality. For this purpose, the fitness level of vector x_{ij} is defined according to Eq. (4).

$$fit_i = \begin{cases} \dfrac{1}{1+f_i} & f_i \geq 0 \\ 1+abs(f_i) & f_i < 0 \end{cases} \quad (4)$$

3.3. Onlooker Bees

At this stage, each of the onlooker bees decides to search around the found food sources with a certain probability. The onlooker bees make their choice based on the possible values of the worker bees. Therefore, Eq. (5) is used to calculate the likelihood that watcher bees will select a food source.

$$p_i = \frac{fit_i}{\sum_{j=t}^{SN} fit_i} \quad (5)$$

The surrounding food source's position is determined using the updated location of the food source after all of the spectator bees have chosen their preferred food sources using Eq. (5). This process continues until the condition required to terminate the program is met.

3.4. Scout Bees

In the ABC algorithm, if an optimal solution cannot be identified after a predefined number of iterations, some bees leave the several and return to their original roles as scout bees to conduct a random search. They deal with the scope of the problem. The likelihood of discovering the overall ideal solution might be considerably increased by implementing the scout bee's stage.

4. THE PROPOSED METHOD

4.1. COCOMO Model

Among algorithmic models, the COCOMO model is the most well-known and often applied. This model is a basic method used to predict the number of people needed each month for software development in the industry. This model is also able to provide an estimate of the development time per month, the amount of effort required in each phase of software development, and the amount of cost required. Boehm has offered three levels of this model, which include basic, intermediate, and advanced COCOMO. Most software projects use the baseline model to estimate software development costs.

The basic COCOMO model is good for estimating software cost estimates, says Boehm about this model. Its accuracy is necessarily limited because there are no factors known for the accuracy of various hardware limitations, the quality and experience of people, the use of modern technology, tools, and other project characteristics, to have a significant impact on the cost. In the COCOMO model, the average cost estimate of software projects is calculated according to Eq. (6) [21].

$$PM = a*(size)^b * \prod_{i=1}^{15} EM_i \quad (6)$$

The values of the fixed parameters a and b in Eq. (6) are determined by the dataset's data. The size argument indicates the project's size in thousands of lines of code (KSLOC). The EM parameter is a coefficient that increases or decreases the effort rate per person/month. Parameters a, b, and c are initialized in the intermediate COCOMO in accordance with (Table 1).

TABLE 1: Values for the middle COCOMO model's parameters

Class of projects	A	B	C
Organic	2. 4	1. 05	2. 5
Semidetached	3. 0	1. 12	2. 5
Embedded	3. 6	1. 20	2. 5

COCOMO: Constructive cost model

The relatively small tasks in the Organic class are completed by highly skilled teams. Typically, if a project is 100 KSLOC or larger, it is assigned to the organic class. Semidetached projects range in size from 100 to 300 KSLOC and are moderately sized; they are neither simple nor complex. Projects that are embedded in classes typically exceed 300 KSLOC. When hardware and operations are already established and do not require any adjustments, this class is utilized.

4.2. Hybrid ABC with AFSA

In this paper, the hybrid of ABC and AFSA is used for SCE. First, the initial formulation is generated by the ABC algorithm, and then, the optimized population is given as input to the AFSA algorithm. The purpose of the proposed method is to improve random answers in the AFSA algorithm. The proposed method's flowchart is shown in (Fig. 1). The proposed method uses the ABC algorithm to diversify the population and prevent it from hitting the local optimum. By creating the initial population and injecting them into the AFSA algorithm, the ABC algorithm strengthens the solutions and discovers global solutions. ABC algorithm increases the global position update probability and convergence speed.

(Fig. 2) shows the Pseudocode of the proposed method.

4.2.1. ABC algorithm

First, a population between 0.9 and 1.4 is formed as the initial population. The set of random values is stored in the matrix x, and then in the worker bee phase, according to Eq. (7), a solution is created in the neighborhood of the existing solution in the memory. The bees update the random values at each step and find the best values of the effort factors.

$$v_{ij} = x_{ij} + \varphi_{ij}\left(x_{ij} - x_{kj}\right) j = 1,2,\ldots,D; k = 1,2,\ldots,SN \quad (7)$$

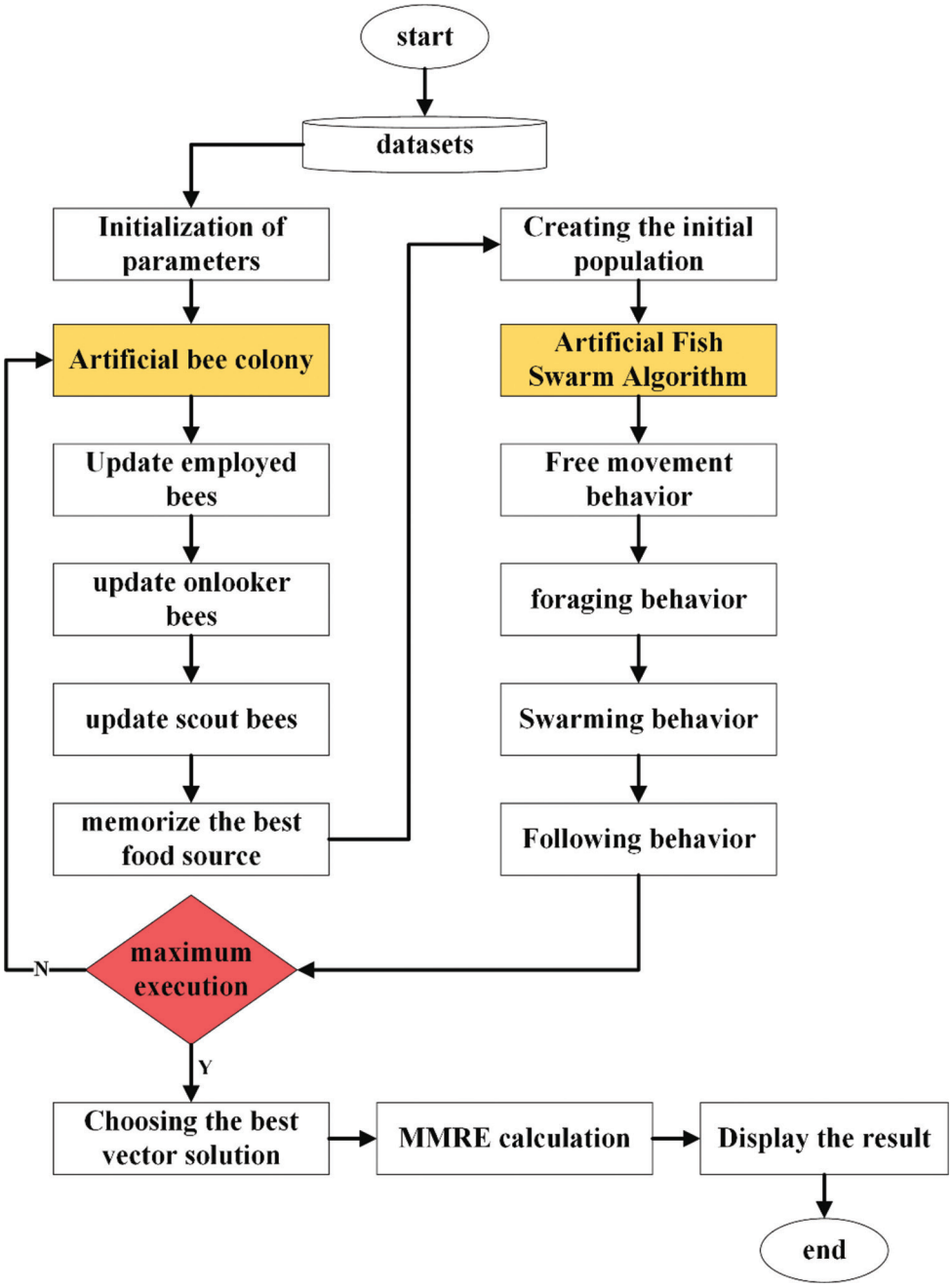

Fig. 1. Proposed method for Software cost estimation.

4.2.2. AFSA algorithm

This algorithm includes the following four behaviors: Free movement, foraging, group movement, and tracking behavior. When fishes fail to find food, they move freely according to Eq. (8). Random search leads to the discovery of all points of the problem space.

$$X(t+1) = X(t) + \text{Step} \times \text{Rand}(-1,1) \quad (8)$$

In Eq. (8), X_i is the i[th] fish's location vector in the D-dimensional space. The uniformly distributed D-dimensional vector of random integers in the range [1 and −1] is produced by the *Rand* function. Food search behavior is defined using Eq. (9). If $X_j > X_i$ (in maximizing issues), the intended artificial fish advances from its present state in the direction of X_j. The food density in X_i is compared with the food density in the current state. In addition, another state, X_j, is chosen using

```
Read the datasets
Initialize the parameters
vector_length = 15
iteration_count = 0
WHILE iteration_count < maximum_execution DO
Perform ABC
Create Initial Population (vector_length, 0.9, 1.4)
For each bee in the population
Update the employed bee's position
Update the onlooker bee's position
Update the scout bee's position
Memorize the best food source
End for end
Perform the AFSA
Initialize the fish swarm
Perform the free movement behavior
Perform the foraging behavior
Perform the swarming behavior
Follow the best food source
Update the fish swarm
Calculate the fitness of each fish
Select the best fish
end
iteration_count = iteration_count + 1
End While
Choose the best vector solution
Calculate the MMRE
Display the result
```

Fig. 2. Pseudocode of the proposed method.

Eq. (8) if $X_j > X_i$ is not the case.

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_j - X_i^{(t)}}{X_j - X_i^{(t)}} \times \text{Step} \times \text{Rand}(0,1) \tag{9}$$

If X_i is the current state of the fish, a state (X_j) is randomly selected in the fish's field of view. The state of X_j is obtained using Eq. (10). The rand function generates a random number with a uniform distribution in the interval [−1, 1].

$$X_j = X_i + \text{Visual} \times \text{Rand}\ (-1,1) \tag{10}$$

The group movement behavior of fishes is defined collectively according to Eq. (11). At this stage, each fish makes an effort to travel toward the central location, or X_C, which corresponds to the center of gravity of the fish group's members' vectors.

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_C - X_i^{(t)}}{X_C - X_i^{(t)}} \times Step \times Rand(0,1) \tag{11}$$

The following behavior is defined according to Eq. (12) that each of the fish is look or the fish that have found more food. In the process of group movement of fish, when a fish or a number of them find food, the neighbors and fishes close to them follow them and quickly reach the food. The update of the current state (X_i) is done by exploring the neighbors (X_j *k*), so the movement toward the optimal points is done by checking the state of the neighbors. The segment step's maximum length is equal to the step. The Euclidean distance $d_{ij} = \|X_i - X_j\|$ denotes the separation between two artificial fish that are in the states X_i and X_j.

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_j - X_i^{(t)}}{X_j - X_i^{(t)}} \times \text{Step} \times \text{Rand}(0,1) \tag{12}$$

4.3. ABC-AFSA

A new hybrid model with the advantages of ABC and AFSA is proposed to increase the discovery capability. The hybrid model leads to the improvement of the convergence rate and the quality of the solutions. The hybrid model creates a new method to find the best optimal solutions. Global exploration and local exploitation are executed in balanced mode. Exploratory search behavior is defined using Eq. (13).

$$X_i^{(t+1)} = \begin{cases} X_i^{(t)} + \frac{X_j - X_i^{(t)}}{X_j - X_i^{(t)}} \times \text{Step} \times \text{Rand}(0,1) & r_1 < 0.5 \\ X_i^{(t)} + \frac{X_{min} - X_i^{(t)}}{X_{min} - X_i^{(t)}} \times \left(1 - \left(\frac{t}{T}\right)^{e1}\right) \times \text{Rand}(0,1) & r_1 \geq 0.5 \end{cases} \tag{13}$$

In Eq. (13), is the current state. The state of is obtained using Eq. (14). The rand function generates a random number with a uniform distribution in the interval [−1, 1]. In the hybrid model, the value of *Step* and *Visual* is equal to 30 and 1, respectively. r_1 is a random number between

0 and 1. X_{min} represents the lowest value in the i^{th} vector. The parameters t and T are the current iteration and the maximum iteration, respectively. e_1 is a positive number between 0 and 1.

$$X_j = \begin{cases} X_i + \text{Visual} \times \text{Rand}\ (-1,1) & r_2 < 0.5 \\ L_i + \text{rand}(0,1) \times (U_i - L_i) & r_2 \geq 0.5 \end{cases} \quad (14)$$

In Eq. (14), U_i and L_i are variables with the upper and lower limits, respectively. *Rand*() is also a function of random numbers in the interval (0,1). r_2 is a random number between 0 and 1.

AFSA has disadvantages such as early convergence and poor local search ability because it cannot use the local information of individuals in the population. ABC can enhance the search ability and avoid getting stuck in local optima. The hybrid model has good global discovery capability as well as local exploitation capability. It can obtain a better solution with a faster convergence speed. The exploitability in the combined model is defined according to Eq. (15). The X_{best} parameter represents the current best value found.

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_{best} - X_i^{(t)}}{X_{best} - X_i^{(t)}} \times ST \times Rand\ (0,1) \quad (15)$$

$$ST = \frac{\text{rand}() \times \text{Visual}}{e^{(\text{rand}() \times \text{Visual})} - e^{(\frac{t}{T} \times \text{Visual})}} \quad (16)$$

4.4. Evaluation Criteria

In the proposed method, the average relative error value [22] is considered as the fitness function. The proposed method's fitness function aims to reduce relative error values when compared to the COCOMO model. Eq. (17) is the definition of the fitness function for the proposed method. The y parameter is equal to the real value and the $\bar{y}$ parameter is equal to the estimated value obtained by the proposed method.

$$\text{Mean Magnitude of Relative Error (MMRE)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left|y_i - \bar{y}_i\right|}{y_i} \times 100\right) \quad (17)$$

$$\text{Mean of Magnitude Error Relative (MMER)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left|y_i - \bar{y}_i\right|}{\bar{y}_i} \times 100\right) \quad (18)$$

$$\text{Mean Squared Error (MSE)} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \quad (19)$$

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (20)$$

$$\text{Mean Absolute Percentage Error (MAPE)} = \sum_{i=1}^{n}\left(\frac{\left|y_i - \bar{y}_i\right|}{y_i}\right) / n \times 100 \quad (21)$$

$$\text{Mean of Absolute Errors (MAE)} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \bar{y}_i\right| \quad (22)$$

$$\text{Median Magnitude of Relative Error (MDMRE)} = \text{Median (MRE)} \quad (23)$$

5. EVALUATION AND RESULTS

In this paper, the SCE was computed using a mix of the ABC algorithm and AFSA, and the results were analyzed using a variety of techniques. The implementation of the hybrid method has been done in the VC#.NET 2021 environment. MMER, MMRE, MDMRE, PRED (0.25), MSE, RMSE, MAPE, and MAE criteria are used to evaluate the proposed method. The evaluation criteria are tested on eight datasets NASA60, NASA63, NASA93, Miyazaki, Maxwell, KEMERER, Desharnais, and Finnish. The initial population and the number of iterations in all algorithms are equal to 50 and 200, respectively. The value of k in the KNN algorithm is equal to 3. According on (Table 2)'s findings, the proposed method performs the best overall on NASA60. In the proposed method, MMRE has a value of 20.08. The MMRE value for COCOMO, ABC, AFSA, and KNN models is 16.09, 19.17, 16.17, and 14.52, respectively. The value of PRED for COCOMO, ABC, AFSA, and KNN models is 0.83, 0.75, 0.92, and 0.92, respectively.

According to Table 3's findings, the proposed method performs the best overall on NASA63 and MMRE has a value of 22.63. The MMRE value for COCOMO, ABC, AFSA, and KNN models is 16.60, 18.16, 15.24, and 15.16, respectively. The value of PRED for COCOMO, ABC, AFSA, and KNN models is 0.85, 0.77, 0.85, and 0.92, respectively.

TABLE 2: Evaluation of the proposed method on the NASA60 dataset

Function	COCOMO	ABC	AFSA	KNN	Proposed method
MMRE	19.43	16.03	16.92	16.67	16.02
MMRE	16.09	19.17	16.17	14.52	20.08
MDMRE	16.27	15.44	16.03	14.68	12.60
PRED	0.83	0.75	0.92	0.92	0.75
MSE	8521.40	5475.48	11570.14	6995.79	3408.14
RMSE	92.31	74.00	107.56	83.64	58.38
MAPE	16.09	19.17	16.17	14.52	20.08
MAE	50.34	42.55	59	43.30	39.24

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model, MAE: Mean absolute error

TABLE 3: Evaluation of the proposed method on the NASA63 dataset

Function	COCOMO	ABC	AFSA	KNN	Proposed method
MMRE	20.20	14.32	14.30	17.66	22.72
MMRE	16.60	18.16	15.24	15.16	22.63
MDMRE	16.27	10.33	14.04	14.68	16.97
PRED	0.85	0.77	0.85	0.92	0.77
MSE	8041.41	3925.67	6617.66	6633.15	16996.64
RMSE	89.67	62.66	81.35	81.44	130.37
MAPE	16.60	18.16	15.24	15.16	22.63
MAE	50.14	36.50	06.46	43.64	76.50

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model, MAE: Mean absolute error

TABLE 4: Evaluation of the proposed method on the NASA93 dataset

Function	COCOMO	ABC	AFSA	KNN	Proposed method
MMRE	23.10	12.28	38.90	21.27	15.65
MMRE	18.23	14.23	25.13	17.24	16.26
MDMRE	20.98	11.46	26.39	18.18	12.68
PRED	0.84	0.89	0.53	0.89	0.89
MSE	5933.62	3758.65	2255.6	4970.08	7518.1
RMSE	77.03	61.31	47.49	70.5	86.71
MAPE	18.23	14.23	25.13	17.24	16.26
MAE	44.49	34.32	36.68	40.04	50.06

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model, MAE: Mean absolute error

According to (Table 4)'s findings, the proposed method performs the best overall on NASA93 and MMRE has a value of 16.26. The MMRE value for COCOMO, ABC, AFSA, and KNN models is 18.23, 14.23, 25.13, and 17.24, respectively. The value of PRED for COCOMO, ABC, AFSA, and KNN models is 0.84, 0.89, 0.53, and 0.89, respectively.

According to (Table 5)'s findings, the proposed method performs the best overall on Miyazaki and MMRE in the proposed method is 31.22. The MMRE value for COCOMO, ABC, AFSA, and KNN models is 272.79, 33.52, 33.09, and 272.79, respectively. The value of PRED for ABC and AFSA models is 0.40 and 0.40, respectively.

According to (Table 6)'s findings, the proposed method performs the best overall on MAXWELL and MMRE in the proposed method is 37.9. The MMRE value for COCOMO, ABC, AFSA, and KNN models is 76.43, 38.1, 37.81, and 43.76, respectively. The value of PRED for ABC and AFSA models and the proposed method is 0.46, 0.46, and 0.52, respectively.

According to (Table 7)'s findings, the proposed method performs the best overall on KEMERER and MMRE in the proposed method is 56.34. The MMRE value for COCOMO, ABC, AFSA, and KNN models is 687.64, 48.1, 45.8, and 687.64, respectively. The value of PRED for COCOMO, ABC, AFSA, and KNN models was obtained as 0, 0.44, 0.44, and 0 respectively.

According to (Table 8)'s findings, the proposed method performs the best overall on Desharnais and MMRE in the proposed method is 48.99. The MMRE value for COCOMO, ABC, AFSA, and KNN models is 99.11, 70.27, 1.71, and

TABLE 5: Evaluation of the proposed method on the Miyazaki dataset

Function	COCOMO	ABC	AFSA	KNN	Proposed method
MMRE	70.83	58.9	57.86	70.83	52.21
MMRE	272.79	33.52	33.09	272.79	31.22
MDMRE	277.48	41.04	40.52	277.48	37.46
PRED	0	0.40	0.40	0	0.50
MSE	18960.41	670.18	650.89	18960.41	555.1
RMSE	137.7	25.89	25.51	137.7	23.56
MAPE	272.79	33.52	33.09	272.79	31.22
MAE	122.16	20.71	20.49	122.16	19.22

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model, MAE: Mean absolute error

TABLE 6: Evaluation of the proposed method on the Maxwell dataset

Function	COCOMO	ABC	AFSA	KNN	Proposed method
MMRE	118.44	94.06	97.42	118.44	91.6
MMRE	76.43	38.1	37.81	43.76	37.9
MDMRE	36.58	34.67	31.31	36.58	32.93
PRED	0.38	0.46	0.46	0.38	0.52
MSE	14513588.9	21537730.41	23327134.14	14513588.9	22634329.41
RMSE	3809.67	4640.88	4829.82	3809.67	4757.55
MAPE	43.76	38.1	37.81	43.76	37.9
MAE	2603	2915.07	3031.72	2603	2964.61

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model, MAE: Mean absolute error

TABLE 7: Evaluation of the proposed method on KEMERER data set

Function	COCOMO	ABC	AFSA	KNN	Proposed method
MMRE	79.23	56.36	58.66	79.23	60.2
MMRE	687.64	48.1	45.8	687.64	56.34
MDMRE	626.6	39.5	31.2	626.6	39.29
PRED	0	0.44	0.44	0	0.44
MSE	3546018.29	68399.11	68923.45	3546018.29	3546018.29
RMSE	1883.09	261.53	262.53	1883.09	268.95
MAPE	687.67	48.1	45.8	687.64	56.34
MAE	1375.95	120.31	119.9	1375.95	130.43

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model, MAE: Mean absolute error

TABLE 8: Evaluation of the proposed method on the Desharnais dataset

Function	COCOMO	ABC	AFSA	KNN	Proposed method
MMRE	13882.49	559	31.562	29108.17	127.12
MMRE	99.11	70.27	1.71	99.35	48.99
MDMRE	99.28	72.29	01.74	99.6	51.9
PRED	0	0.06	0	0	0.12
MSE	19363264.14	9051856.53	9395487.52	176906010.58	86350530.51
RMSE	4400.37	3008.63	3065.21	13300.6	9292.5
MAPE	99.11	70.27	1.71	99.35	48.99
MAE	3879.66	2665.57	2701.35	10642.89	6638.94

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model, MAE: Mean absolute error

99.35 respectively. The value of PRED for ABC models and the proposed method is 0.06 and 0.12, respectively.

The results of (Table 9) show that the proposed method has the best overall performance on Finnish. The value of MMRE in the proposed method is 48.82. The MMRE value for COCOMO, ABC, AFSA, and KNN models is 99.35, 55.75, 57.27, and 99.31, respectively. The value of PRED for ABC models and the proposed method is 0.12 and 0.12, respectively.

TABLE 9: Evaluation of the proposed method on the Finnish dataset

Function	COCOMO	ABC	AFSA	KNN	Proposed method
MMRE	29108.17	174.89	217.89	29108.17	127.12
MMRE	99.35	55.75	57.27	99.31	48.82
MDMRE	99.6	62.22	66.51	99.6	51.9
PRED	0	0.12	0.12	0	0.12
MSE	176906010.58	98437723.32	110721304.73	176906010.58	86350530.51
RMSE	13300.6	9921.58	10522.42	13300.6	9292.5
MAPE	99.35	55.75	57.27	99.35	48.99
MAE	1064.89	7358.82	7783.96	1064.89	6638.94

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model, MAE: Mean absolute error

TABLE 10: Comparison of the proposed model with meta-heuristic algorithms

Datasets	WOA	SCA	MFO	Proposed method
NASA60	21.52	20.48	21.94	20.08
NASA63	22.91	22.19	23.15	22.63
NASA93	16.54	16.82	17.24	16.26
Miyazaki	31.42	31.49	31.12	31.22
Maxwell	38.19	38.14	38.18	37.90
KEMERER	56.71	56.22	56.92	56.34
Desharnais	49.05	49.16	49.08	48.99
Finnish	48.97	49.02	48.95	48.82

WOA: Whale optimization algorithm, SCA: Sine cosine algorithm, MFO: Moth-Flame Optimization

To show the effectiveness of the proposed model, WOA [23], sine cosine algorithm (SCA) [24], and Moth-Flame Optimization (MFO) [25] have been used for comparison. The results of (Table 10) show that the SCA algorithm on NASA63 achieved an error rate of 22.19. The MFO algorithm on Miyazaki achieved an error rate of 31.12. The SCA algorithm on KEMERER has achieved an error rate of 56.22. In general, the proposed model has achieved a lower error value in most of the data sets. Of course, meta-heuristic algorithms have different results and the error value changes with each execution. The most important issue in meta-heuristic algorithms is the balance between exploration and exploitation, which is established in the proposed model.

In Table 11, the results of the models are shown based on the best, mean, and worst criteria. The values of the best, mean, and worst criteria on the NASA93 dataset for the proposed model are 16.26, 16.38, and 16.74, respectively. Furthermore, the values of the best, mean, and worst criteria on the Desharnais dataset for the proposed model are 48.99, 49.07, and 49.21, respectively. The worst value on the Miyazaki and Maxwell datasets for AFSA is 33.65 and 38.34, respectively.

The statistical results are confirmed using a non-parametric statistical test called Mann–Whitney U-test [26]. Statistical tests are used to evaluate the results of the models statistically and meaningfully. For this purpose, the Mann–Whitney U-test has been used, which is a non-parametric statistical test. In Table 12, the results of the statistical test are shown that the models are compared with each other to determine whether the difference in the average errors for the combined model is significant or not. The hybrid model has a lower error value compared to other models.

6. CONCLUSION AND FUTURE WORK

SCE is one of the most challenging project management duties since project planning and budgeting are dependent on realized costs. Since there are a limited number of resources for a project, it is not possible to include all the required resources in the final product. In this paper, the combined method of ABC and AFSA was used for SCE. The proposed method was tested on NASA60, NASA63, NASA93, Miyazaki, Maxwell, KEMERER, Desharnais, and Finnish datasets. The proposed method performed better on the NASA60 dataset. In the NASA93 data set, MMER, MMRE, MDMRE, PRED, MSE, and MAPE criteria have performed better. In the Finnish dataset, better performance is obtained in all measures except MAE. The main challenges of this study are as follows: (1) the data sets used are old. In old projects, all the factors involved in the construction of software projects are usually not mentioned. For example, current databases filter and process data using specific commands. Processing orders are specialized in a field and require specialized manpower. (2) Meta-heuristic algorithms for estimating software projects have a main limitation called local optimality. With the occurrence of local optimality, the algorithm is not able to find optimal values for the effort factors, and therefore, the error value increases. To solve the mentioned problems, weight and adaptive operators should be used in meta-heuristic algorithms. According to the obtained results, for future works, the combined method can

TABLE 11: The results of the models based on the best, mean, and worst criteria

Datasets	MMRE	COCOMO	ABC	AFSA	KNN	Proposed method
NASA60	Best	16.09	19.17	16.17	14.52	20.08
	Mean	16.09	19.32	16.29	14.30	20.15
	Worst	16.09	19.65	16.67	14.92	20.42
NASA63	Best	16.60	18.16	15.24	15.16	22.63
	Mean	16.60	18.25	15.38	15.33	22.91
	Worst	16.60	18.55	15.64	15.76	23.15
NASA93	Best	18.23	14.23	25.13	17.24	16.26
	Mean	18.23	14.81	25.67	17.48	16.38
	Worst	18.23	14.69	25.84	17.65	16.74
Miyazaki	Best	272.79	33.52	33.09	272.79	31.22
	Mean	272.79	33.84	33.24	273.61	31.35
	Worst	272.79	34.15	33.65	273.54	31.50
Maxwell	Best	76.43	38.10	37.81	43.76	37.90
	Mean	76.43	38.54	37.85	43.66	38.05
	Worst	76.43	38.64	38.34	44.21	38.12
KEMERER	Best	687.64	48.10	45.80	687.64	56.34
	Mean	687.64	48.23	46.20	688.11	56.46
	Worst	687.64	48.54	46.25	688.94	56.82
Desharnais	Best	99.11	70.27	1.71	99.35	48.99
	Mean	99.11	70.31	2.19	99.36	49.07
	Worst	99.11	70.56	2.84	99.49	49.21
Finnish	Best	99.35	55.75	57.27	99.31	48.82
	Mean	99.35	56.02	57.61	99.25	48.92
	Worst	99.35	56.16	57.94	99.51	49.25

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony, MMRE: Mean magnitude of relative error, COCOMO: Constructive cost model

TABLE 12: The results of the Mann–Whitney U statistical test

Datasets	ρ value			
	ABC	AFSA	KNN	Proposed method
NASA60	2.362215e-16	3.257400e-12	5.000036e-14	3.261500e-12
NASA63	2.320164e-15	2.326010e-12	2.259487e-12	1.323265e-20
NASA93	4.002649e-12	2.993584e-15	4.013265e-14	3.265148e-15
Miyazaki	2.010062e-14	4.234901e-15	2.103284-16	1.003261e-17
Maxwell	2.360014e-10	2.003947e-14	4.169500e-10	3.269501e-12
KEMERER	6.206554e-10	5.205974e-17	1.249014e-10	5.265491e-10
Desharnais	2.360057e-12	3.123500e-10	3.839142e-10	4.326514e-12
Finnish	3.142510e-14	3.320115e-12	5.760258e-12	2.365214e-15

AFSA: Artificial fish swarm algorithm, ABC: Artificial bee colony

be used for various problems in such fields as weather forecasting, disease forecasting, stock market forecasting, urban growth forecasting, license plate recognition, image recognition, etc.

REFERENCES

[1] S. Hameed, Y. Elsheikh and M. Azzeh. "An optimized case-based software project effort estimation using genetic algorithm." *Information and Software Technology*, vol. 153, p. 107088, 2023.

[2] N. Govil and A. Sharma. "Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors." *Advances in Engineering Software*, vol. 172, p. 103209, 2022.

[3] H. M. Sneed and C. Verhoef. "Cost-driven software migration: An experience report." *Journal of Software: Evolution and Process*, vol. 32, no. 7, p. e2236, 2020.

[4] S. Tariq, M. Usman and A. C. Fong. "Selecting best predictors from large software repositories for highly accurate software effort estimation." *Journal of Software: Evolution and Process*, vol. 32, no. 10, p. e2271, 2020.

[5] K. Rak, Ž. Car and I. Lovrek. "Effort estimation model for software development projects based on use case reuse." *Journal of Software: Evolution and Process*, vol. 31, no. 2, p. e2119, 2019.

[6] T. Hacaloglu and O. Demirors. "An exploratory case study using events as a software size measure." *Information Technology and Management*, vol. 24, pp. 1-20, 2023.

[7] E. Feizpour, H. Tahayori and A. Sami. "CoBRA without experts: New paradigm for software development effort estimation using COCOMO metrics." *Journal of Software: Evolution and Process*,

vol. 35, p. e2569, 2023.

[8] A. K. Bardsiri and S. M. Hashemi. "A differential evolution-based model to estimate the software services development effort." *Journal of Software: Evolution and Process*, vol. 28, no. 1, pp. 57-77, 2016.

[9] X. L. Li, Z. J. Shao and J. X. Qian. "An optimizing method based on autonomous animats: Fish-swarm algorithm." *Systems Engineering-Theory and Practice*, vol. 22, no. 11, pp. 32-38, 2002.

[10] D. Karaboga. "*An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-tr06, Erciyes University, Engineering Faculty, Computer*." Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri/Türkiye, 2005.

[11] P. K. Sethy and S. Rani. "*Improvement in COCOMO Modal Using Optimization Algorithms to Reduce MMRE Values for Effort Estimation. In: 2019 4th International Conference on Internet of Things: Smart Innovation and Usages* (*IoT-SIU*)." pp. 1-4, 2019.

[12] P. Rijwani and S. Jain. "Enhanced software effort estimation using multi layered feed forward artificial neural network technique." *Procedia Computer Science*, vol. 89, no. 1, pp. 307-312, 2016.

[13] N. Rankovic, D. Rankovic, M. Ivanovic and L. Lazic. "Improved effort and cost estimation model using artificial neural networks and taguchi method with different activation functions." *Entropy*, vol. 23, p. 854, 2021.

[14] Z. Shahpar, V. K. Bardsiri and A. K. Bardsiri. "Polynomial analogy-based software development effort estimation using combined particle swarm optimization and simulated annealing." *Concurrency and Computation: Practice and Experience*, vol. 33, no. 20, p. e6358, 2021.

[15] S. W. Ahmad and G. R. Bamnote. "Whale-crow optimization (WCO)-based Optimal Regression model for Software Cost Estimation." *Sādhanā*, vol. 44, no. 4, p. 94, 2019.

[16] A. Puspaningrum and R. Sarno. "A hybrid cuckoo optimization and harmony search algorithm for software cost estimation." *Procedia Computer Science*, vol. 124, no. 1, pp. 461-469, 2017.

[17] 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.

[18] S. P. Singh, G. Dhiman, P. Tiwari and R. H. Jhaveri. "A soft computing based multi-objective optimization approach for automatic prediction of software cost models." *Applied Soft Computing*, vol. 113, no. 1, p. 107981, 2021.

[19] U. Aman, W. Bin, S. Jinfang, L. Jun, A. Muhammad and S. Zejun. "Optimization of software cost estimation model based on biogeography-based optimization algorithm." *Intelligent Decision Technologies*, vol. 14, no. 1, pp. 441-448, 2020.

[20] S. K. Gouda and A. K. Mehta. "A new evolutionism based self-adaptive multi-objective optimization method to predict software cost estimation." *Software: Practice and Experience*, vol. 52, no. 8, pp. 1826-1848, 2022.

[21] B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer and B. Steece. "*Software Cost Estimation with COCOMO II*." Prentice Hall Press, Upper Saddle River, NJ, USA, 2009.

[22] R. H. Martin and D. Raffo. "*A Comparison of Software Process Modeling Techniques. In: Innovation in Technology Management: The Key to Global Leadership, PICMET'*97." pp. 577-580, 1997.

[23] S. Mirjalili and A. Lewis. "The whale optimization algorithm." *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.

[24] S. Mirjalili. "SCA: A Sine Cosine algorithm for solving optimization problems." *Knowledge-Based Systems*, vol. 96, pp. 120-133, 2016.

[25] S. Mirjalili. "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm." *Knowledge-Based Systems*, vol. 89, pp. 228-249, 2015.

[26] Z. Abdelali, H. Mustapha and N. Abdelwahed. "Investigating the use of random forest in software effort estimation." *Procedia Computer Science*, vol. 148, no. pp. 343-352, 2019.