IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 2, 2020, accepted April 14, 2020, date of publication April 20, 2020, date of current version May 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988867

Software Cost Estimation Based on Dolphin Algorithm

ANFAL A. FADHIL, RASHA G. H. ALSARRAJ, AND ATICA M. ALTAIE
College of Computer Science and Mathematics, University of Mosul, Mosul 41001, Iraq

Corresponding author: Anfal A. Fadhil (anfalaaf@uomosul.edu.iq)

ABSTRACT Accurate Software Effort Estimation is of high importance with regard to Software Project Management. It can be specified as the process for predicting Effort regarding costs, needed for developing software products. A lot of techniques related to software effort estimation were carried out for developing models that are generating optimal estimation accuracy. Swarm intelligence is one such technique. The process-related in selecting the optimum estimation algorithm is expert dependent and complex. The presented study optimizes the estimation using the COCOMO II models by two models: the first model applied the dolphin algorithm, the second model applied suggested hybrid dolphin and bat algorithm (DolBat). By applying the two models on two data set and evaluate with the use of Magnitude of Relative Error(MRE) and Mean Magnitude of Relative Error(MMRE). The results indicate that the dolphin algorithm has better than previous algorithms but the (DolBat) is the best to get the coefficient value of the COCOMO II model.

INDEX TERMS Bat algorithm, COCOMO II model, dolphin algorithm, effort estimation, echolocation, NASA project dataset.

I. INTRODUCTION

Software cost estimation is a process of high significance since it is essential for estimating the project's costs at the initial phase. Cost estimation has needed for computing resources and the budget needed for a project [1] and [2]. Costs have related to project depend upon efforts achieved, which involve the number of reviews efficiency throughout implementing and predevelopment processing, and so on [3]. There have applied two significant types of estimation techniques Model-Based and Expert-Based techniques. The first one has established depending on mathematical models, whereas the second one has based on human guidance. There are related certain examples of approaches, like COCOMO II [4] FUNCTION POINT ANALYSIS [5], USE CASE POINT [6], and [7] as well as others. All presented estimation techniques face the problem of the absence of accuracy. The majority of such approaches are focusing on is certain aspects associated with the process of software development while ignoring others. Furthermore, most such approaches are traditional and ineffective concerning IDEs, paradigms, programming languages, in addition to other tools of development. A few adjustment factors are regarding such techniques might be biased toward certain processes of development and work environment or cultures [8]. Swarm intelligence can be considered as the main topic in AI. A lot of algorithms have developed according to the properties and definitions related to swarm intelligence, such as algorithms might be divided into two categories, the first one has considered as the process algorithm which has based on a genetic algorithm that majorly imitates population evolution's process, the second one is the bionic behavior algorithm which is simulating behavior model regarding various species looking for preys. They use behavior rules related to individuals as well as interactions between the individuals for producing certain changes at the group level in addition to achieving specific aims such as bat and dolphin algorithms [9]. This work optimizes the estimation with the use of the COCOMO II model through utilizing two models: the first model has applied the dolphin algorithm, the second model has applied the hybrid dolphin and bat algorithm (DolBat). By applying the two models to two data set taken from the promise software engineering data set. These data are NASA-93 and NASA-60 [10]. The projects of data set consist of (KLOC) thousand source lines of code, actual effort. The cost drivers rating from extremely low to extremely high are also present

The associate editor coordinating the review of this manuscript and approving it for publication was Nadeem Iqbal.

in data set software projects. 15 software projects have been selected from NASA-93 and NASA-60 randomly. By using metrics of evaluation MRE and MMRE, the result show that the suggested DolBat is the best optimization of COCOMO II model coefficients (a, b) values. Furthermore, the current study is classified into eight sections. Section II explains some of the related work, whereas section III describes the COCOMO II model, as for section IV, it points out the dolphin algorithm. In its turn, section V will shed light on the BAT algorithm. Section VI interprets the Dolphin Bat Algorithm (DOLBAT) proposed to estimate software effort. Finally, the results will be given by section VII besides the corresponding discussion. The conclusions and future works have shown in section VIII.

II. REKATED WORK

Software cost estimation has a critical function in the management of the project. Wrong results may bring about over-estimating or underestimating effort, which can have severe consequences on the resources of the project [11]. It is worth mentioning that different studies have utilized various meta-heuristic algorithms to improve the performance related to the COCOMO II model. A study conducted by [12] utilized a genetic algorithm to optimize parameters related to the COCOMO II model, whereas the study [13] utilized the Bat algorithm for estimating software project's costs. The study [1] utilized a differential evolution algorithm for the same purpose. The study [14] utilized a hybrid Cuckoo and Harmony algorithm for predicting software costs. The author [15] uses a hybrid Bat and Grautational algorithm to estimate the effort of software. The author [11] use regression fuzzy models for prediction software estimation effort. It can be stated that in the present study. The study [16] genetic algorithms have used for optimizing the COCOMO-II post architecture model by tuning its four coefficients. This tuning has followed to improve the accuracy of predicting both the effort and the development time. From her part, NASA has conducted the experiments based on data set to compare the accuracy of the optimized coefficient with current COCOMO-II coefficients. The other study which has conducted by [17] proposed a method by which it can analyze datasets, automatically and building estimation models with different techniques of machine learning, and evaluating and comparing their outcomes for a particular dataset to find the model, that produces the most precise estimations. They have been applied in a specific experimental context with the use of nine different estimation techniques. The proposed approach to automated model selection tries to combine the correlation coefficients, Bayesian information criterion, and PRED measures.

In this paper, we used swarm algorithms for optimizing the COCOMO-II by tuning its two coefficients to improve the accuracy of predicting the effort of software development. Experiments were conducted to compare the accuracy of the optimized coefficient with current COCOMO-II coefficients on two NASA data set.

III. COCOMO II MODEL

The Constructive Cost Model has been used for determining software costs, it has created via Boehm depending on empirical analysis regarding software development projects [4]. Such a model utilizes a straightforward regression formula to estimate costs, efforts, and schedules regard to the software projects depending on the project's features [18] and [19] Such a model might be enhanced with the use of a meta-heuristic algorithm [20]. Various studies utilized different meta-heuristic algorithms for improving the performance related to this model. The model has specified in 3 modes: Organic (2-50 KLOC), Semidetached (50-300 KLOC), and Embedded (>300 KLOC). All these modes are corresponding to the complexity level which has related to code, team size, innovation level of the project, environment strictness, and final volatility [21]. The Intermediate model will be estimating the software development effort concerning man-month through taking into account code size, which has determined through thousands of code lines. The model is adjusting estimation value with the use of experimental constants A and B. Project attributes which have specified through 15 drivers that are corresponding to a set of numerical values referred to as multipliers. The Estimated effort regarding COCOMO II has determined with the use of Eq.1 [22].

$$\text{Effort} = A*(Size)^{B}* \prod_{i=1}^{15} EMi \tag{1}$$

where Size in kilo line of code KLOC and $\mathrm{Em_i}$ representing 15 effort multipliers, whereas $\pi_{i=1}^{15}$ the result of their multiplication. Model multipliers representing: computer, product, personnel, and project factors. Values regarding driver multipliers have allocated depending on 6 categories scale:Very Low, Low, Nominal, High, Very High, and Extra High. The exact values are given in Table 1 [8].

IV. DOLPHIN SWARM ALGORITHM (DSA)

Dolphins are very smart animals. Also they have significant living habits and bio-logical characteristics, some of them are as follows [23]:

1) Echolocation: this is a distinctive ability used by dolphins in their search for prey. They have the ability to make sounds and estimating shape, distance, as well as the location related to prey depending on echo intensity. By using echo, dolphins will have excellent perception with regard to the environment around them [24].

2) Cooperation and division of labor: With regard to the majority of cases, the predatory behavior isn't accomplished via just a single dolphin, yet via joint attempts regarding a lot of dolphins via division of labor and co-operation. There is a certain division of labor between dolphins. For example, the dolphin that is close to prey is responsible to track the prey's movement, while dolphins far from prey will surround the prey via creating a circle.

3) Information exchanges this is another property related to dolphins, since they have the ability to express various ideas with the use of sounds at various frequencies, also

TABLE 1. The numeric values of the effort multipliers.

Effort Multipliers	Very Low	Low	Normal	High	Very High	Extra High
Required Software Reliability	0.75	0.88	1	1.15	1.4	
Database Size	0.94	1	1.08	1.16	-1.23	
Process Complexity	0.70	0.85	1	1.15	1.3	1.65
Time Constraint for CPU	1	1.11	1.3	1.66	-1.3	1.66
Main Memory Constraint	1	1.06	1.21	1.56	-1.21	1.56
Machine Volatility	0.87	1	1.15	1.3	-1.49	
Turnaround Time	0.87	1	1.07	1.15	-1.32	
Analysts Capability	1.46	1.19	1	0.86	0.71	
Application Experience	1.29	1.13	1	0.91	0.82	
Programmers Capability	1.42	1.17	1	0.86	0.7	
Virtual Machine Experience	1.21	1.1	1	0.9	1.34	
Language Experience	1.14	1.07	1	0.95	1.2	
Modern Programming practices	1.24	1.1	1	0.91	0.82	
Use of Software Tool	1.24	1.1	1	0.91	0.83	
Schedule Constraint	1.23	1.08	1	1.04	1.1	

they have a distinctive language system. With regard to the predatory process, particularly within a division of labor and co-operation, the capability to exchange information is often utilized for calling other dolphins and updating the prey's location. After using the exchanged information, the dolphin might take excellent actions for making predation more efficient [9].

A. PHASES OF DOLPHIN SWARM ALGORITHMA

DSA has a lot of phases, such as initialization, search, reception, call, and predation, include the predatory process of dolphins, and such habits and characteristics are helping dolphins achieving their goal throughout the process of predatory. The algorithm of dolphin swarm is conforming to thoughts regarding swarm intelligence, yet they are different from conventional algorithms of swarm intelligence. Depending on swarm intelligence, a specific number of dolphins is needed for simulating living habits and biological characteristics indicated in the actual predatory process of dolphins [25]. DSA might be divided into five phases as follows:

1) Initialization phase: evenly and randomly generating initial dolphins swarm, $Dol_i = [x_1, x_2, \ldots, x_D]$ T $(i = 1, 2, \ldots, N)$ where N denoted as number of dolphins. x_j represent component regarding each one of the dimensions which will be optimized. With regard to each dolphin Dol_i, there are two corresponding variables:

- L_i is an optimal solution that Dol_i find in a single time.
- K_i is the neighborhood optimal solution.

where $i = 1, 2, \ldots, N$. After initialize dolphin value, calculate fitness for each one of the dolphins and obtain Fit_k. $Fit_k = \{F\,it_{k,1}, Fit_{k,2}, \ldots, F\,it_{k,N}\}$

2) Search Phase: In this phase, every one of the dolphins performs a search of its surrounding area through emitting sounds in M random directions. Similarly, the sound can be represented in the following way: $V_i = [v_1, v_2, \ldots, v_D]^T$, $(i = 1, 2, \ldots, M)$ in the present research, M stands for a number of the sounds and v_j, $(j = 1, 2, \ldots, D)$ represents the component of every one of the dimensions, which is the sound's attribute of direction. Moreover, the sounds satisfy $||V_i|| = speed$, $(i = 1, 2, \ldots, M)$, "speed" represents a constant which is equivalent to the sound's attribute of the speed. For the sake of preventing dolphins from being stuck in this phase, a maximal search time T1 is set. In the maximal search time T1, sound V_j which Dol_i, $(i = 1, 2, \ldots, N)$ produces in time t is going to search for a new solution X_{ijt}, that may be expressed in the following form:

$$X_{ijt} = Dol_i + V_i * t \tag{2}$$

For the new solution X_{ijt} that Dol_i obtains, its fitness E_{ijt} may be computed using the following formula:

$$E_{ijt} = Fit(X_{ijt}) \tag{3}$$

$$if\ (E_{iab} = min_{j=1,2,\ldots,M;t=1,2,\ldots,T1} E_{ijt}) \tag{4}$$

In this case, the individual optimum solution L_i of Dol_i will be specified as:

$$L_i = X_{iab} \tag{5}$$

$$if\ \ (Fitness\,(L_i) < Fitness\,(K_i)) \tag{6}$$

then K_i is replaced by L_i; otherwise, K_i does not change. After all the Dol_i $(i = 1, 2, \ldots, N)$ update their L_i and K_i (in the case where they may be updated), DSA enters call phase.

3) Call Phase In this phase, every one of the dolphins produces sounds for informing other dolphins of the results in the search phase, which includes whether a more sufficient solution has been obtained and the location of that better solution. The matrix of the transmission time TS where $TS_{i,j}$ stands the rest of the time for sound to travel from Dol_j to Dol_i and requires being updated in the following way: For K_i, K_j, and $TS_{i,j}$

$$\begin{aligned} &if\ (Fitness\,(K_i) < Fitness\,(K_j)\big) \\ &and\ TS_{i,j} > \left\lceil \frac{DD_{ij}}{A * speed} \right\rceil \\ &then\ TS_{i,j} = \left\lceil \frac{DD_{i,j}}{A.speed} \right\rceil \end{aligned} \tag{7}$$

Otherwise T $S_{(i,j)}$ remain it's value. Where (i = 1, 2, ..., N; j = 1, 2, ..., N) and $DD_{i,j}$ is the distance between Dol_i and Dol_j.

$$DD_{ij} = \left\|Dol_i - Dol_j\right\|, \quad i, j = 1, 2, \ldots N, \ i \neq j \tag{8}$$

Speed represents a constant which is equivalent to the sound's attribute of speed. A stands for a constant which represents the acceleration that is capable of making the sounds travel at a higher speed in the case of quite low speed, then, $TS_{i,j}$ will undergo updating based on the Eq. 7.

4) Reception phase In the DSA, the process of the exchange (which includes call and reception phases) will be maintained with the TS, in the case where the DSA enters the phase of the reception, every term $TS_{i,j}$ (i = 1, 2, ..., N; j = 1, 2, ..., N), Then TS decreases by 1 for the sake of indicating that sounds propagate over 1 unit of time. In this case, the DSA requires checking each one of the terms $TS_{i,j}$ in a matrix, and

$$if \ (TS_{i,j} = 0) \tag{9}$$

Meaning that sound which is transmitted from Dol_j to Dol_i may be obtained by Dol_i, in this case where is a need for replacing the $TS_{i,j}$ by new time term which is referred to as the "maximum transmission time" (T2), for indicating that the equivalent sound was received. Performing a comparison of K_i and K_j,

$$if \ (Fitness\,(K_i) > Fitness\,(K_j)) \tag{10}$$

in this case, K_i will be replaced with K_j; else, K_i remains unchanged. After each term in the matrix TS which satisfies Eq. 9 is handled, DSA begins the predation phase.

5) Predation phase In this phase, each one of the dolphins is required to compute the encircling radius R2, determining a distance between the optimum solution of the neighborhood of the dolphin and its position following the phase of predation based on the available data, and afterward, obtains a new position. For every one of the dolphins, the following is computed:

a) distance DK:

$$DK_i = \|Dol_i - K_i\|, \quad i = 1, 2, \ldots N \tag{11}$$

b) distance DKL:
c)

$$DKL_i = \|L_i - K_i\|, \quad i = 1, 2, \ldots N \tag{12}$$

R1: represents the radius of the search, representing the maximal search phase range, may be computed based on the following equation:

$$R1 = T1 \times speed \tag{13}$$

In general, calculating the encircling radius R2 and dolphin's position update has to be discussed in three cases. a)

$$\begin{pmatrix} if \ (DK_i \leq R_1) \\ Then \ R_2 = \left(1 - \frac{2}{e}\right) DK_i \end{pmatrix} \tag{14}$$

$$newDol_i = K_i + \frac{Dol_{i-K_i}}{DK_i} R_2 \tag{15}$$

b)

$$\begin{pmatrix} If \ (DK_i > R_1 \ and \ DK_i \geq DKL_i) \\ Then \ R_2 = \left(1 - \frac{\frac{DK_i}{Fitness(K_i)} + \frac{DK_i - DKL_i}{Fitness(L_i)}}{e.DK_i \frac{1}{Fitness(K_i)}}\right) DK_i \end{pmatrix} \tag{16}$$

$$newDol_i = K_i + \frac{Random}{\|Random\|} R2 \tag{17}$$

c)

$$\begin{pmatrix} If \ (DK_i < DKL_i) \\ Then \ R_2 = \left(1 - \frac{\frac{DK_i}{Fitness(K_i)} - \frac{DKL_i - DK_i}{Fitness(L_i)}}{e.DK_i \frac{1}{Fitness(K_i)}}\right) DK_i \end{pmatrix} \tag{18}$$

Calculate $newDol_i$ as Eq. 17, where e represents a constant which is greater than 2. After Dol_i moves to the position newDoli, comparing $newDol_i$ with K_i concerning fitness,

$$Fitness(newDol_i) < Fitness(k_i) \tag{19}$$

then K_i is replaced by newDolı; otherwise, K_i does not change. After all the Dol_i (i = 1, 2, ..., N) update their locations and K_i (in the case where it may be updated), specify if the DSA satisfies the end condition. In the case

where the end condition has been met, DSA begins the phase of the termination. Else, DSA begins the search phase once more [9], and [26].

V. BAT ALGORITHM

In the year of 2010, Yang has presented a new algorithm of optimization, which is referred to as the Bat Algorithm (BA), according to the swarm intelligence and inspiration form the observation [27]. microbat's behaviors and characteristics. Three main properties of the microbat have utilized for the construction of the main Bat Algorithm structure, have listed below [28]:

a) the echolocation behavior: the majority of bat species use the echolocation for detecting the prey.
b) the frequency: the microbat transmits a fixed frequency fmin with a variable wave-length λ and loudness A0 for searching for the prey.
c) the loudness: the loudness which is presumed to vary from a positive large A0 to a minimal constant value, represented as Amin.

The virtual bat's movement is regulated based on Eq.20 - Eq. 22:

$$f_i = f_{min} + (f_{max} - f_{min})\,\beta \tag{20}$$

where firepresents the frequency which is utilized by a bat which seeks for the prey, suffixes, max, and min, represent the maximum and minimum values, respectively, β is an arbitrary vector, drawn from a uniform distribution, and $\beta \in [0, 1]$.

$$v_i^t = v_i^{t-1} + \left(x_i^t - x_{best}\right) f_i \tag{21}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{22}$$

TABLE 2. The estimate effort using DolBat algorithm and other algorithms for Nasa-93 dataset.

Project No	Actual effort	Estimate by COCOMO II	Estimate by GA	Estimate by CSHS algorithm	Estimate by bat algorithm	Estimate by dolphin Algorithm	Estimate by Dolphin Bat algorithm
1	117.6	49.9115	58.1858	56.7804	80.3913	64.1238	69.5742
10	72	17.2299	20.3917	20.1144	26.1715	21.3653	23.278
20	60	66.2847	76.2562	73.7144	112.403	87.8512	94.9706
30	62	43.7072	53.093	53.3531	59.9984	50.9808	55.9453
40	114	46.7862	56.0976	55.8512	67.559	56.2675	61.5253
50	571.4	197.7477	216.1617	201.4788	408.987	295.510	314.9871
60	720	464.578	465.2299	407.3608	1350.59	852.914	887.419
70	432	210.3112	221.199	200.5821	505.272	44.0766	362.8745
80	703	197.4476	212.7259	196.2363	432.039	305.286	324.108
90	8211	1162.647	1192.2644	1061.7935	3082.0243	2018.6813	2114.1515

x_i represents i^{th} bat location in the space of the solution, v_i is the bat's velocity, t represents the present iteration, and x_{Best} represents global near optimal solution which has obtained to the point, over the entire population [29]. A new solution for the bat will be produced based on Eq. 23:

$$x_{new} = x_{old} + \varepsilon A^t \tag{23}$$

ε represents an arbitrary number which $\varepsilon \in [-1, 1]$, and A^t is the average loudness of all of the bats at the present time step. Following the update of the bats' positions, pulse emission rate rıand loudness At are updated, as well only in the case where the global near optimal solution is altered and the randomly produced number is $< A_i$. A_i and r_i updates are regulated with Eq. 24.

$$A_i^{t+1} = \alpha \cdot A_i^t \tag{24}$$

where α is constants and $\alpha = 0.9$ is used for simplicity, Check the condition of the termination for deciding whether going back to Eq. 20 or terminating the algorithm and producing the near optimal solution [28], and [30].

VI. DOLPHIN BAT ALGORITHM (DOLBAT) PROPOSED TO ESTIMATE SOFTWARE EFFORT

In this work has been using dolphin algorithm and Hybridized with bat algorithm to get better coefficients value to predict effort of software, in the initial phase in dolphin algorithm it configure three matrices randomly

a) Dol (location of dolphin)
b) K (optimal neighborhood)
c) L (optimal individual)

These matrices form of a two dimensional The row of matric represents the number of dataset projects used, and the columns of matric represent the coefficients of COCOMO II (a, b) necessary in the Eq. 1 to get the best estimation, Fitness is then calculated for each dolphin with the rest of the dolphins stored in k matrix, Then the search phase begins by adding the sound to Dol matrix by the Eq. 2 to generate L matrix based on fitness of Dol adding sound and find best fitness of L matrix and k matrix (fit_L, fit_k) by the Eqs. 3-6. which is near to the actual effort, in the calling phase compute the distance between Dol_i and Dol_j named $DD_{i,j}$, The transmission time matrix TS is updated By Eq. 7. Then, the reception phase starts where TS is reduced by (0.1) and update fit_k matrix By Eq.10, The last stage in the dolphin algorithm is the predation which starts after the value of R1 is calculated by Eq. 13 where R1 stands for the maximal domain in the search stage, speed is a constant which is Considered as the sound attribute after several attempts of different values found that speed $= 1$, t1 $= 3$ gave the best results, and compute the distance: between Dol_i and K_i named DK_i Eq. 11, the distance between L_i and K_i named DKL_i Eq. 12. The new position is then configured depend on the distance between (DK_i, DKl_i, R1) according to the Eqs. 14-18 and obtains a new location Eq. 15, or Eq. 17, computes the fitness and updates $Fit_{K,i}$. After new sites have been created using the dolphin algorithm, this new generation (which is arranged by lower fitness). The best location is the first location is sent in the bat algorithm, In the bat algorithm, the new generation is then created after updating the velocity and frequency for population by Eqs.20-21, Calculates the fitness for this generation and select the location that has the best fitness, repeat work of the bat algorithm 100 iterations to get the best location and return it to the Dolphin algorithm. The Dolphin algorithm repeats its steps up to 1000 iteration to get the best variables (a, b) required in the estimate software effort. Figure 1 represents the stages of the dolphin algorithm after Hybridized with bat algorithm.

VII. RESULTS AND DISCUSSION

The dolphin algorithm was applied and hybridized with the bat algorithm in Matlab language and on two sets of data:

a) Nasa-93 which contains information about 93projects.
b) Nasa-60 which contains information about 60 projects.

The results were compared with the COCOMO II model, bat algorithm [13], genetic algorithm [16] and CSHS algorithm [14], The results of the estimate software effort on Nasa-93 data are shown in Table 2. Where the value of the coefficients of COCOMO II (a, b) for the proposed algorithm and other algorithms shown in Table 3.

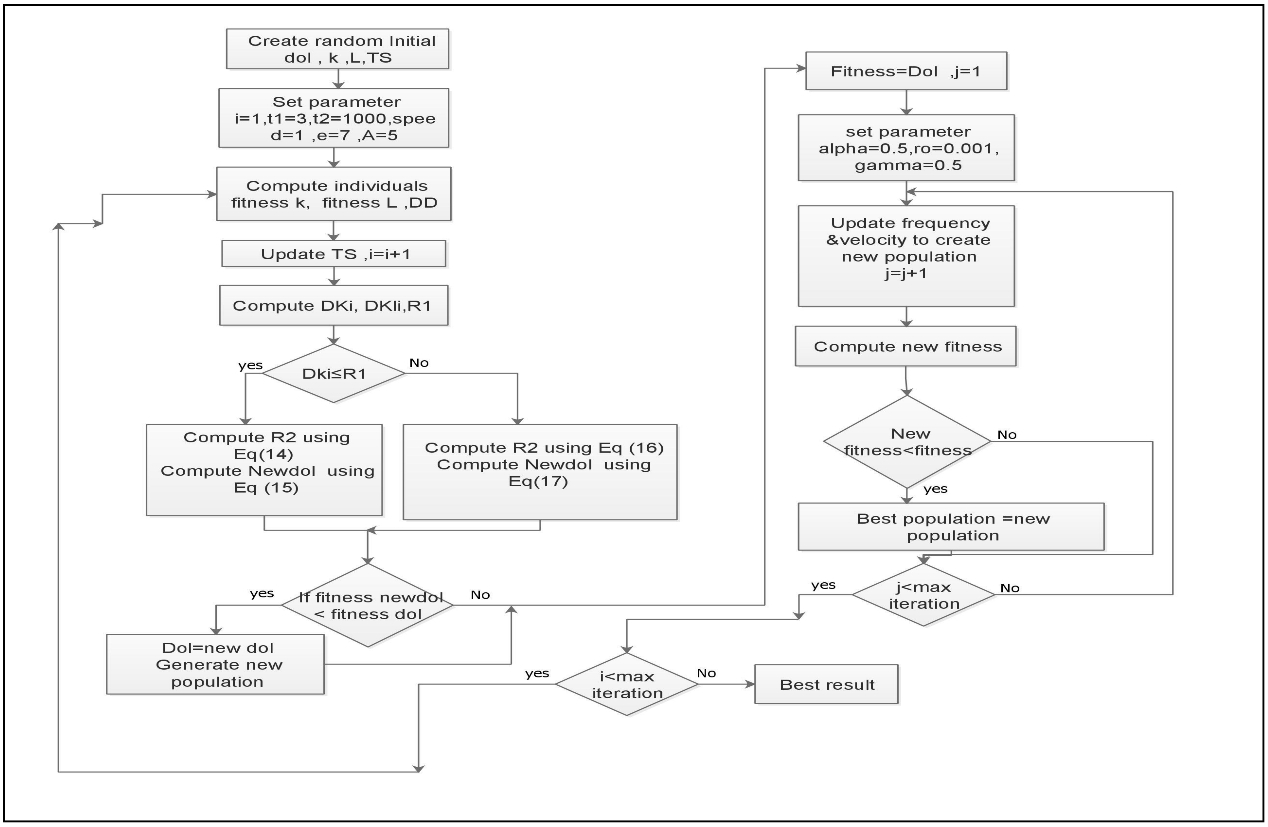

FIGURE 1. The fiowchart of DolBat Algorithm.

TABLE 3. COCOMO-II coefficient values of estimate effort using DolBat algorithm and other algorithm for Nasa-93 dataset.

Coefficient	Estimate by COCOMO II	Estimate by GA	Estimate by CSHS	Estimate by bat	Estimate by dolphin	Estimate by Dolphin Bat algorithm
A	2.94	4.1444	4.631	2.2637	2.4177	2.7643
B	0.91	0.85163	0.81	1.1368	1.0471	1.031

TABLE 4. MMRE values of estimate effort using DolBat algorithm and other algorithm for Nasa93 dataset.

Coefficient	COCOMO II model	GA model	CSHS Algorithm	Bat algorithm	Dolphin algorithm	Dolphin Bat algorithm
MMRE	57.40	53.0498	54.04	53.528	51.8755	50.2757
PROD	7.526	16.12	18.279	31.182	22.58	25.806

TABLE 5. The estimate effort using DolBat algorithm and other algorithm for Nasa-60 dataset.

Project No	Actual effort	Estimate by COCOMO II	Estimate by GA	Estimate by bat	Estimate by dolphin	Estimate by Dolphin Bat algorithm
1	8.4	5.293	7.1257	5.2376	6.0198	6.9184
10	42	22.4314	28.0064	31.6982	36.7152	37.9471
20	62	29.002	35.2301	46.6597	54.1974	53.893
30	117.6	47.6265	55.6897	91.7638	107.006	100.8421
40	300	117.885	130.0582	298.9942	350.749	304.5606
50	571.4	215.368	235.4266	570.5912	669.993	574.2581
60	3240	768.865	767.4658	3130.3904	3710.2295	2798.0869

The proposed algorithm was evaluated with the use of the Magnitude of Relative Error (MRE) [31]. To evaluate the performance in our work use two metrics: 1- Mean Magnitude of Relative Error (MMRE). In order to measure the

TABLE 6. COCOMO-II coefficient values of estimate effort using DolBat algorithm and other algorithm for Nasa-60 dataset.

Coefficient	Estimate by COCOMO II	Estimate by GA	Estimate by bat	Estimate by dolphin	Estimate by Dolphin Bat algorithm
A	2.94	4.1444	2.3403	2.6771	3.2827
B	0.91	0.85163	1.186	1.192	1.1098

TABLE 7. MMRE values of estimate effort using DolBat algorithm and other algorithms for Nasa-60 dataset.

Coefficient	COCOMO II Model	GA model	Bat algorithm	Dolphin Algorithm	Dolphin Bat algorithm
MMRE	33.9581	29.9469	16.98	16.65	14.576
PROD	6.66	11.66	61.66	61.66	66.66

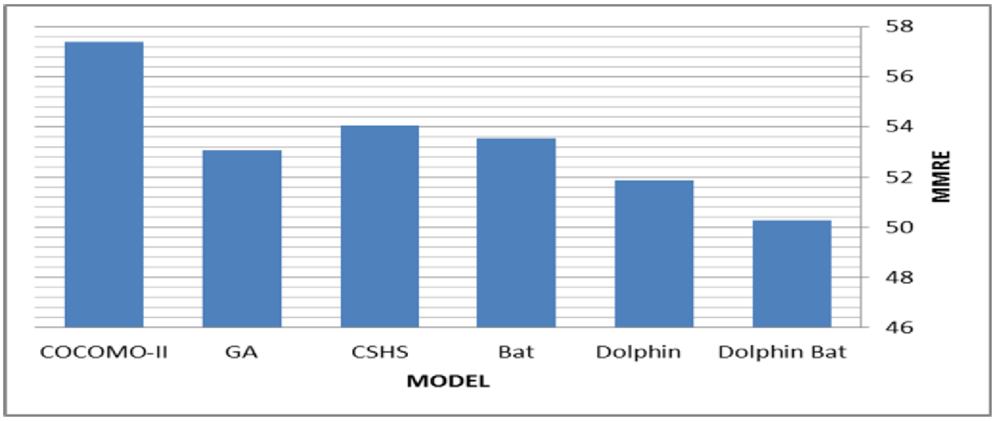

FIGURE 2. MMRE values of estimate effort using DolBat algorithms and other algorithms for Nasa 93 datase.

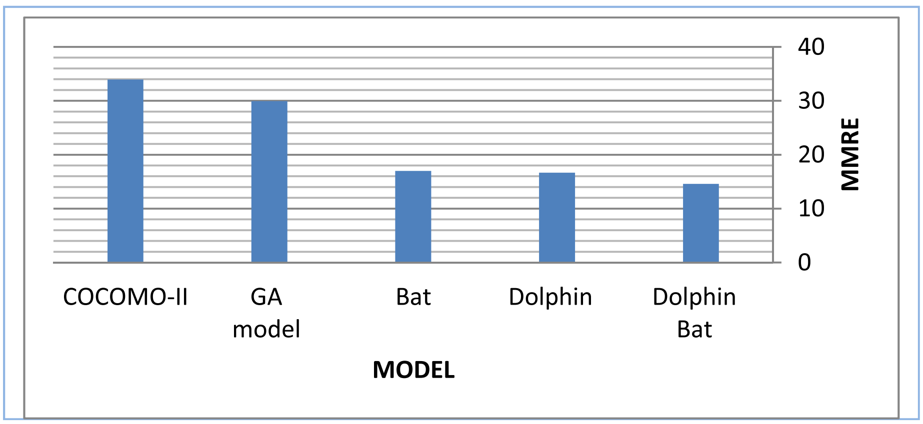

FIGURE 3. MMRE values of estimate effort using DolBat algorithms and other algorithms for Nasa-60 dataset.

appropriate accuracy of COCOMO II, we used the most used evaluation measures in the domain of software engineering. These are MRE Eq. 25

$$MRE = (|estimate\ effort(i) - actual\ effort(i)|) / (actual\ effort(i)) \quad (25)$$

And Mean Magnitude of Relative Error (MMRE) [31] using Eq. 26:

$$MMRE = 1/n \sum_{i=1}^{n} (|estimate\ effort\ (i) - actuale\ ffort(i)|)/actual\ effort(i)) \quad (26)$$

where N is a number of projects in the dataset. 2- PRED is a ratio of predicts found within X percent of the actual values according to the Magnitude of relative error (MRE). PRED is computed in the Eq. 27 [17]

$$\text{PRED(X)} = \frac{100^*}{N} \sum_{i=}^{N} \left\{ \begin{array}{l} 1,\ if\ MREi \le X \\ 0,\ otherwise \end{array} \right\} \tag{27}$$

Generally, X = 0.25 and N is a number of project in the dataset, From the results of the measurements shows that the proposed algorithm gave the best results to estimate the effort as showed in Table 4 and Fig. 2 The results of the estimate software effort on Nasa-60 data are shown in Table 5. Where the value of the coefficients of COCOMO II (a, b) for the proposed algorithm and other algorithms shown in Table 6. the results of the measurements show that the proposed algorithm gave the best results to estimate the effort as showed in Table 7 and Fig. 3

VIII. CONCLUSION AND FUTURE WORK

Software Cost Estimation (SCE) is an important phase in the software development life cycle (SDLC). The good estimation leads the project smoothly towards completion. In this work has applied the dolphin swarm algorithm and hybrid bat algorithm (DolBat) to optimize the cost estimation models COCOMO II coefficients (a, b). Where the algorithm of the dolphin swarm is especially suitable for the optimization tasks, with fewer individuals and more fitness function calls and benefits from the echolocation and adopts a variety of strategies for obtaining the solution more effectively. By applying the proposed algorithm the value of MMRE equal (50.2757) for NASA-93, and the value of MMRE equal (14.576) for NASA-60, as has compared with the previously has applied algorithm the proposed algorithm has the lowest MMRE. In future work, this work can be extended by the hybridization using a dolphin algorithm with other algorithms or using a new algorithm to estimate the efforts of software applications.

ACKNOWLEDGMENT

The authors would like to thank the University of Mosul/ College of Computer Sciences and Mathematics for their provided facilities, which helped to improve the quality of this work.

REFERENCES

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

[2] S. Aljahdali and A. F. Sheta, "Software effort estimation by tuning COOCMO model parameters using differential evolution," in *Proc. ACS/IEEE Int. Conf. Comput. Syst. Appl. (AICCSA)*, May 2010, pp. 1283–1289.

[3] B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approaches U Aă survey (no. 1U4)," JC Baltzer AG, Sci. Publishers, Tech. Rep., 2000, pp. 177–205.

[4] B. Boehm, *Software Engineering Economics*, vol. 197. New York, NY, USA, 1981.

[5] A. J. Albrecht, "Measuring application development productivity," in *Proc. Joint Share, Guide, IBM Appl. Develop. Symp.*, 1979.

[6] G. Karner, "Resource estimation for objectory projects," Objective Syst. SF AB, Tech. Rep., 1993, pp. 1–9, vol. 17.

[7] R. K. Clemmons, "Project estimation with use case points," *J. Defense Softw. Eng.*, vol. 19, no. 2, pp. 18–22, 2006.

[8] A. BaniMustafa, "Predicting software effort estimation using machine learning techniques," in *Proc. 8th Int. Conf. Comput. Sci. Inf. Technol. (CSIT)*, Jul. 2018, pp. 249–256.

[9] T.-Q. Wu, M. Yao, and J.-H. Yang, "Dolphin swarm algorithm," *Frontiers Inf. Technol. Electron. Eng.*, vol. 17, no. 8, pp. 717–729, Aug. 2016.

[10] J. S. Shirabad and T. J. Menzies. (2005). The PROMISE Repository of Software Engineering Databases. School of information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada. [Online]. Available: http://promise.site.uottawa.ca/SERepository

[11] 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, Feb. 2019.

[12] 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.

[13] 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.

[14] 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, Jan. 2017.

[15] D. Nandal and O. P. Sangwan, "Software cost estimation by optimizing COCOMO model using hybrid BATGSA algorithm," *Int. J. Intell. Eng. Syst.*, vol. 11, no. 4, pp. 250–263, 2018.

[16] M. Alajlan and N. Tagoug, "Optimization of COCOMO II model for effort and development time estimation using genetic algorithms," in *Proc. Int. Conf. Commun., Comput. Sci. Inf. Technol.*, Dibai, India, 2016, pp. 1–5.

[17] F. Nayebi, A. Abran, and J.-M. Desharnais, "Automated selection of a software effort estimation model based on accuracy and uncertainty," *Artif. Intell. Res.*, vol. 4, no. 2, pp. 1–10, 2015.

[18] R. Wang, P. Peng, L. Xu, X.-X. Huang, and X.-L. Qiao, "A novel algorithm for software development cost estimation based on fuzzy rough set," *J. Eng. Sci. Technol. Rev.*, vol. 9, no. 4, pp. 217–223, 2016.

[19] N. Akram, Al-Saati, and T. R. Alreffaee, "Using multi expression programming in software effort estimation," 2018, *arXiv:1805.00090*. [Online]. Available: http://arxiv.org/abs/1805.00090

[20] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic algorithms in modeling and optimization," in *Metaheuristic Applications in Structures and Infrastructures*. 2013, pp. 1–24.

[21] A. Idri, A. Abran, and L. Kjiri, "COCOMO cost model using fuzzy logic," in *Proc. 7th Int. Conf. Fuzzy Theory Techn.*, vol. 27, 2000, pp. 1–4.

[22] A. Kaushik, A. Chauhan, D. Mittal, and S. Gupta, "COCOMO estimates using neural networks," *Int. J. Intell. Syst. Appl.*, vol. 4, no. 9, pp. 22–28, Aug. 2012.

[23] W. Yong, W. Tao, Z. Cheng-Zhi, and H. Hua-Juan, "A new stochastic optimization approach—Dolphin swarm optimization algorithm," *Int. J. Comput. Intell. Appl.*, vol. 15, no. 2, Jun. 2016, Art. no. 1650011.

[24] K. Lenin, B. R. Reddy, and M. S. Kalavathi, "Dolphin echolocation algorithm for solving optimal reactive power dispatch problem," *Int. J. Comput.*, vol. 12, no. 1, pp. 1–15, 2014.

[25] W. Qiao and Z. Yang, "Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function optimization problems," *IEEE Access*, vol. 7, pp. 110472–110486, 2019.

[26] Y. Li and X. Wang, "Improved dolphin swarm optimization algorithm based on information entropy," *Bull. Polish Acad. Sci., Tech. Sci.*, vol. 67, no. 4, pp. 679–685, 2019.

[27] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*. Berlin, Germany: Springer, 2010, pp. 65–74.

[28] P. W. Tsai, J. S. Pan, B. Y. Liao, M. J. Tsai, and V. Istanda, "Bat algorithm inspired algorithm for solving numerical optimization problems," *Appl. Mech. Mater.*, vols. 148–149, pp. 134–137, Dec. 2011.

[29] Y. Wang, P. Wang, J. Zhang, Z. Cui, X. Cai, W. Zhang, and J. Chen, "A novel bat algorithm with multiple strategies coupling for numerical optimization," *Mathematics*, vol. 7, no. 2, p. 135, 2019.

[30] S.-S. Guo, J.-S. Wang, and X.-X. Ma, "Improved bat algorithm based on multipopulation strategy of island model for solving global function optimization problem," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–12, Aug. 2019.

[31] B. KumarSingh and A. K. Misra, "Software effort estimation by genetic algorithm tuned parameters of modified constructive cost model for NASA software projects," *Int. J. Comput. Appl.*, vol. 59, no. 9, pp. 22–26, 2012.

ANFAL A. FADHIL was born in Mosul, Nineveh, Iraq, in 1983. She received the B.S. and M.S. degrees in software engineering from Mosul University, in 2006 and 2012, respectively, From 2012 to 2019, she was taught software engineering and techniques in the College of Computer Sciences and Mathematics, University of Mosul. She has research in a field software engineering.

RASHA G. H. ALSARRAJ was born in Mosul, Nineveh, Iraq, in 1984. She received the B.S. and M.S. degrees in software engineering from Mosul University, in 2006 and 2012, respectively. From 2012 to 2019, she was taught software engineering and techniques in the College of Computer Sciences and Mathematics, University of Mosul. She has research in a field software engineering.

ATICA M. ALTAIE was born in Mosul, Nineveh, Iraq, in 1984. She received the B.S. and M.S. degrees in software engineering from Mosul University, in 2006 and 2012, respectively, From 2012 to 2019, she was taught software engineering and techniques in the College of Computer Sciences and Mathematics, University of Mosul. She has research in a field software engineering.