IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 17, 2020, accepted February 6, 2020, date of publication February 10, 2020, date of current version February 18, 2020.
Digital Object Identifier 10.1109/ACCESS.2020.2972826

Parameter Estimation of Software Reliability Model and Prediction Based on Hybrid Wolf Pack Algorithm and Particle Swarm Optimization

LI ZHEN[1,2], YANG LIU[1,2], WANG DONGSHENG[2,3], AND ZHENG WEI[1,2]
[1]School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China
[2]Reliability and System Engineering Open Group, Jiangsu University of Science and Technology, Zhenjiang 212003, China
[3]School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang 212003, China

Corresponding author: Li Zhen (justlz@just.edu.cn)

This work was supported in part by the Project Collaborated With the National Natural Science Foundation of China under Grant 61702234, and in part by the Jiangsu University of Science and Technology, Reliability and System Engineering Open Group (JRSOG) Open Fund under Grant 2019001.

ABSTRACT Software reliability is estimated and predicted based on software reliability model and software failure data. As a new optimization method, swarm intelligence algorithm has been widely used in solving the parameter optimization of the model. WPA (Wolf Pack Algorithm) and PSO (Particle Swarm Optimization) are two typical swarm intelligence algorithms. WPA has a strong global optimization ability, fast convergence speed and various optimization strategies, but the algorithm is relatively complex. PSO algorithm has a simple structure and fast convergence speed, but it is easy to fall into premature, which leads to low accuracy of solution. Considering the advantages and disadvantages of the two algorithms, a hybrid method of WPA and PSO is proposed, and a fitness function is constructed on maximum likelihood estimation, then the parameters of software reliability model are estimated and predicted based on the hybrid algorithm (WPA-PSO). Five sets of data from industry are used to estimate the parameters of GO model and make predictions. The simulation results show that the hybrid algorithm has higher accuracy of parameter estimation, better optimization performance, better accuracy of prediction and algorithm stability than single algorithm, and show obvious advantages than the single algorithm in the case of limited data.

INDEX TERMS Software reliability, parameter estimation, swarm intelligence, wolf pack algorithm, particle swarm optimization.

I. INTRODUCTION

Software reliability is a qualitative indicator for measuring software quality and has important research significance, so it is getting more and more attention from researchers. So far, researchers have put forward nearly a hundred software reliability models, such as GO model [1], MO model [2] and JM model [3], and so on. However, these models are nonlinear function models, and it is difficult to directly estimate their parameters. Therefore, a new idea is to apply the intelligent optimization algorithm to the model parameter estimation.

As a swarm intelligence optimization algorithm, WPA (Wolf Pack Algorithm) is proposed by Wu *et al.* [4]. The algorithm has a good global convergence and high precision of solution. The WPA is a typical swarm intelligence algorithm, which has similar group behaviors with the wolves' group that has a strict mechanism for predation and distribution of prey. The core idea is to search for the optimal solution through the collaboration and information sharing between the wolves, and has now been applied to many fields.

To solve the high-dimension nonlinear optimization model, Zhuang and Jiang adapted the Wolf Pack Algorithm (WPA) to achieve the network loss minimization [5]. To solve the limitations of existing evolutionary algorithms in parameter identification, Li and Wu proposed a novel and efficient oppositional Wolf Pack Algorithm (OWPA), which has a good balance of exploitation and exploration, to estimate the parameters of Lorenz chaotic system [6]. In the literature [7], YongBo *et al.* applied the modified wolf pack search (WPS) algorithm to compute the quasi-optimal trajectories for the rotor wing UAVs in the complex three-dimensional (3D) spaces including the real and fake 3D

The associate editor coordinating the review of this manuscript and approving it for publication was Jagdish Chand Bansal.

spaces. In the literature [8], Xu *et al.* proposed an improved Wolf Pack Algorithm to solve the optimization problem of logistics distribution center location. Chen *et al.* proposed a modified two-part wolf pack search (MTWPS) algorithm updated by the two-part individual encoding approach as well as the transposition and extension (TE) operation for the multiple travelling salesmen problem (MTSP) [9]. Menassel *et al.* provided more detailed study about the Wolf Pack Algorithm for the fractal image compression in the literature [10]. In order to solve the scheduling problems of Re-entrant Hybrid Flowshop (RHFS), Han *et al.* investigated the mathematical programming model of RHFS, and proposed the Wolf Pack Algorithm (WPA) as a global optimization method [11]. In view of the local extreme problem of the gradient descent algorithm which makes the working face of mine gas emission prediction uncertainly, Xu *et al.* combined Wolf Pack Algorithm (WPA) with complex neural network nonlinear prediction method to the established new prediction model [12]. In the literature [13], Gao *et al.* proposed a Quantum-Inspired Wolf Pack Algorithm (QWPA) based on quantum encoding to enhance the performance of the Wolf Pack Algorithm (WPA) to solve the 0-1 knapsack problems. Moreover, in the literature [14], there was a discussion of the wolf group algorithm and its application in many aspects. For example, the convergence of WPA was analyzed, and how to apply the WPA to solve path optimization, reservoir scheduling optimization and microgrid scheduling optimization were discussed.

PSO (Particle Swarm Optimization) was proposed by Eberhart and Kennedy in 1995, which refers to the foraging behavior of birds. The advantage of the PSO is that it has fewer parameters in its model, and is easier to implement. In the early stage, PSO has a faster convergence speed. However, it is easy to fall into local optimum during the search process, resulting in low accuracy of the solution.

In order to study the pyrolysis of a typical agricultural residue, Xu *et al.* used Particle Swarm Optimization (PSO) to estimate the parameters of the pyrolysis kinetic model [15]. And then, Ding *et al.* used Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to estimate the reaction kinetic parameters of biomass pyrolysis. The results showed that the accuracy and efficiency of PSO are higher [16]. Laxmi *et al.* proposed hybridization of Particle Swarm Optimization with simulated annealing for planning and scheduling issues [17]. In the literature [18], Espitia and Sofrony presented the statistical analysis of vortex particle swarm optimization (VPSO) which is a boost algorithm based on self-propelled particle swarms. In the literature [19], Aydilek proposed a hybrid algorithm combining firefly and particle swarm optimization (HFPSO). The proposed algorithm was able to exploit the strongpoints of both particle swarm and firefly algorithm mechanisms.

This paper combines the characteristics of the two algorithms to estimate the parameters of the software reliability model and make prediction with it. First, the section II gives the basic theory related to this paper: software reliability, software reliability model, and the basic principles of the WPA. Then, section III describes the research method: the construction of the fitness function and the specific implementation process of the hybrid algorithm (WPA-PSO). Subsequently, section IV performs experimental simulations and compares the estimated results of different algorithms. Finally, the section V summarizes the research work of this paper.

II. BASIC CONCEPTS

A. SOFTWARE RELIABILITY AND MODEL

Software reliability is the probability that the software will not cause a system failure under the specified conditions and time. The IEEE Computer Society defines the software reliability as follows [20]:

1. in the specified conditions, within a specified time, the probability that the software does not cause a system failure;

2. in the specified time period, the ability of the program to perform the required functions.

Modeling software reliability is a mathematical method to evaluate software reliability, and the choice of model parameters will directly affect the accuracy of software reliability prediction. In this paper, a representative GO model in the software reliability model is selected as the research object, and its parameters are estimated. The estimated function of the cumulative failure number in the software system is as follows:

$$m(t) = a(1 - e^{-bt}) \tag{1}$$

where: m(t) represents the expected function of the cumulative number of failures until time t; a represents the total number of failures the software expects to be detected after the end of the test; b represents the probability that the remaining failures are found, and is a proportional constant with a range of (0, 1).

B. WOLF PACK ALGORITHM

The Wolf Pack Algorithm simulates the hunting behavior of the wolves to deal with the function optimization problem, and divides the wolves into three categories: the head Wolf, the search Wolf and the fierce Wolf, as shown in Figure 1:

The whole hunting activity of the wolves is abstracted into three kinds of intelligent behaviors (walking behavior, summoning behavior, and siege behavior), the ''winner is king'' head Wolf generation rule and ''strong survival'' wolf group renewal mechanism.

(1) Head wolf generation criterion: Starting from an initial prey pack in the space to be searched, the wolf with the best fitness value is selected as the head Wolf.

(2) Walking behavior: Except for the head Wolf, a total of *S-num* best artificial wolves were selected to perform the walking behavior as the search Wolf. *S-num* randomly takes the integer between $[(\alpha + 1)\ , n/\alpha]$. n is the total number of artificial wolves. α is the proportion factor of search Wolf. First, the concentration of prey odor (Y_i) at the current position of the search Wolf (i) is calculated. If $Y_i > Y_{lead}$, then

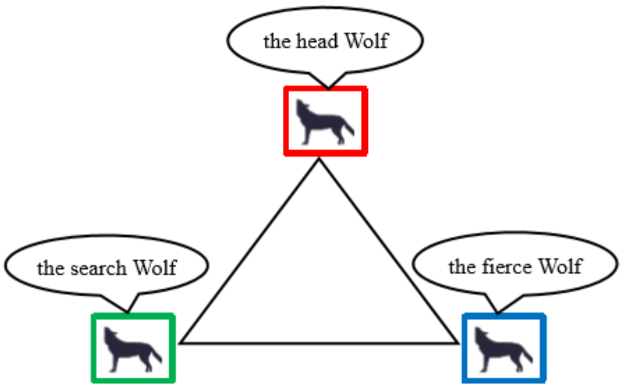

FIGURE 1. Three types of wolves.

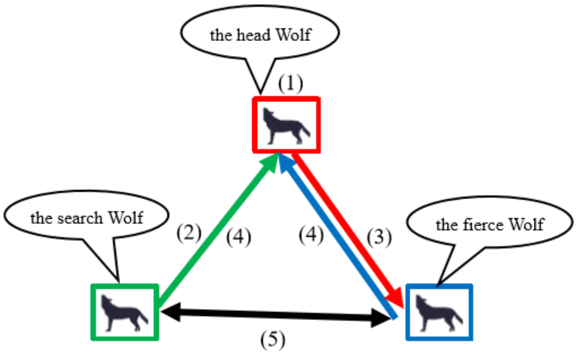

FIGURE 2. Diagram of wolf group algorithm.

$Y_{lead} = Y_i$. That is, the search Wolf takes the place of the head Wolf and initiates summoning behavior. If $Y_i < Y_{lead}$, then the search Wolf moves forward in h directions separately (the step size at this time is called *stepa*). After going in the p-direction($p = 1, 2, 3, \ldots, h$), the position of the search Wolf in the d-dimensional space is:

$$x_{id}^{p} = x_{id} + \sin(2\pi \times p/h) \times step_p^d \tag{2}$$

The search Wolf (i) walks away until odor concentration perceived by one of the search wolves is $Y_i > Y_{lead}$, or the number of walks (T) is T_{max}.

There is a difference in the prey search method for each search Wolf, that is, the value of h is different, and the random integer between $[h_{min}, h_{max}]$ is taken in the actual situation.

(3) summoning behavior: when the head Wolf starts to howl for summoning behavior, notify the surrounding *M-num* fierce Wolf to quickly draw close to the head Wolf, where *M-num* = n-*S-num*-1; When a fierce Wolf hears a howl, it rapidly approaches the head Wolf at a relatively long gallop step (called gallop step $step_b$). Then, when the fierce Wolf (j) experiences the number of $k+1$ iterations, its position in the d-dimensional space is:

$$x_{jd}^{k+1} = x_{jd}^{k} + step_b^d \cdot \frac{g_d^k - x_{jd}^k}{|g_d^k - x_{jd}^k|} \tag{3}$$

In the formula, g_d^k denotes the position of the head wolf of the k-generation population in the d-dimensional space.

During the running process, if the odor concentration perceived by fierce Wolf (j) was $Y_j > Y_{lead}$, then $Y_j = Y_{lead}$ and the fierce Wolf transforms into the head Wolf and initiates summoning behavior. If $Y_j < Y_{lead}$, then the fierce Wolf (j) continues to run, and when the distance (d_{js}) between the first Wolf (s) and the fierce Wolf (j) is less than judging distance (d_{near}), it turns to siege. The judging distance (d_{near}) is estimated by formula (4).

$$d_{near} = \frac{1}{D \cdot \omega} \cdot \sum_{d=1}^{D} |max_d - min_d| \tag{4}$$

In the formula, D is the dimension of the variable space to be optimized; max_d and min_d are the maximum and minimum values of the d-dimensional space to be optimized. w is the distance decision factor, and its different values will affect the convergence speed of the algorithm. When w increases, the convergence speed of the algorithm will be accelerated, but if w is too large, it will make it difficult for the artificial Wolf to enter the siege behavior and lack of fine search for prey.

(4) siege behavior: the wolves conduct siege behavior according to formula (6). For the k-generation of wolves, if the position of a prey in the i-dimension is $G_d^k \mathrm{G}_\mathrm{d}^\mathrm{k}$, then siege behavior of wolves expressed by the following formula.

$$x_{id}^{k+1} + \lambda \cdot step_c^d \cdot \left| G_d^k - x_{id}^k \right| \tag{5}$$

where, λ is the random number between [-1,1]; $step_c^d$ is the attack step taken by artificial Wolf (i) to conduct siege behavior in the first d-dimension.

The steps involved in three kinds of intelligent behavior are walk step ($step_a^d$), gallop step ($step_b^d$), attack step ($step_c^d$), and they have the following relationship in the d-dimensional space.

$$step_a^d = \frac{step_b^d}{2} = 2 \cdot step_c^d = |max_d - main_d| / S \tag{6}$$

where, S is the step factor.

(5) the Wolf renewal mechanism of "the strong survival": The artificial wolves (R) with the worst value of the objective function are removed, and new artificial wolves (R) are randomly generated at the same time. The value of R is a random integer between [n/((2×β)), n/β], and β is the population renewal scaling factor.

III. METHOD DESCRIPTION

A. CONSTRUCTION OF FITNESS FUNCTION

In this paper, the GO model is used as an example to illustrate the hybrid algorithm of WPA and PSO. And the maximum likelihood method is used to get the solution of a and b, that are parameters of the GO model. The calculation formulas of a and b are shown as follows:

$$\begin{cases} a = \dfrac{n}{1 - e^{-bt_n}} \\ \dfrac{n}{b} = at_n e^{-bt_n} \end{cases} \tag{7}$$

In the above formula, n represents the known failure number; t_i is the time of failure (i); $i = 1, 2, 3, \ldots n$.

In this paper, a new fitness function is constructed according to the maximum likelihood estimation formula of the parameters a and b of the GO model. The specific method is to substitute the first term in formula (7) into the second term and carry out mathematical transformation to construct a formula only related to parameter b, as shown below:

$$f = \left| b - \frac{n(1-e^{-bt_n})}{nt_n e^{-bt_n} + (1-e^{-bt_n})\sum_{i=1}^{n} t_i} \right| \tag{8}$$

f is the new fitness function, and all the parameters in the formula are known except b. The smaller f is, the better the effect of parameter b estimation is. Through the hybrid algorithms (WPA-PSO) iterative search, the optimal parameter b is get when the algorithm stops criterion is reached, and then the corresponding optimal parameter a is obtained by substituting the maximum likelihood estimation formula of parameter a.

B. IMPLEMENTATION OF WPA-PSO

Based on the advantages and disadvantages of WPA and PSO, the hybrid algorithm (WPA-PSO) can achieve a good complementary effect. The approach taken in the solution space is: after the particles of the PSO search for particles, the search process of the wolves in the WPA is used to search through the rules to determine the final new position. The algorithm flow is shown in Figure 3:

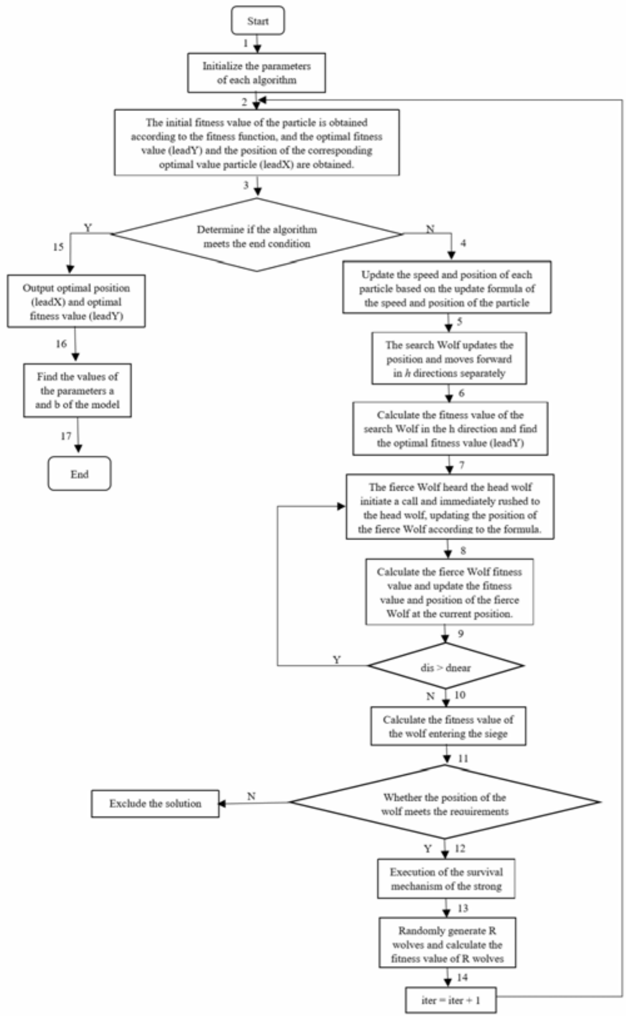

FIGURE 3. Process of the WPA-PSO.

Input: WPA parameters and PSO parameters, actual software failure number n and each failure occurrence time t_i;

Output: Estimation result of parameter b; parameter a is obtained by parameter b according to formula (7).

Each specific operation in Figure 3 is described as follows:

(1) Initialize all parameters. Total number of artificial wolves: *wolf-num* = 50 (Number of particles); the search Wolf scaling factor: $\alpha = 4$; the distance decision factor: $w = 100$; the step factor $S = 1000$; the update scaling factor: $\beta = 10$; the maximum number of migration limit: *T-max* = 30; Maximum number of iterations *G-max* = 500; The value of the *dth* variable to be searched is [*lb*,*ub*], where $ub = \pi$, $lb = -\pi$; Walk step: *stepa* = (*ub* - *lb*)/*S*; gallop step: *stepb* = (*ub* - *lb*)/*S*∗2; Attack step size: *stepc* = (*ub* - *lb*) / (*S*∗2); Randomly generate the walk direction h, which is a random number of [*h-min*, *h-max*], where *h-min*= 2, *h-max* = 15; the accuracy of the adaptive value requires $k \leq$1e(-5); The position of each wolf is also the parameter x of the GO model, initialized to a random number between (0, 1) (where x is the value of b); Inertia weight $W = 0.9$; learning factor $C1 = C2 = 1.5$, speed v is a random number in [-1,1].

(2) Substitute b into the fitness function, find the current initial fitness value of each particle. Then assign the optimal fitness value to *leadY*, and the value of the location of the optimal fitness value is taken as *leadX*.

(3) Determine whether the algorithm stop condition is satisfied: *iter* = *G-max* or *leadY* has an accuracy of k. If the precision reaches k, go to step (15), otherwise go to step (4);

(4) Update the speed and position of each particle according to the update formula of the particle swarm algorithm.

(5) The search Wolf swims in the h direction, performs position update according to the walk formula (2), and finds the adaptation value (*bestnexty*) at the updated position;

(6) Find the optimal fitness value (*leadY*) and the optimal position (*leadX*) of the updated head wolf.

(7) The head Wolf initiated the summoning behavior, and the fierce Wolf approaches the head Wolf according to the gallop formula (3).

(8) Calculate the fitness value of the updated fierce Wolf (*Mnexty*), update the value of the fierce Wolf at the current position ($Y(j)$ and $X(j)$).

```
start
Initialize parameters and the position of the wolf group
for m=1: wolf-num
    Calculate the initial fitness value of each particle;
end for
Find the optimal fitness value and the position (x) and give them to the head
Wolf;
Sort initial fitness values;
iter=0;
while (iter< G-max &&leadY>=0.00001)
    Update the speed and position of each particle;
    Set the parameters of the search Wolf;
    for t = 1: T-max
        for j = 2: S-num + 1
            Randomly generate the direction of the walk;
            for k = 1: h
                Starting from the second wolf, each search Wolf walks
                within the maximum number of walks;
                When the location of the search Wolf meets the
                requirements, calculate the corresponding fitness value;
            end for
            Find the optimal fitness value (bestnexty) and the
            corresponding position in the process of walking;
            if bestnexty < Y(j)
                Replace the current individual optimal value in the
                PSO;
            end if
            if bestnexty < leadY
                Replace the global optimal value in PSO;
                Exit loop;
            end if
        end for
    end for
    for i = S-num + 1 + 1: wolf-num
        while 1
            Summoning all the time, update the position of the wolf;
            Calculate the fitness value of the fierce Wolf after the
            updated position;
            Directly update the position and the fitness value of the
            fierce Wolf at the current position;
            Calculate the dis and dnear;
            if Mnexty < leadY
            If the fitness value of the updated fierce Wolf (Mnexty)
                is better than leadY, it replaces leadY;
            elseif dis < dnear
                conduct siege behavior;
                Calculate the fitness value of the wolf performing the
                siege (Gnexty);
                if Gnexty < Y(i)
                    If the Gnexty is better than Y(i), it replaces Y(i);
                end if
            end elseif
            break;
        end while
    end for
    if (X (i, :))<0.0001
        The wolf position meets the requirements and continues to search
        for excellence;
    end if
    Perform the survival of the strong, and discard the worst R wolves;
    Re-randomly generate R wolves;
    for p=wolf-num: -1: wolf-num-R+1
        Calculate the fitness value of the R wolves;
    end for
    iter=iter+1;
    When the algorithm end condition is satisfied, the optimal fitness
    value leadY and the optimal position leadX are output.;
end while
Know the value of b and solve the value of a according to the formula.
end
```

FIGURE 4. Implementation of the WPA-PSO.

TABLE 1. Failures of dataset.

Data Sets	Failure number
SYS1	136
S3	278
SR1	397
CSR2	129
CSR3	104

(9) Determine whether *dis* is greater than *dnear*, if it is greater, then perform step (7), if it is less, then perform step 10.

(10) For the wolf entering the siege behavior, the position is updated according to formula (5), the fitness value *Gnexty* is obtained at the same time, and the value of the current position is updated.

(11) After the *i-th* wolf is updated, it is judged whether the accuracy of the position *Xi* satisfies the requirement. If yes, discard the location of the particle update, using the original location instead; otherwise, retain the value at the updated location; and perform step (12);

(12) According to the update mechanism of the wolves, perform the survival of the strong, and discard the worst R wolves.

(13) Randomly generate R wolves and calculate the fitness value of R wolves

(14) iter = iter + 1, go to step (2).

(15) When the algorithm end condition is satisfied, the optimal fitness value *leadY* and the optimal position *leadX* are output.

(16) Know the value of *b* and solve the value of *a* according to the formula.

According to the process above, the hybrid algorithm is described as follows in Fig 4:

As shown in Figure 4, the time complexity of WPA-PSO is O(*G-max***T-max***S-num***h*), where *G-max* represents the maximum number of iterations, *T-max* represents the maximum number of migration limits, *S-num* represents the number of artificial wolves, and *h* represents the direction of the walk. Therefore, according to the calculation principle of algorithm time complexity, the time complexity of WPA-PSO is $O(n^4)$, and the actual time used in the experiment is analyzed and solved in the last part of this paper.

IV. SIMULATION RESULTS AND COMPARISON

In this paper, five sets of software failure data (SYS1, SS3, CSR1, CSR2 and CSR3) obtained in an actual industrial project are used. The address of data downloaded is http://www.cse.cuhk.edu.hk/lyu/book/reliability/data.html. The specific failure number and testing time is in the Table 1 and Table 2.

TABLE 2. Testing time.

Data Sets	Testing Time(/s)
SYS1	886282
SS3	56065.684
CSR1	108890
CSR2	88968
CSR3	15360.5

Especially, in this five sets of failure data, the failure number and testing time of CSR3 are obviously less than the other four groups. This limited failure number and testing time of CSR3 will put higher requirements on generality and accuracy of algorithm. And the solution of CSR3 is naturally a well distinguishment on performance of different algorithms.

A. PARAMETER ESTIMATION ON ALL DATA

The parameter settings of the hybrid algorithm (WPA-PSO) are shown in section III. The parameters of single WPA and PSO are set as follows:

- WPA:

Total number of artificial wolves: *wolf-num* = 60(Number of particles);
the search Wolf scaling factor: $\alpha = 4$;
the distance decision factor: $w = 100$;
the step factor $S = 1000$;
the update scaling factor: $\beta = 10$;
the maximum number of migration limit: *T-max*= 30;
Maximum number of iterations *G-max* = 500;
The value of the *dth* variable to be searched is [*lb*, *ub*], where $ub = \pi$, $lb = -\pi$;
Walk step: $stepa = (ub - lb)/S$;
gallop step: $stepb = (ub - lb)/S^*2$;
Attack step size: $stepc = (ub - lb)/(S^*2)$;
Randomly generate the walk direction h, which is a random number of [*h-min*, *h-max*], where $h - min = 2$, $h - max = 15$;
accuracy of the adaptive value requires $k \leq$1e(-5);
The position of each wolf is also the parameter x of the GO model, initialized to a random number between (0, 1) (where x is the value of b).

- PSO:

The number of particles: $n = 60$;
the inertia weight: $W = 0.9$;
the maximum number of iterations: $G - max = 500$;
the learning factor $C1 = C2 = 1.5$;
the accuracy of the fitness value requires $k \leq$1e(-5);
b is a random number in (0,1);
the velocity v is a random number in [-1,1].
Each algorithm iterates 500 times each time, runs 20 times, and selects the best results.

TABLE 3. Estimation results of WPA.

Data Sets (Failure number)	Parameter Estimation Results of GO Model		Fitness value
	a	*b*	*J*
SYS1(136)	138.3343	4.6029e-5	7.9656e-5
SS3(278)	281.5277	7.8115e-5	3.6408e-5
CSR1(397)	398.7180	5.0024e-5	6.6645e-5
CSR2(129)	129.9803	6.6098e-5	2.2681e-5
CSR3(104)	108.8486	2.0255e-4	3.2028e-5

TABLE 4. Estimation results of PSO.

Data Sets (Failure number)	Parameter Estimation Results of GO Model		Fitness value
	a	*b*	*J*
SYS1(136)	139.6329	6.3917e-5	4.6293e-5
SS3(278)	282.2291	6.4294e-5	7.1639e-5
CSR1(397)	400.3293	6.8260e-5	4.7001e-5
CSR2(129)	142.5681	2.6318e-5	2.5276e-5
CSR3(104)	120.5481	2.3953e-4	9.1039e-5

TABLE 5. Estimation results of WPA-PSO.

Data Sets (Failure number)	Parameter Estimation Results of GO Model		Fitness value
	a	*b*	*J*
SYS1(136)	137.1257	8.6448e-5	4.3027e-5
SS3(278)	280.6701	8.3029e-5	4.1018e-5
CSR1(397)	397.7309	7.7850e-5	2.3975e-5
CSR2(129)	129.2648	8.9583e-5	4.6040e-5
CSR3(104)	104.0253	7.3542e-4	6.0892e-5

TABLE 6. The error rate of three algorithms.

Data Sets (Failure number)	The error rate		
	WPA-PSO	WPA	PSO
SYS1(136)	0.83%	1.72%	2.53%
SS3(278)	0.96%	1.27%	1.63%
CSR1(397)	0.18%	0.43%	1.09%
CSR2(129)	0.21%	0.76%	1.70%
CSR3(104)	0.02%	4.66%	15.91%

It is known that the actual cumulative failure numbers of the five data sets SYS1, SS3, CSR1, CSR2, and CSR3 are 136, 278, 397, 129 and 104 respectively. The comparison of the experimental results is shown in Table 3, Table 4 and Table 5:

Based on the fitness function proposed in the paper, the error rate of the actual results and the estimation results of the WPA, PSO and WPA-PSO are shown in Table 6 and Figure 3.

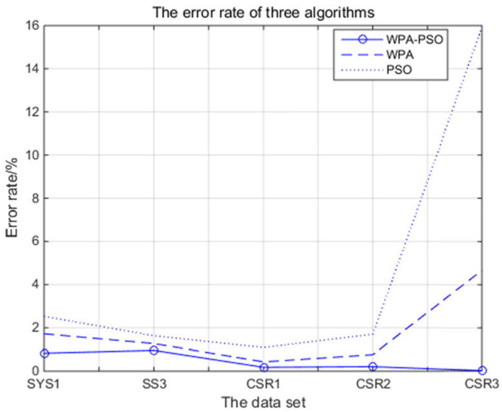

FIGURE 5. The error rate of three algorithms.

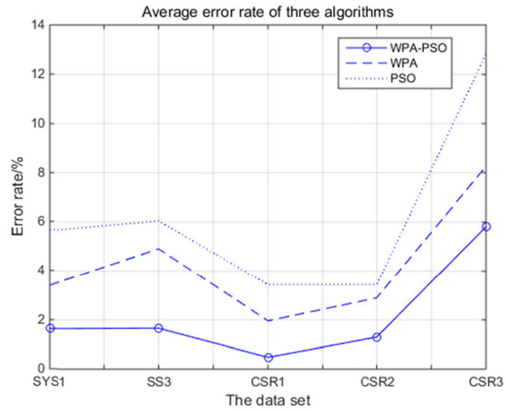

FIGURE 6. Average error rate of three algorithms.

TABLE 7. The average value of three algorithms.

Data Sets (Failure number)	Average of a		
	WPA-PSO	WPA	PSO
SYS1(136)	138.2256	140.6394	143.6831
SS3(278)	282.6218	291.6492	294.8317
CSR1(397)	399.6286	404.7934	410.6683
CSR2(129)	130.8211	132.7472	135.8544
CSR3(104)	110.0372	112.5853	117.3329

TABLE 8. Average error rate of three algorithms.

Data Sets (Failure number)	The error rate		
	WPA-PSO	WPA	PSO
SYS1(136)	1.64%	3.41%	5.65%
SS3(278)	1.66%	4.91%	6.05%
CSR1(397)	0.48%	1.96%	3.44%
CSR2(129)	1.30%	2.90%	3.44%
CSR3(104)	5.81%	8.26%	12.83%

It can be seen from Figure 5 that the error rate of WPA-PSO is the smallest in the five data sets when the three algorithms run 20 times. Its error rate in the finite data set of CSR3 is still the most optimal, and there is no sudden increase in the error rate as PSO and WPA in the CSR3 data set. It shows that the WPA-PSO has obvious algorithm accuracy and stability in the case of limited data than a single algorithm.

The results of the three algorithms running 20 times are counted below. The average value is shown in Table 7. The error rate of the average value is shown in Table 8 and Figure 6.

It can be seen from Tables 6∼8 that the accuracy of a estimated by the WPA-PSO proposed in this paper is higher than that of a single algorithm. Moreover, it can be seen from Table 7 and Table 8 that it is more stable within 20 times, which is a powerful illustration of the effectiveness of the proposed hybrid algorithm (WPA-PSO).

TABLE 9. Estimation results of WPA (half data).

Data Sets (Failure number)	Parameter Estimation Results of GO Model		Fitness value
	a	b	J
SYS1(136)	138.0087	4.5802e-5	1.7106e-5
SS3(278)	280.9488	6.1276e-5	1.7203e-5
CSR1(397)	398.5609	5.6377e-5	1.1623e-6
CSR2(129)	129.4219	5.4361e-5	2.1392e-5
CSR3(104)	110.0301	1.7701e-4	1.3173e-5

TABLE 10. Estimation results of PSO (half data).

Data Sets (Failure number)	Parameter Estimation Results of GO Model		Fitness value
	a	b	J
SYS1(136)	142.2521	4.7219e-5	2.0452e-6
SS3(278)	290.7036	5.0926e-5	5.0975e-5
CSR1(397)	403.6743	6.3351e-5	3.4472e-5
CSR2(129)	141.5382	3.0418e-5	2.1073e-5
CSR3(104)	122.4285	1.3616e-4	8.9427e-5

B. ESTIMATION AND PREDICTION ON HALF DATA

In this section, this paper combines the parameter estimation with the model prediction to examine the performance of the three algorithms in the case of limited data. For the three algorithms, the first half of the five sets data sets are used to estimate the parameters of the GO model, and then the estimated parameters are substituted into the function expression of the GO model to predict the occurrence time of the latter half of the failures.

1) ESTIMATION

Each algorithm iterates 500 times each time, runs 20 times, and selects the best results. The results of parameter estimation are shown in Table 9, Table 10 and Table 11:

Based on the fitness function proposed in the paper, the error rates of WPA, PSO algorithm and WPA-PSO

TABLE 11. Estimation results of WPA-PSO (half data).

Data Sets (Failure number)	Parameter Estimation Results of GO Model		Fitness value
	a	b	J
SYS1(136)	136.0130	4.5582e-5	2.0023e-6
SS3(278)	279.0661	5.4186e-5	2.8172e-5
CSR1(397)	397.6108	4.1947e-5	3.9948e-5
CSR2(129)	129.1912	4.7894e-5	9.0421e-5
CSR3(104)	105.2653	1.7391e-4	1.0744e-5

TABLE 12. The error rate of three algorithms (half data).

Data Sets (Failure number)	The error rate		
	WPA-PSO	WPA	PSO
SYS1(136)	0.01%	1.48%	4.82%
SS3(278)	0.38%	1.06%	4.57%
CSR1(397)	0.15%	0.39%	1.68%
CSR2(129)	0.15%	0.33%	9.72%
CSR3(104)	1.22%	5.80%	17.72%

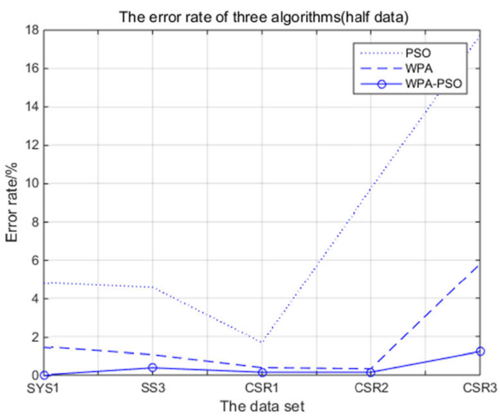

FIGURE 7. The error rate of three algorithms (half data).

TABLE 13. The average value of three algorithms (half data).

Data Sets (Failure number)	Average of a		
	WPA-PSO	WPA	PSO
SYS1(136)	138.6639	140.5581	149.7742
SS3(278)	285.5592	286.7524	291.6825
CSR1(397)	402.5931	403.9952	409.4228
CSR2(129)	131.0428	132.0021	138.3802
CSR3(104)	110.8847	113.7754	116.6624

TABLE 14. Average error rate of three algorithms (half data).

Data Sets (Failure number)	The error rate		
	WPA-PSO	WPA	PSO
SYS1(136)	1.96%	3.35%	10.59%
SS3(278)	2.72%	3.15%	6.18%
CSR1(397)	1.41%	1.76%	3.29%
CSR2(129)	1.58%	2.33%	8.94%
CSR3(104)	6.62%	9.40%	14.19%

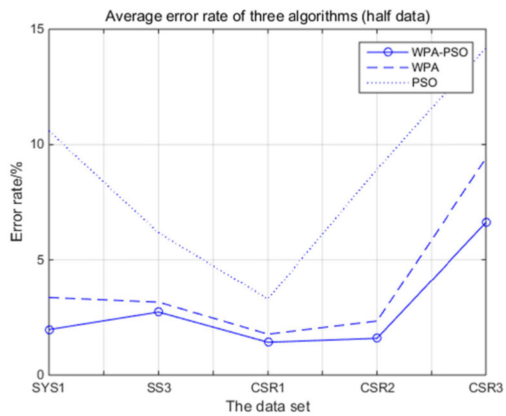

FIGURE 8. Average error rate of three algorithms (half data).

estimation results and actual results are shown in Table 12 and Figure 7:

It can be seen from Figure 7 that the error rate of WPA-PSO is the smallest in five data sets when the three algorithms estimate the parameters in the case of half data. And its error rate of the limited data of CSR3 is still the most optimal, and there is no sudden increase of the error rate of PSO and WPA in the CSR3. It is further explained that the WPA-PSO has obvious algorithm accuracy and stability in the case of limited data than a single algorithm.

The results of running the three algorithms for 20 times are statistically calculated. The average values and the average error rates are shown in Table 13, Table 14 and Figure 8:

In the case where the data set has only the first half failures, it can be seen from Tables 9 to 14 and Figures 7 and 8 that the accuracy estimated by the WPA-PSO is higher than that of the single algorithm. Moreover, it can be seen from Table 14 and Figure 8, the hybrid algorithm is more stable in 20 times, which is a powerful illustration of the effectiveness of the proposed method.

2) PREDICTION

When making predictions, we only use the first half of the data to make parameter estimation of the model, and then to predict the latter half of the data. So, the actual latter half of the data is not used in the parameter estimation of the model, and is only used to be the actual data as a comparison to the latter half data predicted in this paper.

The parameters in Table 9, Table 10 and Table 11 are respectively brought back to the formula (1), and the occurrence time of the latter half of the failure of the five data sets is predicted according to the formula. And compare the obtained prediction curve with the actual curve, as shown in Figure 9∼13:

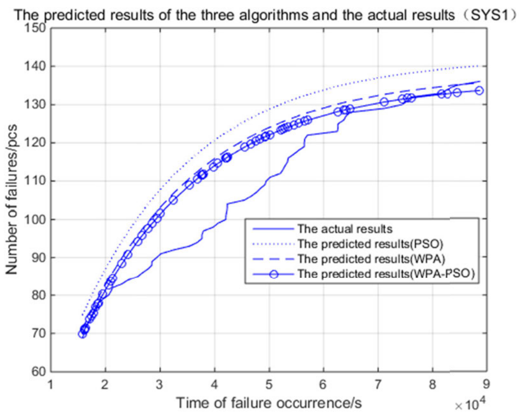

FIGURE 9. Actual and predicted results of three algorithms(SYS1).

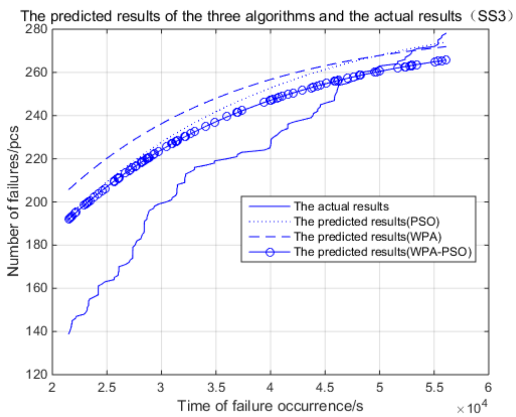

FIGURE 10. Actual and predicted results of three algorithms(SS3).

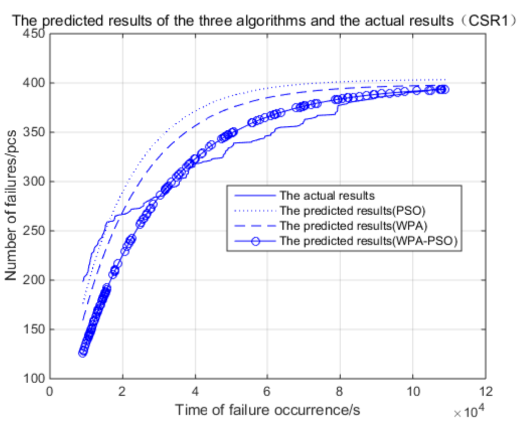

FIGURE 11. Actual and predicted results of three algorithms(CSR1).

Observed from Fig. 9∼13, it can be found that the curve predicted by the WPA-PSO proposed by the article is closer to the actual curve than the curve of the single algorithm. And the curve is exponentially distributed, and the slope of the curve is constantly increasing, indicating that the time interval in which software failure occurs is increasing. It shows that the reliability of the software is gradually improving, which is same with the fact that the reliability of the actual software

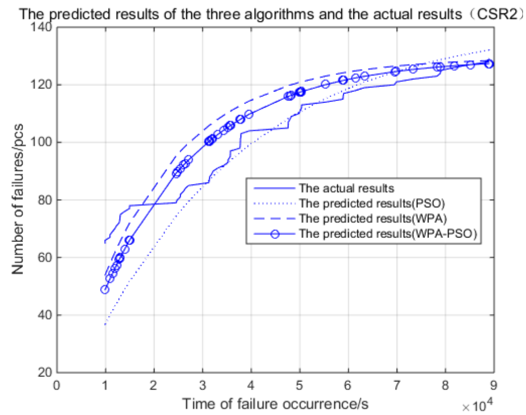

FIGURE 12. Actual and predicted results of three algorithms(CSR2).

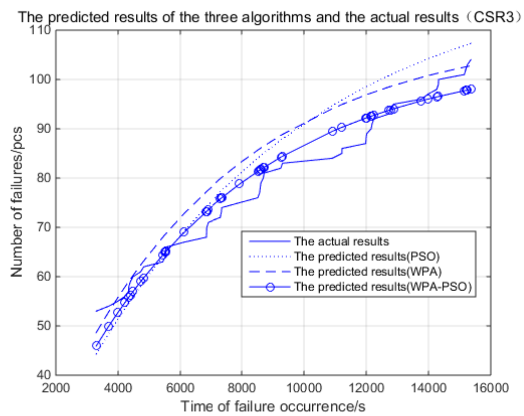

FIGURE 13. Actual and predicted results of three algorithms(CSR3).

test is improved with the discovery and modification of the failure.

It can be seen from the above that the WPA-PSO uses half data as the model parameter estimation, and then predicts the moment when the subsequent failure occurs by the model. And it shows an obvious advantage in the accuracy and stability of the estimation and prediction of a single algorithm.

C. EFFECT OF STABILITY

It is known that the actual failure number of the five data sets SYS1, SS3, CSR1, CSR2, and CSR3 are 136, 278, 397, 129, and 104, respectively. It can be seen from Figure 4 that although WPA-PSO and WPA have similar error rates, the values of multiple occurrences of the hybrid algorithm are closer to the actual values, and the values between the data after running multiple times are not much different. In other words, the WPA-PSO shows a better performance statistically.

This paper record the results of each algorithm running 20 times in each iteration of 500 times. The statistical distribution of each group of data is shown in Figure 14∼18 and the standard deviation of the 20 samples is shown in Table 15:

It can be found from Figures 14∼18 that the estimation results of WPA-PSO are closer to the actual values than the

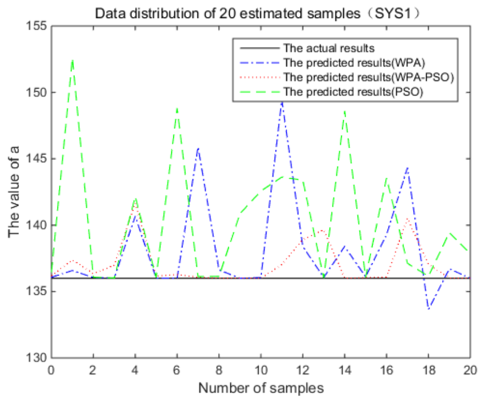

FIGURE 14. Data distribution of 20 estimated samples(SYS1-136).

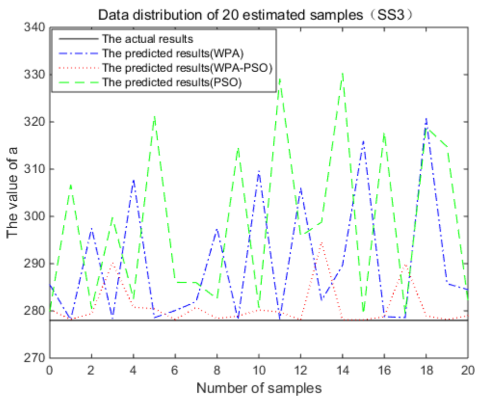

FIGURE 15. Data distribution of 20 estimated samples(SS3-278).

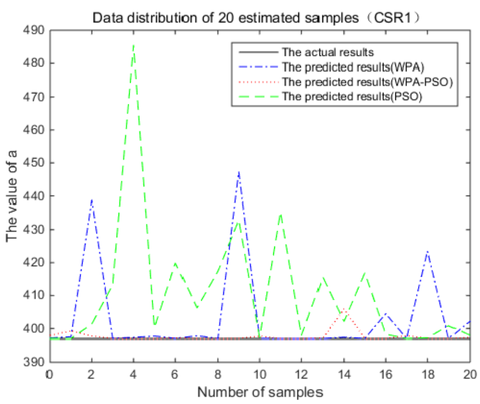

FIGURE 16. Data distribution of 20 estimated samples(CSR1-397).

other two algorithms. This shows that the WPA-PSO greatly improves the parameter estimation and prediction accuracy, and the randomness of the algorithm operation is also optimized. At the same time, it can be seen intuitively from the standard deviation of Table 15 that the stability of the WPA-PSO is the most optimal.

D. EFFECT OF ITERATIONS

In this section, we set the number of iterations for each algorithm to 50, 100, 200, 300, 400, 500, and run each algorithm 10 times to take the optimal value of each data set, and observe the effect of the number of iterations on each algorithm.

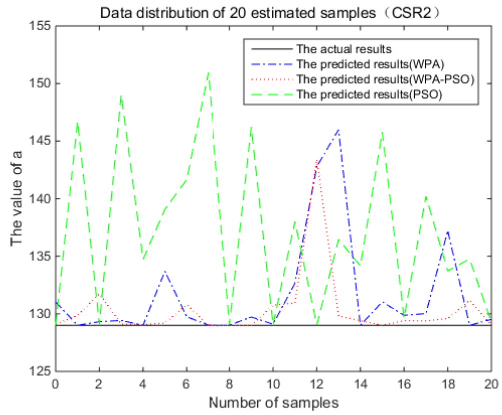

FIGURE 17. Data distribution of 20 estimated samples(CSR2-129).

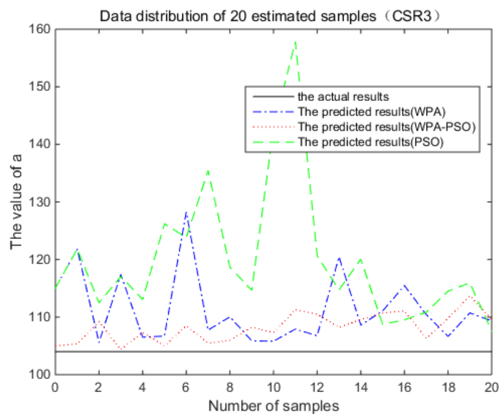

FIGURE 18. Data distribution of 20 estimated samples (CSR3-104).

TABLE 15. Standard deviation of three algorithms.

Data Sets (Failure number)	The standard deviation		
	WPA-PSO	WPA	PSO
SYS1(136)	1.69	3.98	5.13
SS3(278)	4.73	14.33	18.37
CSR1(397)	2.00	14.87	21.18
CSR2(129)	3.17	4.80	7.37
CSR3(104)	2.58	6.34	12.14

1) ANALYSIS OF ESTIMATED RESULTS

It is known that the actual failure number of the five data sets SYS1, SS3, CSR1, CSR2, and CSR3 are 136, 278, 397, 129, and 104, respectively. The following is an analysis and comparison of the estimated results of each data set under different iterations.

It can be seen from Figures 19∼23 that the WPA-PSO proposed in this paper can not only ensure that the error rate

TABLE 16. The estimated results of different iterations (SYS1-136).

The number of iterations	a		
	WPA-PSO	WPA	PSO
50	136.3831	138.6045	142.0130
100	136.3348	138.3760	142.2501
200	136.3679	139.2961	143.5944
300	136.2961	138.0755	143.0035
400	136.5401	138.6167	142.8487
500	136.4107	138.1190	142.4446

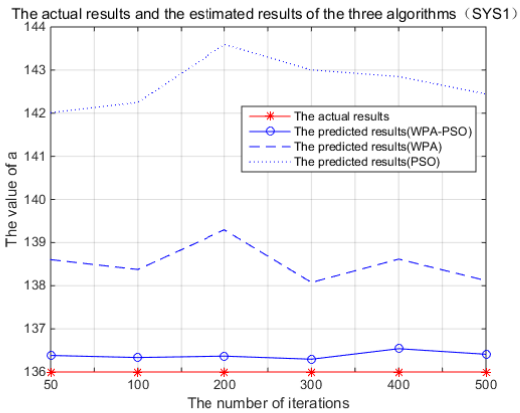

FIGURE 19. The actual results and the estimated results of the three algorithms (SYS1-136).

TABLE 17. The estimated results of different iterations (SS3-278).

The number of iterations	a		
	WPA-PSO	WPA	PSO
50	280.6023	301.5920	311.4316
100	280.2264	303.5995	316.9156
200	279.9986	301.0749	318.3507
300	280.0563	306.6212	311.2723
400	280.2122	304.4054	311.4530
500	280.1721	305.3720	310.1424

TABLE 18. The estimated results of different iterations (CSR1-397).

The number of iterations	a		
	WPA-PSO	WPA	PSO
50	397.2601	398.6832	404.5596
100	397.1873	399.6834	404.8332
200	397.0509	398.7648	402.7663
300	397.4131	398.2052	404.2071
400	397.2445	398.3097	404.5801
500	397.0864	398.3520	402.4343

is small, but also keep the error rate relatively stable under the influence of different iteration times.

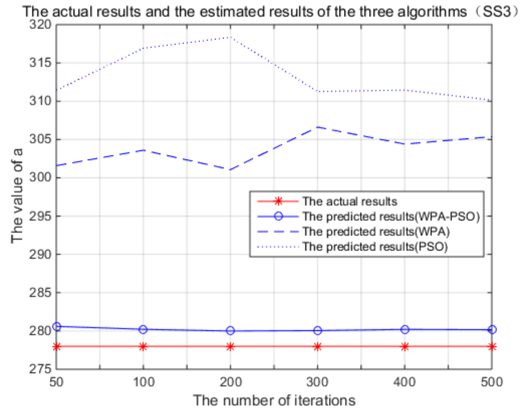

FIGURE 20. The actual results and the estimated results of the three algorithms (SS3-278).

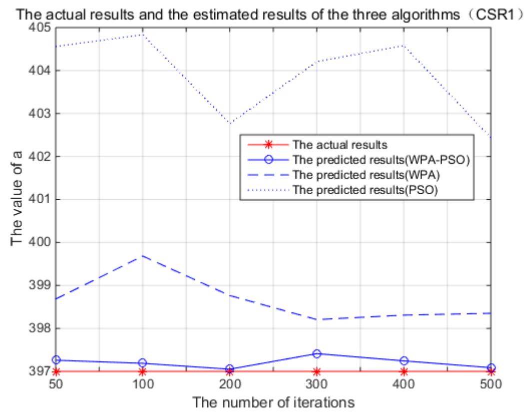

FIGURE 21. The actual results and the estimated results of the three algorithms (CSR1-397).

TABLE 19. The estimated results of different iterations (CSR2-129).

The number of iterations	a		
	WPA-PSO	WPA	PSO
50	129.2162	130.8766	133.2999
100	129.1912	130.6791	133.5929
200	129.3149	130.6903	132.9451
300	129.2088	130.7010	133.4774
400	129.2121	130.5397	132.9644
500	129.4038	130.4259	133.3615

2) ANALYSIS OF ERROR RATE

In order to more intuitively compare the influence of the number of iterations on the three algorithms, the error rates of the estimation results of the three algorithms and the actual results are calculated, as shown in Tables 21～25, and the error rates of the three algorithms are plotted as shown in Figure 24～28:

It can be found from Figure 24～28 that the error rate of the WPA-PSO remains the most optimal and stable for

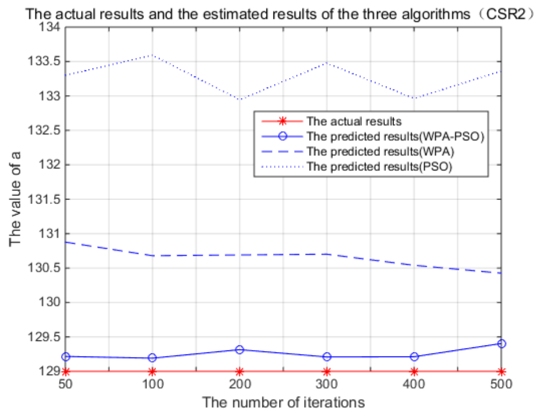

FIGURE 22. The actual results and the estimated results of the three algorithms (CSR2-129).

TABLE 20. The estimated results of different iterations (CSR3-104).

The number of iterations	*a*		
	WPA-PSO	WPA	PSO
50	104.0093	113.5274	131.7361
100	104.0002	113.4575	134.8978
200	104.0002	113.9451	133.3510
300	104.0001	113.5816	132.4938
400	104.0051	114.1294	132.4938
500	104.0109	113.1801	134.5478

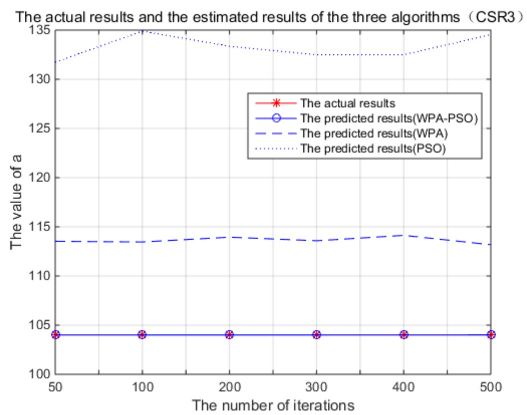

FIGURE 23. The actual results and the estimated results of the three algorithms(CSR3-104).

different iterations. Especially, it still shows the best results and stability on the limited dataset of CSR3. This shows that the WPA-PSO not only improves the accuracy of parameter estimation and prediction, but also improves the stability of the results.

3) ANALYSIS OF EACH ITERATION

In order to show how exploration and exploitation change, this part sets the maximum number of iterations of the three algorithms to 50, and runs three algorithms respectively. The

TABLE 21. Error rates of different iterations (SYS1).

The number of iterations	The error rate		
	WPA-PSO	WPA	PSO
50	0.28%	1.92%	4.42%
100	0.25%	1.75%	4.60%
200	0.27%	2.42%	5.58%
300	0.22%	1.53%	5.15%
400	0.40%	1.92%	5.04%
500	0.30%	1.56%	4.74%

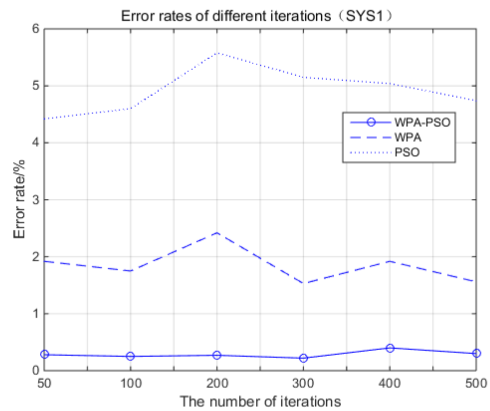

FIGURE 24. Error rates of different iterations (SYS1).

TABLE 22. Error rates of different iterations (SS3).

The number of iterations	The error rate		
	WPA-PSO	WPA	PSO
50	0.94%	8.49%	12.03%
100	0.80%	9.21%	14.00%
200	0.72%	8.30%	14.51%
300	0.74%	10.30%	11.97%
400	0.80%	9.50%	12.03%
500	0.78%	9.85%	11.56%

TABLE 23. Error rates of different iterations (CSR1).

The number of iterations	The error rate		
	WPA-PSO	WPA	PSO
50	0.07%	0.42%	1.90%
100	0.05%	0.68%	1.97%
200	0.01%	0.44%	1.45%
300	0.10%	0.30%	1.82%
400	0.06%	0.33%	1.91%
500	0.02%	0.34%	1.37%

results of lg(f-value) (f-value: value of the fitness function) are compared as shown in the Figure 29∼33:

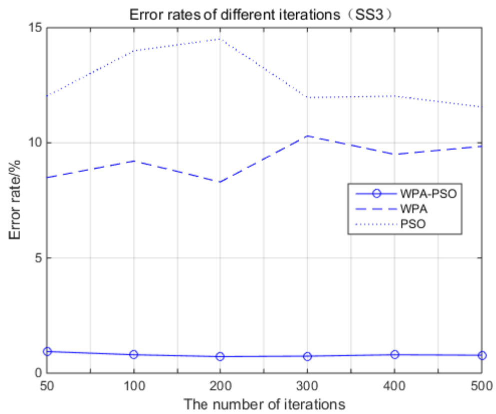

FIGURE 25. Error rates of different iterations (SS3).

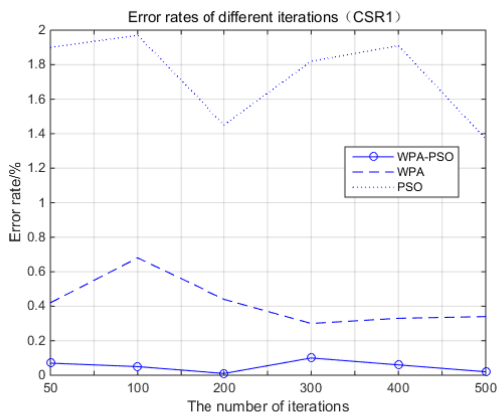

FIGURE 26. Error rates of different iterations (CSR1).

TABLE 24. Error rates of different iterations (CSR2).

The number of iterations	The error rate		
	WPA-PSO	WPA	PSO
50	0.17%	1.45%	3.33%
100	0.15%	1.30%	3.56%
200	0.24%	1.31%	3.06%
300	0.16%	1.32%	3.47%
400	0.16%	1.19%	3.07%
500	0.31%	1.11%	3.38%

It can be seen from Figure 29 to 33 that under the same conditions, the value of the fitness function of WPA-PSO is the smallest, indicating the advantage of the algorithm. At the same time, because the failure number of the last data set CSR3 is relatively small, the difference between the algorithms is more obvious in Fig33 which shows the WPA-PSO is much better especially in limited data. This article will continue to take the first half of the five data sets to observe the data changes.

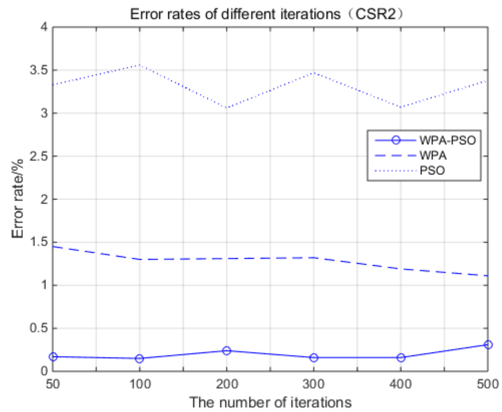

FIGURE 27. Error rates of different iterations (CSR2).

TABLE 25. Error rates of different iterations (CSR3).

The number of iterations	The error rate		
	WPA-PSO	WPA	PSO
50	0.01%	9.16%	26.67%
100	0.00%	9.09%	29.71%
200	0.00%	9.56%	28.22%
300	0.00%	9.21%	27.40%
400	0.00%	9.74%	27.40%
500	0.01%	8.83%	29.37%

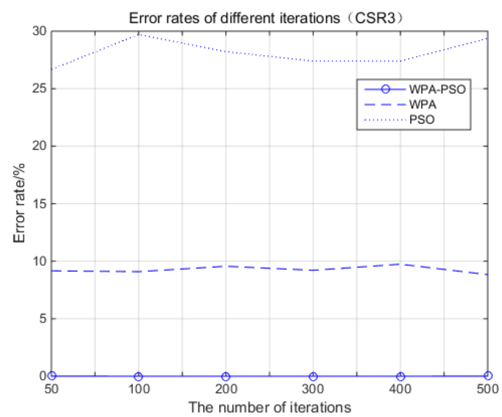

FIGURE 28. Error rates of different iterations(CSR3).

To test the performance in limited data of the three algorithms, the comparison results of half of the data sets are shown in Figure 34∼38:

It can be clearly seen from Figure 34∼38 that when the failure number becomes smaller, the convergence speed of WPA-PSO is also faster, and the fitness value of WPA-PSO is also the smallest, which indicates that the performance of the proposed algorithm is much better in limited data.

E. SUMMARY OF SIMULATION

In the experimental simulation of section IV, part A is the comparison of the estimation results of the three algorithms,

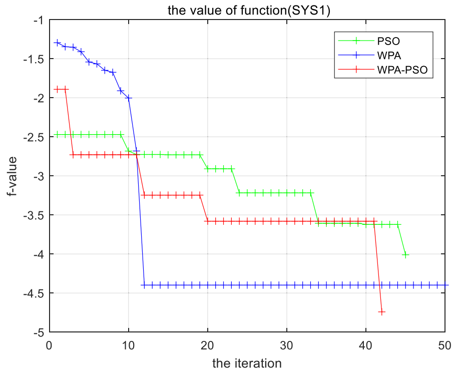

FIGURE 29. **The f-value changed with iteration (SYS1).**

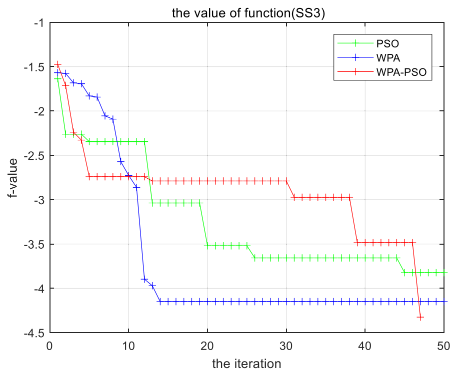

FIGURE 30. **The f-value changed with iteration (SS3).**

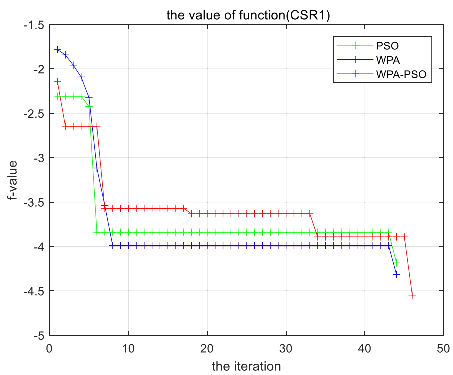

FIGURE 31. **The f-value changed with iteration (CSR1).**

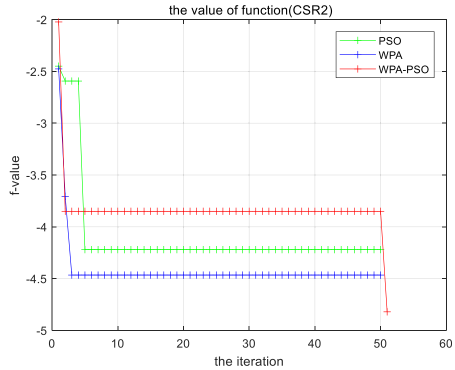

FIGURE 32. **The f-value changed with iteration (CSR2).**

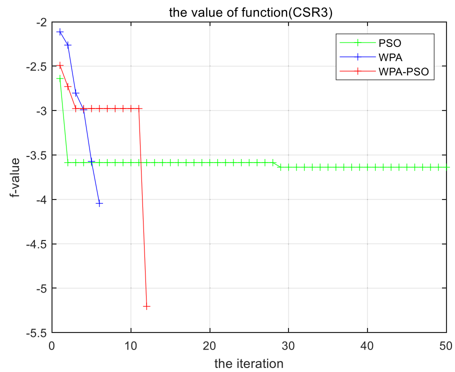

FIGURE 33. **The f-value changed with iteration (CSR3).**

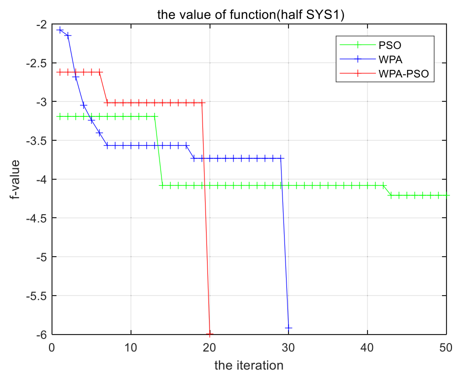

FIGURE 34. **The f-value changed with iteration (half SYS1).**

which shows the hybrid algorithm has the highest accuracy. The part B is the estimation using the first half failures and the estimation result is used to predict the last half failures. The results show that the estimation and prediction of the hybrid algorithm are better than the other two algorithms. Part C gives the sample results of 20 runs per algorithm and finds that the hybrid algorithm has best stability. Finally, part D studies the effect of the number of iterations of the algorithm on the results. It is found that the error rate of the hybrid algorithm remains the most optimal and stable.

Here we made an analysis on the results of experiments in section IV. As we all know, Particle Swarm Optimization (PSO) simply follows the historical optimal solution for random solution. So PSO is easy to stay in the local optimal solution rather than the global optimal solution. And although the algorithm is simple and the convergence speed is fast, it has strong randomness and is not stable enough. Different from the simple solving rules of PSO, the Wolf Pack Algorithm (WPA) has strict operation and control strategies, such as walking behavior, summoning behavior, siege behavior, the "winner is king" generation rule of the head Wolf and "strong survival" Wolf

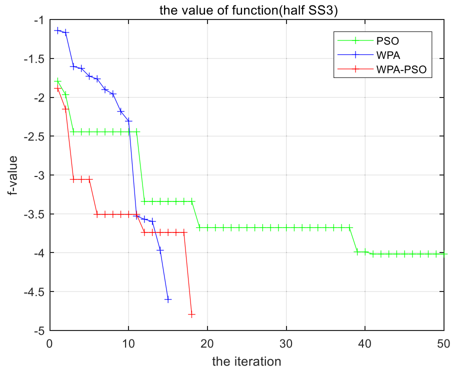

FIGURE 35. The f-value changed with iteration (half SS3).

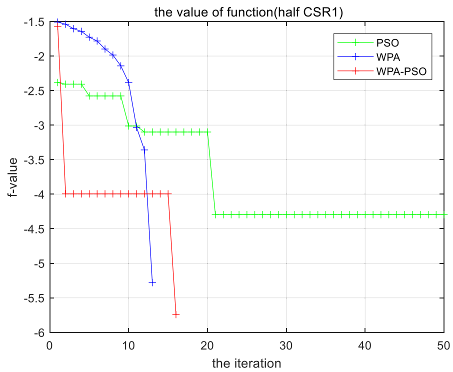

FIGURE 36. The f-value changed with iteration (half CSR1).

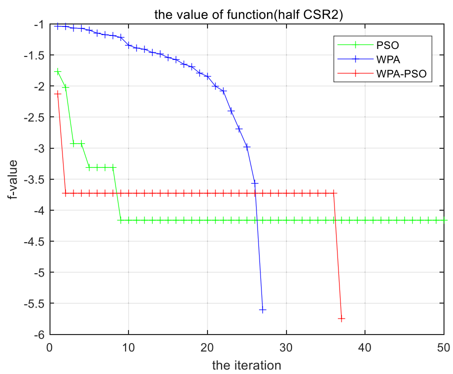

FIGURE 37. The value of the fitness function (half CSR2).

pack renewal mechanism. There is a whole set of systematic solution process of WPA, so even though the convergence speed of WPA is slow and the solution time is long, the solution obtained has good accuracy and stability. The hybrid algorithm proposed in the article combines the characteristics of the above two algorithms. It uses the wolf swarm algorithm on the historical optimal solution of the particle swarm algorithm, so that it has a comprehensively faster solution speed, accurate solution results and good stability.

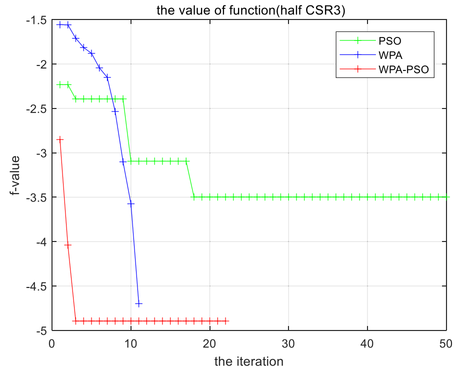

FIGURE 38. The value of the fitness function (half CSR3).

V. CONCLUSION

This paper put forward a hybrid WPA and PSO algorithm to estimate and predict failure data based on G-O model which is a typical software reliability model.

The experimental results show that the proposed hybrid method of WPA and PSO can improve the accuracy of parameter estimation and prediction of software reliability models. Especially in limited data (half data and limited test data set CSR3), the hybrid algorithm shows obviously higher accuracy than a single algorithm. And it also has a great improvement in stability compared to a single algorithm, improving the accuracy of the result in whole, and making the result closer to the actual value.

In this paper, the parameters of the classical GO model are estimated. Similarly, if the parameters of other software reliability models can be solute and then construct the fitness function, the same performance can be achieved by using the hybrid algorithm (WPA-PSO) proposed in this paper.

In the future research, reasonable probability sampling and rules can be considered to select some meaningful solutions of PSO solutions, and then the wolf group algorithm can be used to search the selected solutions again, so as to improve the efficiency of the algorithm.

REFERENCES

[1] F. Ogîgău-Neamţiu and H. Moga, "A cyber threat model of a nation cyber infrastructure based on goel-okumoto port approach," *Land Forces Acad. Rev.*, vol. 23, no. 1, pp. 75–87, Mar. 2018.

[2] E. Abuta and J. Tian, "Reliability over consecutive releases of a semiconductor optical endpoint detection software system developed in a small company," *J. Syst. Softw.*, vol. 137, pp. 355–365, Mar. 2018.

[3] S. Ameli, F. Naghdy, D. Stirling, G. Naghdy, and M. Aghmesheh, "Chemotherapy-induced fatigue estimation using hidden Markov model," *Biocybernetics Biomed. Eng.*, vol. 39, no. 1, pp. 176–187, Jan. 2019.

[4] H. W. Zhang and Y. Wu, "New swarm intelligence algorithm-Wolf Pack Algorithm," *Syst. Eng. Electron.*, vol. 35, no. 11, pp. 2431–2437, Aug. 2013.

[5] H. M. Zhuang and X. J. Jiang, "A wolf pack algorithm for active and reactive power coordinated optimization in active distribution network," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 40, no. 1, pp. 1315–1715, Aug. 2016.

[6] H. Li and H. Wu, "An oppositional wolf pack algorithm for Parameter identification of the chaotic systems," *Optik*, vol. 127, no. 20, pp. 9853–9864, Oct. 2016.

[7] C. Yongbo, M. Yuesong, Y. Jianqiao, S. Xiaolong, and X. Nuo, "Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm," *Neurocomputing*, vol. 266, pp. 445–457, Nov. 2017.

[8] X.-P. Xu, X.-T. Shi, and F. Wang, "Solving logistics distribution center location problem using a wolf pack algorithm," *J. Discrete Math. Sci. Cryptogr.*, vol. 20, nos. 6–7, pp. 1269–1273, Oct. 2017.

[9] Y. Chen, Z. Jia, X. Ai, D. Yang, and J. Yu, "A modified two-part wolf pack search algorithm for the multiple traveling salesmen problem," *Appl. Soft. Comput.*, vol. 61, pp. 714–725, Dec. 2017.

[10] R. Menassel, B. Nini, and T. Mekhaznia, "An improved fractal image compression using wolf pack algorithm," *J. Exp. Theor. Artif. Intell.*, vol. 30, no. 3, pp. 429–439, May 2018.

[11] Z. Han, X. Tian, X. Ma, and S. Lin, "Scheduling for re-entrant hybrid flowshop based on wolf pack algorithm," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 382, no. 3, pp. 1757–1763, Jul. 2018.

[12] L. Xu, B. Wang, X. Du, and Y. Hong, "Prediction method of mine gas emission based on complex neural work optimized by Wolf pack algorithm," *Syst. Sci. Control Eng.*, vol. 6, no. 3, pp. 85–91, Sep. 2018.

[13] Y. Gao, F. Zhang, Y. Zhao, and C. Li, "Quantum-inspired wolf pack algorithm to solve the 0-1 knapsack problem," *Math. Problems Eng.*, vol. 2018, Jun. 2018, Art. no. 5327056, doi: 10.1155/2018/5327056.

[14] L. Xiang, *Research and Application of Optimization of Wolf Group Algorithm*. Qingdao, China: Qingdao Technological Univ., 2018.

[15] L. Xu, Y. Jiang, and L. Wang, "Thermal decomposition of rape straw: Pyrolysis modeling and kinetic study via particle swarm optimization," *Energy Convers. Manage.*, vol. 146, pp. 124–133, Aug. 2017.

[16] Y. Ding, W. Zhang, L. Yu, and K. Lu, "The accuracy and efficiency of GA and PSO optimization schemes on estimating reaction kinetic parameters of biomass pyrolysis," *Energy*, vol. 176, pp. 582–588, Jun. 2019.

[17] L. A. Bewoor, V. C. Prakash, and S. U. Sapkal, "Production scheduling optimization in foundry using hybrid particle swarm optimization algorithm," *Procedia Manuf.*, vol. 22, pp. 57–64, 2018.

[18] H. E. Espitia and J. I. Sofrony, "Statistical analysis for vortex particle swarm optimization," *Appl. Soft. Comput.*, vol. 67, pp. 370–386, Jun. 2018.

[19] İ. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Appl. Soft. Comput.*, vol. 66, pp. 232–249, May 2018.

[20] *IEEE Recommended Practice on Software Reliability*, IEEE Standard 1633 TM-2008, IEEE, New York, NY, USA, 2008.

LI ZHEN was born in Xinghua, Jiangsu, China, in 1977. He received the degree in management from Nanjing Normal University, the bachelor's degree in computer application from Nanjing University, the M.S. degree in signal and information processing from the Jiangsu University of Science and Technology, Zhenjiang, in 2006, and the Ph.D. degree in aerospace system engineering from Beihang University, Beijing, in 2011.

From 2011 to 2014, he was a Lecturer with the School of Electronics and Information, Jiangsu University of Science and Technology. Since 2014, he has been an Assistant Professor. He is the author of more than 15 articles, and has two inventions. His research interests include reliability and system engineering, safety engineering, swarm intelligence, resilience, and software testing.

YANG LIU was born in Suqian, Jiangsu, China, in 1995. She received the bachelor's degree from the Jinling Institute of Technology, in 2018. She is currently pursuing the master's degree in electronics and communication engineering with the Jiangsu University of Science and Technology, Zhenjiang.

Her research interests include swarm intelligence and software reliability.

WANG DONGSHENG was born in Yancheng, Jiangsu, China, in 1982. He received the master's degree in computer science from the Jiangsu University of Science and Technology, in 2007, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2012. His primary research interests include software engineering, question answering, and natural language understanding.

ZHENG WEI was born in Chibi, Hubei, China, in 1982. He received the Ph.D. degree from the Institute of Biomedical Electronics, Nanjing University, in 2011. He has been with the School of Electronics and Information, Jiangsu University of Science and Technology, since 2011, and is currently an Assistant Professor. His research interests include adaptive filtering, time-frequency analysis, non-linear analysis, and non-stationary analysis of ECG signals.