ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 8, Issue 1, January 2019

The research of parameter estimation of software reliability model based on hybrid PSO-ABC algorithm

Miao Miao Yu, Bing Zhu, Zhen Li

Abstract*：** ***The software reliability is mainly obtained through modeling and estimating. The existing software reliability models are nonlinear, and the parameter estimation of these models is difficult. At the same time, there are many optimization methods for solving nonlinear function problems, such as artificial bee colony algorithm (ABC) and Particle Swarm Optimization (PSO). ABC has the characteristics of less control parameters, strong exploration ability, and high accuracy of solution. The PSO algorithm has the characteristics of relatively small amount of computation and fast search speed, but it has premature convergence (especially in dealing with complex multi-peak search problems) and the problem of poor local search ability. Therefore, the paper proposes parameter estimation method of software reliability model based on hybrid PSO-ABC, constructs a new fitness function based on maximum likelihood estimation, removes the wrong solution during the algorithm execution process, and adds knowledge to improve the solution accuracy. The article uses five classic sets of software failure data to estimate the GO model parameters and make predictions, and performs a variety of comparisons of the algorithm results. The experimental results show that the new fitness function is better, the solution of parameter estimation using hybrid PSO-ABC is more accurate, and the hybrid PSO-ABC has a great advantage in general and especially in limited data.

Keywords: **Artificial swarm algorithm; Particle swarm optimization; Software reliability model; Parameter estimation.**

I. INTRODUCTION

Software reliability is one of the important features of software quality. Its purpose is to quantify software reliability status and behavior, so it is more and more valued by researchers. Many scholars have studied the prediction of software reliability models. Cong Jina et al [1] studied the software reliability prediction model based on support vector regression for improved estimation of distributed algorithms. Ruchika Malhotra et al [2] proposed a reliability model based on particle swarm optimization, which is predicted by software reliability model. In the selection of parameters is very important. At the same time, researchers have published nearly a hundred software reliability models that play an important role in software prediction. Alaa Sheta [3] proposed a swarm optimization algorithm to solve the problem of non-linear function based on the swarm intelligence optimization algorithm, the PSO algorithm was used to optimize the parameters of the model to predict the number of software failures using the model. Tarun Kumar Sharma et al [4] proposed to use an improved ABC algorithm in the parameter estimation of software reliability growth model, the improved algorithm has the ability of bidirectional search, which makes the algorithm's global exploration ability stronger and the performance better, and predict the parameters of the model much accurately.

In this paper, particle swarm optimization, artificial bee colony algorithm and their hybrid algorithms are used to implement new software reliability model parameter estimation methods. The hybrid algorithm introduces the artificial bee colony search operator into the particle swarm optimization algorithm, retaining their respective advantages and making up for the shortcomings of both sides, whether it is better in both parameter estimation and model prediction.

II. BASIC CONCEPTS

A. Software reliability and model

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, we will choose the representative software reliability model GO model, and estimate its parameters [5]. The estimated function of the cumulative failure number in the software system is as follows:

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

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 8, Issue 1, January 2019
Where: *m(t)* denotes the expected function of the accumulated failure number up to the time t; *a* denotes the total number of failures expected to be detected in the software after the end of the test; *b* denotes the probability that the residual failure is found, the failure detection rate is a proportional constant in the range (0,1).

B. Particle Swarm Optimization

Particle Swarm Optimization (PSO) proposed by Eberhart and Kennedy in 1995[6].The PSO algorithm is a swarm intelligence-based stochastic optimization algorithm that simulates the foraging behavior of bird populations in the natural world. In the PSO algorithm, the potential solution to each problem is a particle in the search space. PSO first initializes a group of random particles and determines a fitness value of each particle in the population (the fitness value refers to the difference between the current position (solution) and the optimal position (solution). In each iteration of the search for the optimal solution, each particle will follow its two extremums to adjust its position. One is the optimal fitness value that the particle itself has found so far, called the individual extremum, the other is the optimal fitness value found so far by all other particles in the population, called the global extremum.

The particles follow these two extremums through an iterative search to determine the final optimal solution. The mathematical expression of the particle swarm optimization algorithm is represented as: in an n-dimensional search space, there is a population of m particles, ie, $X=\{X_1,\ldots X_2,\ldots X_m\}$and the position of the *i-th* particle is expressed as $X_i=\{X_i,X_{i2},\ldots X_{in}\}^T$, its speed is expressed as $V_i=\{V_{i1},V_{i2},\ldots V_{in}\}^T$.The individual extremum of the *i-th* particle is expressed as $P_{bi}=\{P_{bi1},P_{bi2},\ldots P_{bin}\}^T$, and the global extremum of the population is expressed as $g_b=\{g_{b1},g_{b2},\ldots g_{bn}\}^T$. According to the basic principle of the particle swarm algorithm, the *i-th* particle can update its velocity and position according to equations (2) and (3):

$$v_{id}^{(t+1)} = wv_{id}^{t} + c_1 rand_1()(p_{bid}^{t} - x_{id}^{t}) + c_2 rand_2()(g_{bd}^{t} - x_{id}^{t}) \quad (2)$$

$$x_{id}^{(t+1)} = x_{id}^{t} + v_{id}^{(t+1)} \quad (3)$$

Where: w is the inertia weight and the range is [0,1]; c_1 and c_2 are learning factors, usually the value is $c_1 = c_2$ =2; $rand1()$ and $rand2()$ are between (0,1) varying random numbers[7].

C. Artificial bee colony algorithm

Artificial bee colony algorithm (ABC) is a random optimization algorithm based on swarm intelligence proposed by Karaboga in 2005 to simulate the behavior of honeybees in nature [8]. In the artificial bee colony algorithm, the bees divide into three kinds according to different division of labor: honeybee, observation bee and detection bee, in which the number of honeybees, the number of observation bees and the number of solutions in the optimization problem are equal. The process of collecting honey by honeybees is abstracted as the process of searching for optimal solutions. The position of each honey source represents a possible solution. The amount of nectar of the honey source corresponds to the quality of each possible solution, and is represented by the fitness value.

The algorithm is described as follows: first, the initial population is randomly generated, that is, *N* initial solutions. Each solution $X_i(i=1,2,\ldots N)$ represents a *D*-dimensional vector, D represents the number of optimization parameters. Then, each of the three bees began to search for all the initial solutions cyclically. The honeybee remembers the best solution to his own history and searches in the neighborhood. The search formula is (4):

$$x_{ij}^{'} = x_{ij} + \varphi(x_{ij} - x_{kj}) \quad (4)$$

there, $k \in \{1,2,\ldots N\}$, $j \in \{1,2,\ldots D\}$, and $k \neq i$, φ is a random number between [-1,1].

The honeybees compare their own historical best solution and neighborhood search solution. When the neighborhood search solution is better than its own historical optimal solution, it replaces its own historical best solution. Otherwise, it remains unchanged. When all the honeybees' searches are completed, they share the honey source information with the observation bee through the dance area. The observation bee selects a honey position as the current historical optimal position according to the probability associated with the nectar amount (solution); honeybees with a large amount of nectar attract bees that have a greater probability of observing bees than those with a small amount of nectar. Observation bee updates historical optimal location like the honeybees, and checks the amount of nectar in the new location. If the new location is better than the historical optimal location, the

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 8, Issue 1, January 2019
historical optimal location is replaced with the new location; otherwise, it remains the unchanged. The probability P_i of the observation bees choosing a nectar source is calculated as (5):

$$P_i = \frac{f_i}{\sum_{n=1}^{N} f_n} \quad (5)$$

There, f_i denotes the fitness value of the *i-th* solution, and N denotes the number of nectar sources.
If a particular nectar source has not changed after a defined number of cycles, then the honeybee at that nectar source becomes a detection bee, and the nectar source will be replaced by the random new location found by the detection bee. Assuming that the abandoned position is x_i, the formula for detecting bee replacement is (6):

$$x_{ij}^{'} = x_{\min}^{j} + rand()(x_{\max}^{j} - x_{\min}^{j}) \quad (6)$$

There, $x_{\max}^{j}$ and $x_{\min}^{j}$ are the upper and lower limits on the *j-th* dimension, and *rand()* is the random number between [0,1]. The above formula is also the formula for generating N initial solutions randomly.

D. PSO-ABC Hybrid Algorithm
The new algorithm proposed in this paper is called PSO hybrid ABC algorithm. After updating the speed and position of each particle based on the PSO calculation process, the search operator in the artificial bee colony algorithm is applied to the particle swarm algorithm, after each particle's position is updated, using the artificial bee colony search operator to search again around the new location to determine the final new location. The advantages and disadvantages of the two algorithms, PSO and ABC, are well utilized to achieve better results.

III. METHOD DESCRIPTIONS
Papers [9] and [10] have respectively used particle swarm optimization and ant colony algorithm to estimate the reliability of GO model parameters. What is common to these two methods is that they construct a fitness function and turn the parameter estimation problem into a function optimization problem. The fitness function constructed is as follows:

$$J = \sqrt{\sum_{t=0}^{T} [m(t) - \hat{m}(t)]^2} \quad (7)$$

In the formula (7), J represents the Euclidean distance between the actually measured software failure number and the software failure number estimated by the model. The smaller J is, the higher the accuracy of the model prediction is and the better the parameter estimation is. *m(t)* represents the cumulative number of failures actually found during the test period [0, t). *m^(t)* denotes the cumulative failure number estimated by the model in the test time period [0, t). *t* denotes the time when failure occurred and *T* represents the time to terminate the test.

This paper proposes a new method that constructs a new fitness function by using the maximum likelihood estimation formula to estimate the parameters of software reliability model, and eliminates those obviously erroneous solutions during the execution of the algorithm, search in the direction of more accurate solution based on prior knowledge.

A. Construction of new fitness function
We construct a new fitness function according to the maximum likelihood estimation formula of GO model. The equations of parameters a and b are written as below:

$$\begin{cases} a = \dfrac{n}{1 - e^{-bt_n}} \\ \dfrac{n}{b} = at_n e^{-bt_n} \end{cases} \quad (8)$$

Where: n is the number of known failures; t_i is the moment of the *i-th* failure; $i = 1, 2, 3, \ldots n$. From formula (8), if we can get the number of failures and the time at which each failure occurs, then it is feasible to solve the parameters including a and b using the maximum likelihood formula. However, the calculation process may be

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 8, Issue 1, January 2019

tedious, especially for some more complex models. Nevertheless, we can construct a new fitness function based on the maximum likelihood formula to realize the iterative search algorithm simply and get a more accurate solution.

Then substitute the first item in equation (8) into the second term and mathematically transform it into a formula only related to the parameter *b* as shown below:

$$f = \left| b - \frac{n(1-e^{-bt_n})}{nt_n e^{-bt_n} + (1-e^{-bt_n})\sum_{i=1}^{n} t_i} \right| \quad (9)$$

f is the new fitness function, parameters in the formula are all known except *b*.

B. Elimination of problem solution

In the algorithm implementation of the G-O model, since the model parameter *b* is randomly initialized within the range of (0, 1), some problem solutions may occur during the iterative search of the algorithm. In order not to affect the experimental results, we need to eliminate the problems in the experiment.

According to several experiments, it can be found that when the precision of the parameter *b* reaches 1e-6 or higher, a problem solution will occur. This may be caused by too much high precision which is beyond the limited length of bytes in Windows operation system. Therefore, in the program, the limit condition added to the parameter *b*, and the precision of the control parameter *b* is kept within 1e-5.

C. An improved solution of adding prior knowledge

The prior knowledge here can get from formula (8). According to formula (8), it can be seen that the parameter *a* and *b* are in reverse relation when in the same *t*: *b* is larger, *a* is smaller, and *b* is smaller, then *a* is larger. If the cumulative failure number *a* obtained from the result *b* of the first operation is larger than the known failure number, it is desirable that the value of *a* should become smaller, then the parameter *b* will be selected from larger solutions. If the parameter *a* obtained from the result *b* of the first run is smaller than the known number of failures, and the value of *a* is expected to become larger, then the value of the parameter *b* will be selected from smaller solutions. Thus, by the iterations of the next round of algorithms, parameter solutions will be found more quickly.

IV. EXPERIMENT DESCRIPTIONS

This paper got experimental data from five sets of interval data sets of software failure: SYS1, SS3, CSR1, CSR2 and CSR3 obtained in the actual industrial project. The data download address is http://www.cse.cuhk.edu.hk/lyu/book/reliability/data.html. [9]Units of the five data sets unify in seconds.

A. Comparison between a hybrid algorithm and single algorithm

Based on the respective advantages and disadvantages of the PSO algorithm and the ABC algorithm, this section proposed a PSO-ABC hybrid algorithm to estimate the parameters of the GO model based on that proposed in this paper, and compare the execution results with a single algorithm.

1. Parameter Estimation

In the experiment, the first step is to initialize. The three algorithms runs for 20 times. According to the principle in Section 3, the best result was taken as a solution. The experimental results are shown in Table 1The comparison with the results of a single algorithm is shown in Table 2:

Table 1 PSO-ABC estimation results based on new fitness function

Table 1data sets （Actual Failures）	parameter estimation results of GO model		the fitness value
	a	*b*	*J*
SYS1(136)	136.0496	8.9268e-5	4.8916e-5
SS3(278)	278.1835	1.3063e-4	8.7711e-5

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 8, Issue 1, January 2019

CSR1(397)	397.1000	7.0061e-5	2.5901e-5
CSR2(129)	129.0286	9.4567e-5	5.0713e-5
CSR3(104)	104.0008	7.6611e-4	5.7188e-4

Table 2 Comparison of the estimation results of the three algorithms

data sets (Actual Failures)	GO model parameter a estimate		
	PSO	ABC	PSO-ABC
SYS1(136)	136.0416	136.7724	136.0496
SS3(278)	278.2565	278.0001	278.1835
CSR1(397)	397.0031	399.0705	397.1000
CSR2(129)	129.0001	129.0110	129.0286
CSR3(104)	116.4310	113.2761	104.0008

From Table 2, it can be found that: 1)The three methods all have a very accurate solution on the first four data sets.2)The PSO and ABC also has a unsatisfied solution on data set CSR3.3)The hybrid method has a solution very close to the actual failure in CSR3 that make a great advantage to single PSO and ABC.3)The hybrid method has a minimum mean square error which mean it has a better performance in general

2. Estimation and prediction

In this section, we implement parameter estimation and model prediction based on the PSO-ABC hybrid algorithm. We estimate the parameters of the GO model by using the first half data of the five data set, and then substitute the estimated parameters into GO function expression of the model, the failure occurrence moment of the second half failure is predicted, and the execution effect of the hybrid algorithm is viewed and compared with the result of a single algorithm.

The algorithm is run 20 times initially, and the best results are taken according to the principle in the section 3. The experimental results are shown in Table 3. Table 4 shows the comparison with the results of a single algorithm:

Table 3 PSO-ABC estimation results based on new method (Half the number of failures)

data sets (Actual Failures)	parameter estimation results of GO model		the fitness value
	a	*b*	*J*
SYS1(136)	135.0663	4.5826e-5	1.7114e-6
SS3(278)	275.7797	3.2601e-5	1.6220e-6
CSR1(397)	397.8557	7.6521e-5	3.8249e-6
CSR2(129)	124.4872	7.4575e-5	4.4078e-6
CSR3(104)	109.9090	1.9450e-4	1.1743e-5

Table 4 Comparison of the estimation results of the three algorithms (Half the number of failures)

data sets （Actual Failures）	GO model parameter *a* estimate		
	PSO	ABC	PSO-ABC
SYS1(136)	145.5578	140.2974	135.0663
SS3(278)	301.2871	282.3210	275.7797
CSR1(397)	385.0133	381.0109	397.8557
CSR2(129)	126.6821	127.4061	124.4872
CSR3(104)	165.4456	125.5690	109.9090

It can be found from Table 4 when only the first half failures is known:1)PSO-ABC hybrid algorithm has the ability to acquire better solution in general considering the mean square error.2)PSO-ABC hybrid algorithm has a great advantage on limited data considering the solution of CSR3 which much better than single PSO and ABC

The parameters in Table 3 and Table 4 are brought back to formula (1) below. According to the formula, we predicted the occurrence moment of the second half failures of the five data sets, and the obtained prediction result curve is compared with the actual curve, such as Figure 1~5 shows:

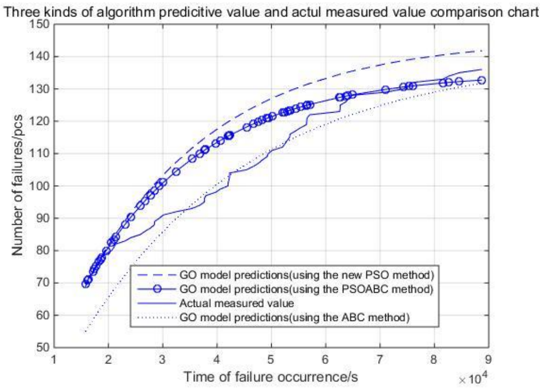

Fig. 1. Occurrence moment of the latter half of failures of SYS1 dataset

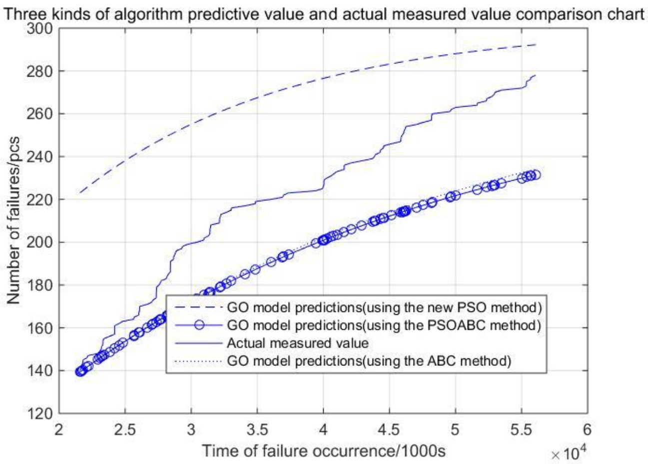

Fig. 2. Occurrence moment of the latter half of failures of SS3 dataset

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 8, Issue 1, January 2019

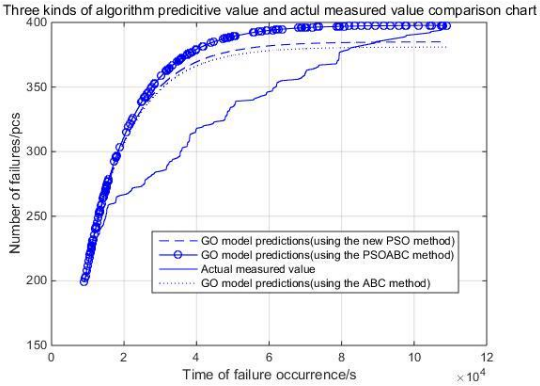

Fig.3. Occurrence moment of the latter half of failures of CSR1 dataset

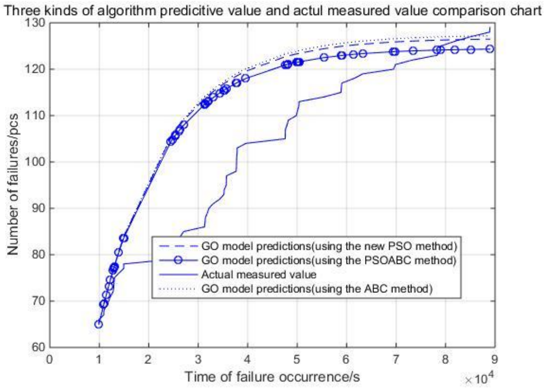

Fig.4. Occurrence moment of the latter half of failures of CSR2 dataset

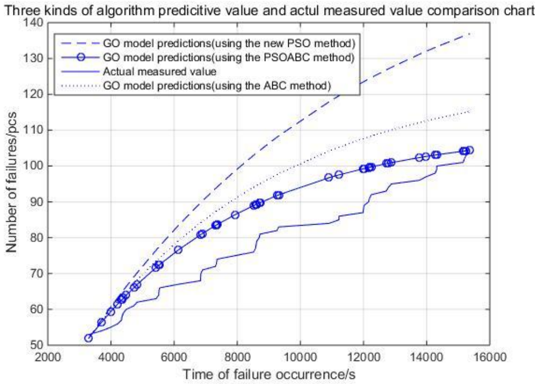

Fig.5. Occurrence moment of the latter half of failures of CSR3 dataset

From figures 1 to 5, it can be found that:1)The curves predicted by the three algorithms are compared with the actual curves, although there are certain errors, but the basic trends are consistent.2)The PSO-ABC hybrid algorithm is simple to implement and has a fast convergence speed.3)Using the PSO-ABC hybrid algorithm to estimate the model parameters and predict the moment that subsequent failure occurs is feasible in practice, and is more accurate and reasonable than single PSO and ABC.

V. CONCLUSION

This paper firstly compared the PSO-based software reliability model parameter estimation method proposed by previous researchers with a proposed method using a new fitness function. The PSO-ABC hybrid algorithm proposed in this paper was used to estimate the model parameters and the results were compared. The results on whole and half data showed that the hybrid algorithm is not only simple and efficient, but also the accuracy of the

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 8, Issue 1, January 2019
parameter estimation results is greatly improved compared to a single algorithm, and it can well meet the actual needs especially in limited data and testing time.

REFERENCES

[1] Cong Jin, Shu-Wei Jin. Software reliability prediction model based on support vector regression with improved estimation of distribution algorithms [J]. Applied Soft Computing Journal, 2014, Vol.15, pp.113-120.

[2] Ruchika Malhotra, Arun Negi.Reliability modeling using Particle Swarm Optimization[J].International Journal of System Assurance Engineering and Management, 2013, Vol.4 (3), pp.275-283.

[3] Alaa Sheta. Reliability Growth Modeling for Software Fault Detection Using Particle Swarm Optimization[C]. 2006 IEEE Congress on Evolutionary Computation. Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada July 16-21, 2006.

[4] Tarun Kumar Sharma, Millie Pant, Ajith Abraham. Dichotomous Search in ABC and its Application in Parameter Estimation of Software Reliability Growth Models[C]. 2011 Third World Congress on Nature and Biologically Inspired Computing, 2011:207-212.

[5] A.Tang，Q.Y.Huang. Research on Goel-Okumoto Software Reliability Model [J]. Science and Technology Information，2009, 31:3-4.

[6] Harish Garg , MonicaRani. An approach for reliability analysis of industrial systems using PSO and IFS technique [J]. ISA Transactions，2013, Vol (52), pp: 701–710.

[7] Liu Junfang. Hybrid Optimization algorithm of Particle Swarm Optimization and artificial Bee Colony to optimize SVM parameters and its Application [D]. Master thesis of Taiyuan University of Science and Technology.2012:22-23.

[8] D.Karaboga , B.Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm. Journal of Global Optimization, 2007, 39(3):459-471.

[9] K.H.Zhang, A.G.Li, B.W.Song. Parameter estimation method of software reliability model based on PSO [J]. Computer Engineering and Applications, 2008, 44(11):47-49.

[10]C.Y.Zheng, X.M.Liu, S.Huang. Parameter estimation of software reliability model based on ant colony algorithm [J]. Computer application, 2012，32 (4):1147-1151.

AUTHOR BIOGRAPHY

Miao Miao Yu(1994-),Female, Master 's Degree, Department of Electronics and Information, Jiangsu University of Science and Technology, The main research direction is software reliability.

Bing Zhu(1980-),Male, Master's Degree, senior engineer, shanghai merchant ship design and research institute, The main research direction is electrical Automation.

Zhen Li(1977-), Male, Associate Professor, Ph.D., Department of Electronics and Information, Jiangsu University of Science and Technology, Golden Ship Software Ltd ,The main research directions : system safety ,system reliability.