IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 4 Issue 2, February 2017
ISSN (Online) 2348 – 7968 | Impact Factor (2016) – 5.264
www.ijiset.com

A new parameter estimation method for software reliability model based on ABC

Min Mao[1], Zhen Li[1,2]*, Yang Li[1], Hong Miao[3], Min Wang[1]

1 Department of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, 212003, China

2 Golden Ship Software Ltd, Zhenjiang, Jiangsu, 212003, China

3 Department of Economics and Management, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, 212003, China

Abstract

The parameter estimation results of software reliability model will affect the accuracy of software reliability prediction. Most of the existing software reliability models are nonlinear, and it is difficult to estimate their parameters. Artificial bee colony algorithm (ABC) is a kind of stochastic optimization method which is suitable for solving nonlinear function; it has the characteristics of few control parameters, strong exploration ability and high accuracy. Based on this, a new parameter estimation method of software reliability model based on ABC is proposed in this paper. First, it's combined with maximum likelihood estimation to construct a new fitness function, second, it can clear the obvious wrong solution in the process of implementing the algorithm, and at last, the prior knowledge is added to improve the accuracy of solution. In the following content, we estimate the parameters of the GO model with 5 sets of classic software failure data, and to predict and compare. The experimental results show that the method is simple and easy to implement, the accuracy of parameter estimation is high, and it has a better prediction effect.

Keywords: *artificial bee colony algorithm; software reliability model; parameter estimation; model prediction*

1 Introduction

Software reliability is one of the important characteristics of software quality, and is the main quantitative criterion for evaluating software quality, so it is paid more and more attention by researchers. Software reliability modeling is one of the important areas of theoretical research and engineering practice for software reliability. Its purpose is to quantify software reliability state and behavior, to help develop reliable software and to test the reliability of software. So far, researchers have published nearly 100 kinds of software reliability models, such as G-O model [1], M-O model [2] and J-M model [3]. However, these models are basically non-linear function models; it is difficult to estimate their parameters.

Because of the nonlinearity of the software reliability model, a new idea is to apply the intelligent optimization algorithm to the estimation of model parameters. Artificial bee colony algorithm (ABC), which is a kind of group intelligent optimization algorithm, is characterized by simple operation, few control parameters and excellent performance. It has been widely used in the fields of function optimization, pattern recognition, moving target tracking and robot path planning has been widely used.

The group intelligent algorithm has been widely used in power system, aerospace, wireless sensor network and other fields. But the research in the reliability is relatively scarce: Harish Garg et al [4] proposed PSO algorithm for industrial system reliability analysis, through the PSO algorithm to optimize the key parameters of the system to improve the industrial system performance and reliability; Tarun Kumar Sharma [5] proposed an improved ABC algorithm for the software reliability growth model parameter estimation, the improved algorithm has the ability of dichotomous search, which makes the algorithm has more powerful global exploration and better performance; Alaa Sheta [6] proposed the PSO algorithm for the problem of software reliability growth model, through the PSO algorithm optimization model parameters, so as to better predict the number of software failures .

Similarly, there is a relative lack of research on the use of group intelligent optimization algorithms for reliability, especially for artificial bee colony algorithms. Zhengchu Wang et al[7] proposed the PSO algorithm to solve the complex system reliability optimization problem, and verified the feasibility and validity of the algorithm through two examples; Kehan Zhang et al [8] used the PSO algorithm for software reliability, the shortcomings are that the search range is large and the convergence speed is slow; Changyou Zheng et al [9] proposed the ant colony algorithm used in the software reliability model parameter estimation, the advantage is that the speed and accuracy of the convergence is high, but the disadvantage is that the algorithm is more complex and computationally intensive.

In view of the above research status and existing problems, a new method for parameter estimation of software reliability model based on ABC is proposed in this paper. The difference of this method lies in constructing a new fitness function, eliminating those obviously wrong solutions in the progress of algorithm execution, and making the algorithm search in the direction of more accurate solutions according to the prior knowledge.

2 Basic Concepts

2.1 Software reliability and model

Software reliability is the probability that the software will not cause a system failure under the specified conditions and time. IEEE Computer Society has made the following definition of software reliability [10]: 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 of software reliability is a mathematical method to evaluate software reliability, and the selection 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 as the research object, and estimate its parameters. 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) 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).

It can be seen that the parameters of GO model are a and b, their selection will affect the accuracy of the model prediction. The following is a formula using the maximum likelihood method to estimate the a, b :

$$\begin{cases} a = \dfrac{n}{1 - e^{-bt_n}} \\ \dfrac{n}{b} = at_n e^{-bt_n} \end{cases} \tag{2}$$

Where: n is the number of known failures; t_i is the time when the i-th failure occurred: i=1,2,3,...n.

From the formula (2), if the number of failures and the time which each failure occurs are known, using maximum likelihood formula to solve the parameters of a and b is feasible and more accurate, but the calculation process will be more cumbersome. Especially for some more complex models, it is difficult to solve the parameters directly according to the maximum likelihood formula. However, if we construct a new fitness function according to the maximum likelihood formula and design an algorithm to iterate search, it is not only easy to implement, but also the estimated parameters will be more accurate.

2.2 Artificial bee colony algorithm

Artificial bee colony algorithm (ABC) [11] is a random optimization algorithm based on swarm intelligence proposed by Karaboga in 2005 to simulate the behavior of honey bees in nature. ABC algorithm has been widely used in the fields of numerical optimization, constraint optimization, robot path planning and so on, because of its simple operation, few control parameters and excellent performance.

In the artificial bee colony algorithm, the bees can be divided 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 the bee harvesting is abstracted as the process of searching for the optimal solution. The location of each nectar represents a possible solution, and the nectar amount of the nectar source corresponds to the mass of each possible solution, expressed as a fitness value.

The algorithm is described as follows: First, the initial population is generated randomly, that is, N initial solutions. Each solution Xi (i = 1,2, ... N) represents a vector of D dimensions, and D represents the number of optimization parameters. Then, the three kinds of bees began to loop search for all the initial solution. The bee keep remember the optimal solution they have found, and then search in the neighborhood , the search formula shown as(3) :

$$x_{ij}^{'} = x_{ij} + \phi(x_{ij} - x_{kj}) \tag{3}$$

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

Compared with the historical optimal solution and neighborhood search solution, the honeybee replaces its historical optimal solution when the neighborhood search solution is superior to its own historical optimal solution, otherwise, it keeps unchanged. When all the honeybees finish search, they share the nectar information with the observation bees through the dance area. The observation bees choose a nectar source position as the current historical optimal position (solution) according to the nectar-related probability. Nectar source with a large number of nectar has the higher probability of attracting the observation bees than the small amount of nectar. The observation bee updates the historical optimal position according to Eq(3), and check the nectar of the new position. If the new position is better than the historical optimal position, the historical optimal position is replaced with the new position; otherwise, it remains unchanged. The probability P_i of the observation bees choosing a nectar source is calculated as (4):

$$p_i = \frac{f_i}{\sum_{n=1}^{N} f_n} \tag{4}$$

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 position has not changed after a defined number of cycles, the honey bee at that nectar source becomes a detection bee, which will be replaced by a random new position found by the detection bee. Suppose the abandoned position is x_i , the detection bee replace x_i according to formula (5):

$$x_{ij}^{'} = x_{\min}^{j} + rand()(x_{\max}^{j} - x_{\min}^{j}) \tag{5}$$

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

3 Method Descriptions

In the literature [8] and [9], PSO and ant colony algorithm have been used to estimate the parameters of the software reliability model. What these two methods have in common is to construct a fitness function which transforms the parameter estimation problem into the function optimization problem. The fitness function is constructed as follows:

$$J = \sqrt{\sum_{t=0}^{T}[m(t) - \hat{m}(t)]^2} \tag{6}$$

In the formula (6), 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, the better the parameter estimation is; m(t) represents the cumulative number of failures actually found during the test period [0, t); $m^{\wedge}(t)$ denotes the cumulative failure number estimated by the model in the test time period [0, t); t denotes the time at which the failure occurred; T represents the time to terminate the test.

Due to the above-mentioned fitness function, the parameters of the software reliability model need to be initialized randomly, this leads to the problems of large search range, slow convergence speed and low accuracy.

Aiming at these problems, this paper proposes a new method to construct a new fitness function by the maximum likelihood estimation formula of the software reliability model parameters. And then eliminate those obvious errors in the process of algorithm implementation. Finally, search in the direction of more accurate solutions according to the prior knowledge will be described in more details below.

3.1 Construction of fitness function

We construct a new fitness function according to the maximum likelihood estimation formula of GO model parameters a and b. By substituting the first term in equation (2) into the second term and transforming, constructing a formula related only to the parameter b, as follows:

$$f = \left| b - \frac{n(1-e^{-bt_n})}{nt_n e^{-bt_n} + (1-e^{-bt_n})\sum_{i=1}^{n} t_i} \right| \tag{7}$$

f is the new fitness function, parameters in the formula are all known in addition to b , the smaller f is, the better the estimated parameter b is. The individual extremum and the global extremum can be updated according to the fitness value.

By the iterative search of ABC algorithm, the algorithm will output the optimal parameters b when it achieves the stop criteria. And then substitute b into the first parameter a's maximum likelihood estimation formulate in the formula (2) to solve the corresponding optimal parameters a.

3.2 Elimination of problem solution

In the algorithm implementation of GO model, since the model parameter b is randomly initialized in the range of (0, 1), some problem solutions may arise in the iterative search of the algorithm. In order not to affect the experimental results, we need to eliminate the problem solutions. According to several experiments we can find that, when the precision of the parameter b reach to 1e-6 or higher, there will be a problem solution. Therefore, we add restrictions to the parameter b in the program, control b to maintain the accuracy within 1e-5. So as to achieve the purpose of eliminating problem solutions, making the algorithm continue to search in a better range.

3.3 An improved solution of adding prior knowledge

The prior knowledge is obtained from Eq. (2). According to Eq. (2), we know that the parameters a and b are inversely related: if b is large, then a is small; if b is small, then a is large.

If the cumulative failure a calculated from the result b of the first run is greater than the known failure number, hope the value of a to be smaller, then the value of the parameter b should be larger according to the prior knowledge, so continue to run the program to find the larger b. If the cumulative failure a calculated from the result b of the first run is less than the known failure number, hope the value of a to be larger, then the value of the parameter b should be smaller according to the prior knowledge, so continue to run the program to find the smaller b. As a result, as the start of the iteration of the next round of algorithms, more accurate parameters can be obtained.

4 Example Descriptions

4.1 Parameter estimation

The experimental data of this paper are derived from the five sets of software failure interval data sets 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[8]. Because the GO model is involved in the software failure occurs time, so firstly we must transfer the five sets of software failure interval time series into software failure occurs time sequence, and units are unified with seconds. Then we will compare the software reliability model parameter estimation method based on PSO proposed in the literature [8] with the new software reliability model parameter estimation method based on ABC proposed in this paper.

The parameters of ABC algorithm are set as follows: the number of nectar sources m = the number of candidate solutions = 60, the maximum number of iterations Tmax = 500, the maximum number of times that a nectar source is mined = 100, the fitness value accuracy requirements k⩽1e(-6) .The location of each candidate solution, that is the parameter b of the GO model, is initialized to a random number between (0, 1). The stopping condition of the algorithm is to achieve the maximum number of iterations or the required accuracy of the fitness value. The algorithm was run initially 20 times, and the best results were taken in accordance with the principles in Section 2. The experimental results are shown in Table 1 and Table 2:

Table1 The results estimated by this paper’s method

data sets	parameter estimation results of GO model		the fitness value
	a	b	J
SYS1	136.7724	3.2207e-5	1.5619e-5
SS3	278.0001	3.6415e-5	4.7982e-6
CSR1	399.0705	4.8318e-5	5.1396e-6
CSR2	129.0110	1.0535e-4	6.1476e-5
CSR3	113.2761	1.6291e-4	9.4971e-6

Table2 The results estimated by paper[8]’s method

data sets	parameter estimation results of GO model		the fitness value
	a	b	J
SYS1	103.4267	7.8375e-5	119.9992
SS3	148.1259	9.7655e-5	933.5666
CSR1	269.0877	1.4576e-4	927.4841
CSR2	67.5188	1.8315e-4	293.2742
CSR3	75.4822	4.0557e-4	94.8255

It is known that the cumulative failure number n of SYS1, SS3, CSR1, CSR2 and CSR3 are respectively 136,278,397,129 and 104. It can be seen from Table 1 and Table 2 that the cumulative failure number a estimated by the new method proposed in this paper is more accurate than that in the literature [8], and the error between the estimated results and the actual results are all within 5%. So it proves that the new method is feasible and accurate.

4.2 Estimation and prediction

In this section, we focus on the combination of parameter estimation and model prediction. For the two methods, the parameters of the GO model are estimated by using the first half

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 4 Issue 2, February 2017
ISSN (Online) 2348 – 7968 | Impact Factor (2016) – 5.264
www.ijiset.com

failure of the five data sets, and then the estimated parameters are substituted into the function expression of the GO model to predict the occurrence moment of the second half failures. The algorithm was run initially 20 times, and the best results were taken in accordance with the principles in Section 2. The experimental results are shown in Table 3 and Table 4:

Table3 The results estimated by this paper's method

data sets	parameter estimation results of GO model		the fitness value
	a	b	J
SYS1	140.2974	4.3397e-5	1.6251e-6
SS3	282.3210	3.1519e-5	1.4819e-6
CSR1	381.0109	8.1514e-5	4.5243e-6
CSR2	127.4061	7.2078e-5	4.0338e-6
CSR3	125.5690	1.6228e-4	8.9294e-6

Table4 The results estimated by paper[8]'s method

data sets	parameter estimation results of GO model		the fitness value
	a	b	J
SYS1	54.2952	2.1069e-4	47.3875
SS3	108.7989	1.1513e-4	186.6797
CSR1	115.2995	4.3896e-4	528.2823
CSR2	38.3686	3.7827e-4	103.8701
CSR3	43.5738	0.001028	33.5226

It can be seen from Table 3 and Table 4 that , if only the first half of the data sets are used to estimate the parameters, the error between the estimated results using the new method in this paper and the actual values is still small. But the error between the results estimated by the method in literature[8] and the actual values is very large, especially for the 3-th and 4-th groups, the estimated value even do not reach 1/3 of the actual value . It is shown that the new method proposed in this paper can estimate more reasonable cumulative number of software failures in a real industrial project with a small amount of known failure data.

Next, the parameters in Table 3 and Table 4 are brought back to the formula (1). According to the formula, we predicted the occurrence moment of the second half failures of the five data sets. The comparisons between the predicted curve and the actual curve shown in Figure 1 to 5:

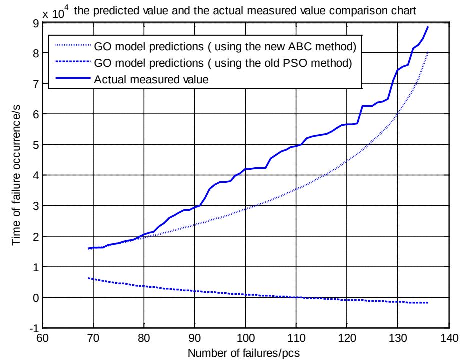

Fig.1 Occurrence moment of the second half failures of SYS1 dataset

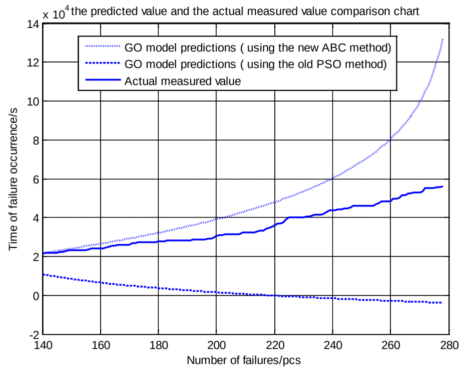

Fig.2 Occurrence moment of the second half failures of SS3 dataset

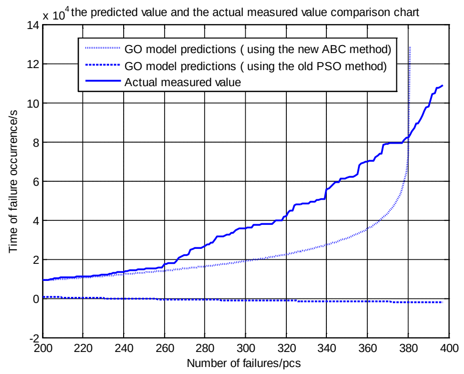

Fig.3 Occurrence moment of the second half failures of

CSR1 dataset

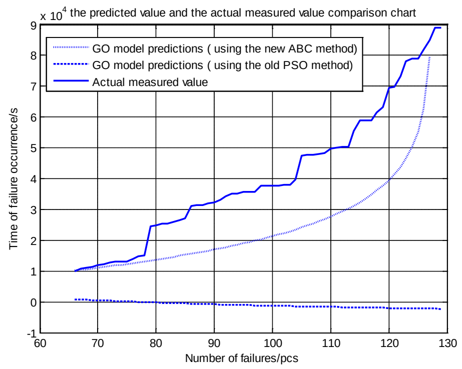

Fig.4 Occurrence moment of the second half failures of CSR2 dataset

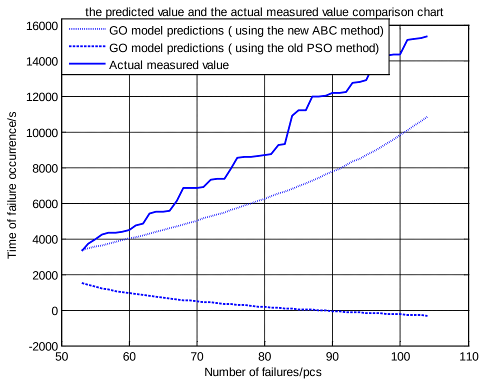

Fig.5 Occurrence moment of the second half failures of CSR3 dataset

From the Figure 1 to 5, we can see that the curves predicted by the new method proposed in this paper although has a certain error with the actual curve, but the basic trend is the same, that is, the failure time increases with the increase of the failure number. And the curve is exponentially distributed, the slope of the curve increases, showing that the software failure occurrence time interval is increasing, indicating that the reliability of the software is gradually improved, which is consistent with the factual situation that the reliability of software is improved with the discovery and modification of failures during software testing . However, the trend of curves predicted by the method in literature [8] is completely opposite to that of the actual curve, which is due to the very large error between the estimated value and the actual value. It can be shown that use the new method proposed in this paper with half of the failure datas to estimate the model parameter, and then predict the occurrence moment of the second half failures through the model is feasible and more accurate in reality.

5 Conclusion

The effect of parameter estimation of software reliability model will directly affect the accuracy of model prediction and must be paid attention to. In this paper, a new method for parameter estimation of software reliability model based on ABC is proposed on the basis of previous research experience. The feasibility and accuracy of the new method proposed in this paper are validated by the research, experiment and comparision of the GO model. The results show that the new proposed method is feasible and accurate for the parameter estimation of software reliability model. This paper also combined the parameter estimation with the model prediction , and results proved that the new method is also effective. It is worth noting that the new method proposed in this paper is not only limited to the GO model, but also applicable to other models.

References

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

[2]X.Y.Ma. Analysis of software reliability growth and its dynamic evaluation[D]. Harbin Institute of Technology Master's degree thesis，2011：22-43.

[3]W. He，N.X.Ling，M.C.Xu. Improvement of JM Software Reliability Model[J]. Statistics and Decision，2007，19：2-3.

[4]H.Garg, M.Rani. An approach for reliability analysis of industrial systems using PSO and IFS technique[J]. ISA Transactions, 2013, 52(6):701–710.

[5]T.K.Sharma, M.Pant, A.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.

[6]A.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, 2006:3071-3078.

[7]Z.C.Wang, Y.W.Zhao. Research on Quantum-behaved Particle Swarm Optimization Algorithm for Reliability Redundancy Optimization of Complex Systems[J]. Manufacturing Information Engineering of China，2007,36(11):72-75.

[8]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.

[9]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.

[10]IEEE Std 1633 TM-2008, IEEE recommended practice on software reliability[S]. New York: IEEE, 2008.

[11]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.

Min Mao(1991-）,Female，Master 's Degree，The main research direction is software reliability.

Zhen Li(1977-), Male, Associate Professor, Ph.D., The main research directions : system safety ,system reliability.

Yang Li(1978-), Male, Lecturer, Master 's Degree, The main research directions : System of Electronics and Information.

Hong Miao(1979-), Female, Lecturer, Ph.D., The main research directions : Management Information System, Data Analysis.

Min Wang(1974-), Female, Associate Professor, Master 's Degree, The main research directions : System of Electronics and Information.