ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20210807090754
Original scientific paper

A Study of Test Suite Reduction Based on Ant Lion Optimizer

Wei WEI, Weidong CHENG, Li YE, Shuibin XIA, Yueping WANG, Ying XING, Xingde WANG*

Abstract: The development of smart meters to Internet of Things is the infrastructure for the Internet to carry out user electricity management and enhance user experience with electricity. As the iteration of smart home service system based on smart meters continues to accelerate, the development process is becoming more and more demanding for software testing. Test suite reduction is one of the common methods to improve the efficiency of software testing. In this paper, we proposed an optimization algorithm based on the Ant Lion Optimizer applied to test suite reduction problem of smart IoT meters. The algorithm improved the traditional Ant Lion Optimizer by converting the smart IoT meter test suite reduction problem into a binary coverage problem and combining the Greedy Algorithm to obtain the optimal test case subset. The experimental results showed that the algorithm based on Ant Lion Optimizer performed better on the test suite reduction problems compared to similar algorithms.

Keywords: ant lion optimizer; smart meters; software testing; test suite reduction

1 INTRODUCTION

As the data interface of customer energy service, smart energy meter is the key to carry out value-added electric power data service. In order to adapt to the demand of smart grid Internet of Things (IoT) network, smart energy meters need to be equipped with energy use safety monitoring, energy use service guidance, IoT networking and edge computing capabilities. With smart meters as the core, we can build IoT of household smart devices, further enhance the granularity and diversity of meter collection data, and promote the construction of our world-class energy Internet enterprise.

According to statistics, the failure rate of power meter operation is about 0.004%, among which about 20% is caused by software. The performance of the smart meter software system is an important factor that affects the stability and reliability of smart meters. The reliability of the software quality is directly related to the vital interests of the grid company and the healthy development of smart grid [1]. In the software development process, the program of smart home service system is constantly iterated, which puts forward higher requirements for software testing. During software testing, we apply test suites to ensure correctness between program branches during system development [2]. As the size of the test suite continues to increase, the cost of software testing also continues to increase. According to statistics, software testing occupies more than 40% of the overall development costs [3].

Test suite reduction problem aims at reducing the number of executed test cases to a minimum. In order to effectively reduce the test suite, the reduced test suite must meet the same coverage as the original test suite when performing the reduction. Statement coverage, the percentage of statements covered during test execution, is a common test requirement in the real world [4].

The Ant Lion Optimizer (ALO) is a novel swarm intelligence algorithm proposed by imitating the intelligent behaviour of ant lions in nature when hunting ants. The algorithm optimizes the problem by implementing the interaction between ants and ant lions through simulation, and introduces the random wandering of ants to achieve global search. The diversity of the population is ensured by roulette strategy and elite strategy. The ant lions are equivalent to the solutions of the optimization problem, which achieve the update of the near-optimal solutions by hunting for highly adapted ants [5]. With the characteristics of diverse populations, strong performance in finding optimization, few adjustment parameters, and easy implementation, the ALO has been successfully applied to optimization problems such as power grid security [6], radar communication [7], and image matching [8], and has become a research hotspot in solving optimization problems.

Therefore, we propose a smart meter test suite reduction algorithm based on ALO. The algorithm utilizes the random movement of ants and the predatory behaviour of ant lions to discover the optimal subset of the test suite. Also, we analyse the effectiveness and robustness of ALO through experimental results.

The rest of this paper is organized as follows. In Section 2, we introduce the related work of this paper. The model of the test suite reduction problem is illustrated in section 3. In Section 4, we explain the design of the proposed algorithm. In Section 5, the experimental and empirical evaluations of the proposed method are conducted. Section 6 concludes this paper and highlights directions for future research.

2 RELATED WORK

In order to improve the efficiency of software testing, Harrol et al. [9] were the first to propose the concept of test suite reduction. Since then, many methods have been applied to the study of test suite reduction. Chvatal et al. [10] proposed the Greedy Algorithm (GA), a classical method applied to find the optimal subset of the test suite which continuously moved the test suite covering the most unsatisfied test requirements from the original set to the solution set until all requirements were covered. Chen et al. [11] proposed the Greedy Redundant Essential (GRE) algorithm, which first selected all the basic test suites and then applied the GA to the remaining test suite. These traditional methods found the optimal test suite in the current test requirements, so it is difficult to find the global optimum.

In order to obtain better results, scholars began to apply various intelligent optimization algorithms to the study of test suite reduction problem. Junlin Quan et al. [12] modelled the test suite reduction problem in the software testing process as a set coverage problem, proposed an algorithmic model based on GA to solve such problems. Jiaze Sun et al. [13] used a combination of Particle Swarm Optimization (PSO) and Stochastic Algorithm, proposed a two-dimensional Stochastic Particle Swarm Optimization to solve the test suite reduction problem. Shen Limin et al. [14]

combined GA and Ant Colony Optimization (ACO) and proposed a Genetic Ant Colony based fusion algorithm for solving the same problem, which drew the advantages of each in GA and ACO to achieve complementary advantages. Wenjing Liu et al. [15] proposed an algorithmic model for constructing Immune Algorithm (IA) to solve the complex set coverage optimization problem, and applied IA to generate the optimal subset of test suite that meets the coverage requirements within the test suite. Shuyan Wang et al. [16] designed variation factors and generated variants based on the source program features, and proposed a variation of analysis-based test suite reduction algorithm based on whether the specified variants can be detected by the test suite, which can effectively reduce the test suite. Wei Wei et al. [17] used the Firefly Algorithm (FA) to construct a binary cellular automaton model from two aspects of smart meter program branch coverage as well as execution cost, and applied it to smart meter test suite reduction.

3 PROBLEM MODEL

3.1 Basic Concept

The problem of simplifying the set of test suite based on smart meter programs can be summarized as the following definition.

1) Input

a) Test suite *T*: denoted as $T = \{t_1, t_2, \ldots, t_n\}$, is a set of test suite designed for the test requirements of the smart meter program, and *n* indicates the number of test suite.

b) Test requirements *R*: denoted as $R = \{r_1, r_2, \ldots, r_m\}$, is the number of paths or the branches, etc. of the smart meter program, and *m* indicates the number of test requirements.

c) Test relationship *S*: is the relationship between the test suite and the test requirements, which can be expressed as $S = \{(t, r) \mid t \text{ satisfies } r, t \in T \text{ and } r \in R\}$.

d) Execution cost *C*: denoted as $C = \{c_1, c_2, \ldots, c_m\}$, represents the overhead incurred when each test case is used for testing.

2) Objective

After the reduction, a subset *RS* is obtained that has minimal execution cost and can cover the same test requirements as the original test suite *T*.

3.2 Test Suite Reduction Model

The test suite reduction problem is a minimum set coverage problem. Let the test requirements be $R = \{r_1, r_2, \ldots, r_m\}$, the test suite be $T = \{t_1, t_2, \ldots, t_n\}$, $\exists T_1 \subset T$, where: T_1 can meet all test requirements *R*, $\forall T_x \subset T$, make the execution cost of the test suite contained in subset T_1 no greater than that of the test suite contained in T_x.

An example of test suite reduction is given. The program under test is the control program of the AD module. This example program is used to detect whether the alarm is normal when the external voltage is under voltage.

As shown in Tab. 1, the number of test suite *T* is 1680 and the number of test requirements *R* is 105, where ▪ represents the test relationship *S* and the number of execution cost *C* is 1680, corresponding to each test case.

Table 1 Example of test suite

Test suite *T*	Test requirements *R*						Execution cost *C*
	r_1	r_2	r_3	r_4	…	r_{105}	
t_1		▪	▪		…		c_1
t_2	▪			▪	…	▪	c_2
t_3					…	▪	c_{13}
…	...	…	…	…	…	…	…
t_{1680}				▪	…	▪	c_{1680}

The above problem can be converted into a mathematical problem: the test relationship *S* is replaced by a 0/1 matrix, where ▪ is replaced by 1 indicating that the test case satisfies the test requirements, and elsewhere by 0 indicating that the test case does not satisfy the test requirements. A vector indicates the execution cost *C*. Then Tab. 1 can be transformed into the following matrix.

$$\begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ \cdots \\ t_{1680} \end{array} \begin{array}{c} \begin{array}{cccccc} r_1 & r_2 & r_3 & r_4 & \cdots & r_{105} \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 1 & \cdots & 1 \end{bmatrix} \end{array} \begin{array}{c} \\ \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \cdots \\ c_{1680} \end{bmatrix} \end{array}.$$

The test relationship *S* can be represented as a 0/1 matrix with 1680 rows and 105 columns. The purpose is to find partial test suite from the test suite *T* so that it can cover the same test requirements as the full set and can guarantee the minimum execution cost. Let the vector *x* denote the result of the reduction, $x_j = 1$ indicates that the test case t_i is selected, $S_{ij} = 1$ denos that t_i can satisfy the test requirement t_j and the solution objective is expressed as:

$$\min f(x) = \vec{x} \cdot \vec{c} \tag{1}$$

$$\text{s.t.} \prod_{j=1}^{n} \left(\sum_{i=1}^{m} \left(x_i \cdot s_{ij} \right) \right) > 0 \tag{2}$$

where Eq. (1) is the reduction objective and Eq. (2) is the constraint.

4 ALO-BASED TEST SUITE REDUCTION

4.1 Introduction to the ALO algorithm

The ALO simulates the interaction process of ant lions hunting ants in a trap. Due to the fact that ants move randomly in nature while seeking food, the movement of ants is simulated by choosing a random walk. As is shown in Eq. (3).

$$\begin{aligned} X(t) = [0, &\operatorname{cumsum}(2r(t_1) - 1), \operatorname{cumsum}(2r(t_2) - 1), \ldots, \\ &\operatorname{cumsum}(2r(t_n) - 1) \end{aligned} \tag{3}$$

where cumsum is calculated as the cumulative sum, *n* is the maximal number of iterations, *t* denotes the number of steps of the random wander, and *r*(*t*) is passed through the random function defined as Eq. (4).

$$r\left(t\right)=\begin{cases}1, & \text{rand} > 0.5\\ 0, & \text{rand} \le 0.5\end{cases} \tag{4}$$

where rand is a random number that obeys uniform distribution within [0, 1].

The ants update their positions by wandering randomly in each step of the optimization. However, because each search space has a bound, Eq. (3) cannot be used directly for the update of ant positions. To keep the random wandering within the search space, it is qualified by using Eq. (5) as follows.

$$X_i^t=\frac{\left(X_i^t-a_i\right)\times\left(d_i^t-c_i^t\right)}{\left(b_i-a_i\right)}+c_i \tag{5}$$

where a_i is the small value of the i-th variable randomly wandering and b_i is the largest one. c_i^t is the smallest value of the i-th variable in the t-th iteration, and d_i^t is the largest one. It is necessary to have Eq. (5) in each iteration to ensure that random wandering occurs in the search space.

The random wandering of ants is impacted by traps as shown in Eq. (6) and Eq. (7).

$$c_i^t = Antlion_j^t + c^t \tag{6}$$

$$d_i^t = Antlion_j^t + d^t \tag{7}$$

where c^t is the smallest value of all variables in the t-th iteration, d^t is the largest value of all variables in the t-th iteration, and $Antlion_j^t$ is the position of the selected j-th ant lion in the t-th iteration.

The roulette strategy is used to select which specific ant is preyed upon by a particular ant. Each ant can only be preyed upon by one ant lion, and the higher the fitness, the higher the probability that the ant is captured by the ant lion. The phenomenon is simulated by Eq. (8) and Eq. (9).

$$c^t=\frac{c^t}{I} \tag{8}$$

$$d^t=\frac{d^t}{I} \tag{9}$$

where I is the ratio.

When the ant's fitness value is smaller than that of the ant lion, the ant lion is considered to have captured it, at which point the ant lion updates its position according to the ant's position, as shown in Eq. (10).

$$Antlion_j^t = Ant_i^t, f\left(Ant_i^t\right) > f\left(Antlion_j^t\right) \tag{10}$$

where Ant_j^t is the position of the i-th ant at the t-th iteration; f is the fitness function.

After each iteration, the ant lion with the best adaptation is selected as the elite ant lion. The position of the t-th ant at next iteration is identified by Eq. (11).

$$Ant_i^{t+1}=\frac{R_{\mathrm{A}}^t\left(l\right)+R_{\mathrm{E}}^t\left(l\right)}{2} \tag{11}$$

where $R_{\mathrm{A}}^t\left(l\right)$ is the value generated by an ant wandering randomly around an ant lion selected by roulette at the t-th iteration for the l-th step, and $R_{\mathrm{E}}^t\left(l\right)$ is the value generated by an ant wandering randomly around an elite ant lion at the t-th generation for the l-th step. l is any value within the number of random steps taken by the ant.

Algorithm ALO-based test suite reduction algorithm for smart meters

```
begin
    input:a test suite T
    input:the test requirements R
    input:the test relationship S
    input: the execution cost C
    output: the reduced test case suite RS
    max_iter: the maximum number of iterations
    pop: the number of ant lion population
    initialize algorithm parameters
    initialize ant lion populations and ant populations
    calculate the corresponding fitness in the ant lion population
    prioritize ant lion populations based on fitness
    take the first ant lion after sorted as the elite ant lion, and get the corresponding fitness
    for cur = 1 :max_iter:
        for i= 1 :pop:
            use roulette to select ant lion
            compute $R_{\mathrm{A}}^t(l)$ and $R_{\mathrm{E}}^t(l)$ by simulating ants wandering randomly
            update the location of ants
        end for
        calculate the fitness of the corresponding ant population
        update surrounding ant lion populations based on ants
        calculate the fitness of newly generated antlion populations
        update elite ant lions and get the appropriate fitness
    end for
    analyze the obtained experimental results
    return RS
end
```

4.2 Description of ALO-Based Test Case Reduction Algorithm

The ALO based smart meter test suite reduction algorithm proposed in this paper is shown in the following algorithm.

In the above algorithm, lines 1-5 represent the input and output, lines 6 and 7 mean the variable of the algorithm, lines 8 and 9 indicate the initialization of the ALO, the fitness corresponding to each ant lion is calculated in line 10, line 11 represents the rank of the ant lion population according to the fitness, line 12 introduces the elite ant lions and their corresponding fitness obtained, lines 13 to 23 represent the traversal of the ant lion population and the ant population, updating the ant population and the ant lion population for continuous iteration until the maximum number of iterations is reached, lines 24 and 25 process and visualize the results after the algorithm.

5 EXPERIMENTAL VERIFICATION

5.1 Experimental Environment

The experimental environment in this paper is a 64-bit operating system with x64-based processor i5-8250U, system version Windows 10, 1.6 GHz main frequency, 8.00G DDR4-2400 running memory, and programming language python 3.9.0. In the test suite reduction of each program under test, the ALO algorithm is applied to execute 20 times, and the results of 100 iterations of each ant lion population are analyzed and processed.

5.2 Test Data Set

The programs under test in this experiment are 5 modules in the smart meter program, and test suite of different sizes are prepared for different programs under test, which can achieve full branch coverage, and the specific information is shown in Tab. 2.

Table 2 Program information under test

Program Under Test	LOC	Number of branches	Test Suite Size	Description
ap_ad.c	167	105	1 680	detect AD analog-to-digital conversion
ap_CommProgram.c	355	78	1 326	program record flag clearing, and corresponding event recording
ap_recorder.c	292	96	1 632	responsible for reading load records
GuoWangPrepay.c	114	118	2 006	intelligent detection of plug-in users
Ap_ClockBatLowV.c	232	37	1 221	detect if the clock battery is under-voltage

5.3 Experimental Questions

The experiments in this paper aim to investigate the following questions.

1) Explore the ability of ALO in reducing execution cost

The reduction of execution cost is the most direct manifestation of improving the efficiency of regression testing. In this paper, the Execution Cost of the Representative Set (*ECRS*) is used to express the effect of simplification [18], and the formula is shown in (12).

$$ECRS = \sum_{i=RS} c_i \tag{12}$$

where c_i represents the cost of the i-th test case, and the smaller the value of *ECRS* after reduction, the better the reduction effect.

2) Explore the reduction effect of ALO in test suite of different sizes

3) Explore the reduction effect of ALO in the test suite of different smart meter programs under test

With the above problem as a benchmark, the following experiments are designed in this paper. We extract the test requirements for the smart meter programs. Five module code programs from the smart meter project are selected for the experiment, and test suites of different sizes are prepared for the different module code programs.

Table 3 Optimal number of test suite and corresponding ECRS values after reduction

Program Name	Algorithm	the Size of Test Case Suite									
		100		200		500		1000		full	
		Times	ECRS	Times	ECRS	Times	ECRS	Times	ECRS	Times	ECRS
ap_ad.c	ALO	8	170.95	**13**	158.7	**14**	147.05	**17**	139.5	**17**	137.85
	FA	**10**	171	4	164.6	8	149.15	4	143.9	5	138.6
	GRE	3	175.15	4	161.35	1	151.65	4	141	0	140
	greedy	0	100.15	2	169.05	0	154.45	0	151	0	141
ap_CommProgram.c	ALO	9	112.75	**13**	106	**17**	99.1	9	95.95	**16**	93
	FA	**11**	111.8	5	108.25	4	101.1	5	97.4	6	94.75
	GRE	4	114.7	9	106.85	2	103.05	**13**	95	0	95
	greedy	0	124.75	0	114.5	0	105.15	0	103	9	94
ap_recorder.c	ALO	14	148.5	**12**	141.7	**15**	129.7	**18**	124.25	**14**	122.8
	FA	6	149.8	2	144.6	2	133.7	4	126.5	7	124.3
	GRE	1	156.35	6	142.6	2	136.45	0	128	0	130
	greedy	0	166.2	3	148.05	2	136.3	0	127	0	136
GuoWangPrepay.c	ALO	**15**	201	**16**	186.2	**15**	172.15	**17**	163.05	**16**	160.25
	FA	4	205.05	1	193.05	1	175	5	166.05	7	161.85
	GRE	2	207.65	2	193.75	3	176.1	5	169.05	0	168
	greedy	3	211.75	2	196.45	2	180.45	0	174.65	0	167
Ap_ClockBatLowV.c	ALO	**12**	44.25	**19**	42.6	**16**	40.15	13	39.25	**17**	38.75
	FA	10	44.6	7	43.6	7	41.3	**16**	39.1	14	38.9
	GRE	5	46.65	3	44.95	3	43.45	1	40	0	40
	greedy	1	49.15	0	44.95	3	43.7	0	44	0	44

Note: The ECRS values are the average values after the algorithms are executed 20 times.

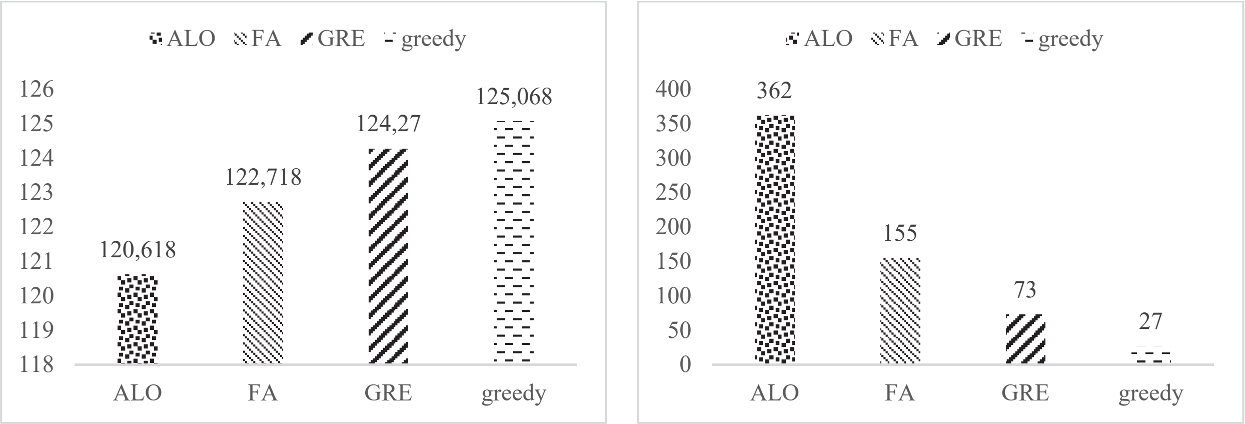
5.4 Experimental Investigation

In this paper, the experiments are performed on the test programs separately as follows: 100, 200, 500, 1000 and the full set are randomly selected in the original test suite to form a new test suite for reduction, while the same coverage as the full set is guaranteed. In this paper, we reproduce the test suite reduction algorithm based on the FA [17], GA and GRE to reduce the test suite of the selected five smart meter module programs, and count the number of times and *ECRS* values that perform best among the four algorithms. The experimental results are shown in Tab. 3, and the subsequent analyses are based on Tab. 3.

5.5 Analysis of Experimental Results

1) Question one

In this paper, by comparing the experimental results with FA, GRE and GA, the ECRS values of the four algorithms after the reduction are counted separately, as shown in Fig. 1a. The mean values of ECRS are obtained as 120.618, 122.718, 124.27 and 125.068. Meanwhile, the results of the ALO algorithm for test suite reduction are 1.71% less than the FA, 2.94% less than the GRE, and 3.56% less than the GA. The statistics from the optimal number is shown in Fig. 1b. The number of times the four algorithms reached optimality after reduction of the same set of test cases was 362, 155, 73 and 27. From both aspects, it can be seen that the ALO has better reduction effect.



(a) ECRS Comparison (b) Comparison of the optimal number of simplifications

Figure 1 Comparison of the optimal number of reduction and ECRS under the three algorithms

2) Question two

In this paper, by randomly generating test suites of 100, 200, 500, 1000 and the full set from the original test suites of the five tested programs for approximate simplification, the number of times that the ALO algorithm performs test suite reduction and gets the optimal subset is 58, 73, 77, 74 and 80, accounting for 49.15%, 59.34%, 65.81%, 56.49% and 62.5%, respectively. It can be seen that among the four algorithms, as the size of the test suite increases, the reduction effect of the ALO algorithm becomes better and better, and the specific data are shown in Tab. 4.

Table 4 Optimal number of times at different sizes

Algorithm	the Size of Test Case Suite				
	100	200	500	1000	full
ALO	**58**	**73**	**77**	**74**	**80**
FA	41	19	22	34	39
GRE	15	24	11	23	0
greedy	4	7	7	0	9

3) Question three

In this paper, the experimental results of five different tested programs are counted and the data are shown in Table 5. In the five programs under test, the optimal number of times of the ALO is 60%, 48.48%, 67.6%, 68.1%, and 52.38%, which indicates that the ALO-based optimization algorithm has good performance in different programs under test, and the ALO has good robustness in solving test suite reduction problems.

Table 5 Optimal number of times under different programs under test

Algorithm	the Size of Test Case Suite				
	ap_ad.c	ap_CommProgram.c	ap_recorder.c	GuoWangPrepay.c	Ap_ClockBatLowV.c
ALO	**69**	**64**	**73**	**79**	**77**
FA	31	31	21	18	54
GRE	12	28	9	12	12
greedy	2	9	5	7	4

6 CONCLUSION

The study of the test suite reduction problem has great significance to the software testing of smart meters. In this paper, the problem is modeled as a set coverage problem, which belongs to the classical problem of NP optimization. We improve the ALO to convert the smart IoT meter test suite reduction problem into a binary coverage problem, and combine GA to obtain the optimal subset of test suite.

The experimental results which are built on five smart meter programs under test show that the ALO-based optimization algorithm can complete the reduction of the test suite with high efficiency while maintaining the same coverage as the full set. Meanwhile, the experimental results for different sizes and different programs under test show that the ALO has good robustness.

We will try to use more optimization algorithms to solve the problems related to the approximate simplicity of smart meter test suites in the future. In addition, we will try to analyze and improve the performance improvement methods of ALO on small-scale test suites in the future.

Acknowledgement

The authors would like to thank the State Grid Corporation of China for its great support on this research. The paper is funded by the key technology research of smart IoT meter for energy service (52153220001Y) of State Grid Hubei Electric Power Co., Ltd.

7 REFERENCES

[1] Yuan, R., Ju, H., Wang, P., Song, X., Ding, H., & Liu, Y. (2018). Research on software testing of the smart meter based on black-box testing technology. *Electrical Measurement & Instrumentation*, *55*(22), 135-139.

[2] Ding, G., Zheng, Y., & Zhang, L. (2009). Study of Test Suite Minimization Based on Ant Colony Algorithm. *Computer Engineering*, *35*(6), 213-215, 218.

[3] Guo, W. & Guo, X. (2008). Study and Exploration on Automatic Test. *Computer Development & Applications*, *21*(12), 10-12+15.

[4] Harrold, & Jean, M. (2009). Reduce, reuse, recycle, recover: Techniques for improved regression testing. *IEEE International Conference on Software Maintenance*, 5-5.ž https://doi.org/10.1109/ICSM.2009.5306347

[5] Seyedali, M. (2015). The Ant Lion Optimizer. *Advances in Engineering Software*, *83*. https://doi.org/10.1016/j.advengsoft.2015.01.010

[6] Liu, Y., Wang, Y., Wang, X., & Han, Y. (2021). Optimization method of wind power cluster energy storage capacity allocation based on Ant-Lion algorithm. *Acta Energiae Solaris Sinica*, *42*(01), 431-437.

[7] Wang, X. & Ruan, J. (2020). Waveform Design for Radar Communication Integration Based on the Multi-objective Ant Lion Optimization. *Radio Engineering*, *50*(08), 652-655.

[8] Zhang, H., Gao, Z., Zhang, X., & Shi, K. (2019). Image Matching Method Combining Hybrid Simulated Annealing and Antlion Optimizer. *Computer Science*, *46*(06), 328-333.

[9] Harrold, M, Gupta, J., & Soffa, R. (1993). A methodology forcontrolling the size of a test suite. *Acm Transactions on Software Engineering & Methodology*, *2*(3), 270-285. https://doi.org/10.1145/152388.152391

[10] Chvatal, V. (1979). A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, *4*(3), 233-235. https://doi.org/10.1287/moor.4.3.233

[11] Chen, T. Y. & Lau, M. F. (1998). A new heuristic for test suite reduction. *Information and Software Technology*, *40*(5-6), 347-354. https://doi.org/10.1016/S0950-5849(98)00050-0

[12] Quan, J. & Lu, L. (2009). Research test case suite minimization based on genetic algorithm. *Computer Engineering and Applications*, *45*(19), 58-61+80.

[13] Sun, J., Wang, S., & Cao, X. (2009). Improved PSO Algorithm for Test Suite Minimization Problem. *Computer Engineering*, *35*(15), 201-202+205.

[14] Shen, L. & Gao, J. (2012). Test Case Minimization Research Based on Combination Algorithm of Genetic and Ant Colony. *Computer Engineering*, *38*(16), 57-60+64.

[15] Liu, W., Xing, Y., Han, H., & Gong, Y. (2017). Efficient regression test suite reduction mechanism based on artificial immune algorithm. *Journal of Southeast University (Natural Science Edition)*, *47*(S1), 170-175.

[16] Wang, S., Chen, P., & Sun, J. (2017). Reduction method of test suites based on mutation analysis. *Journal of Computer Applications*, *37*(12), 3592-3596.

[17] Wei, W., Su, J., Ye, L., Li, F., & Wang, X. (2021). Test Case Suite Reduction Based on an Intelligent Optimization Algorithm. *Journal of CAEIT*, *16*(02), 111-118+126.

[18] Shuai, W., Shaukat, A., & Arnaud, G. (2015). Cost-effective test suite minimization in product lines using search techniques. *The Journal of Systems & Software*, *103*. https://doi.org/10.1016/j.jss.2014.08.024

Contact information:

Wei WEI, PhD
Marketing Service Center, State Grid Hubei Electric Power Co., Ltd.,
488 Qingwang Road, East Lake Scenic Area, Wuhan, Hubei Province, China
E-mail: wweihn@126.com

Weidong CHENG, Engineer
Holley Technology Co., Ltd.
181 Wuchang Avenue, Wuchang Street, Yuhang District, Hangzhou,
Zhejiang Province, China
E-mail: weidong.cheng@holley.cn

Li YE, MA. Eng.
Marketing Service Center, State Grid Hubei Electric Power Co., Ltd.,
488 Qingwang Road, East Lake Scenic Area, Wuhan, Hubei Province, China
E-mail: 21515996@qq.com

Shuibin XIA, MA. Eng.
Marketing Service Center, State Grid Hubei Electric Power Co., Ltd.,
488 Qingwang Road, East Lake Scenic Area, Wuhan, Hubei Province, China
E-mail: 2261900807@qq.com

Yueping WANG, MA**.** Eng.
Holley Technology Co., Ltd.,
181 Wuchang Avenue, Wuchang Street, Yuhang District, Hangzhou,
Zhejiang Province, China
E-mail: yueping.wang@holley.cn

Ying XING, Associate Professor, PhD
Beijing University of Posts and Telecommunications,
10 Xitucheng Road, Haidian District, Beijing, China
E-mail: xingying@bupt.edu.cn

Xingde WANG, postgraguate
(Corresponding author)
Beijing University of Posts and Telecommunications,
10 Xitucheng Road, Haidian District, Beijing, China
E-mail: wxd_sanders@163.com