MC/DC Test Data Generation Algorithm Based on Whale Genetic Algorithm

LIU Huiying, LIU Ziyang, YAN Minghui

(*North China Institute of Aerospace Engineering*, *Langfang* 065000)

Abstract: The automatic generation of test data is a key step in realizing automated testing. Most automated testing tools for unit testing only provide test case execution drivers and cannot generate test data that meets coverage requirements. This paper presents an improved Whale Genetic Algorithm for generating test data required for unit testing MC/DC coverage. The proposed algorithm introduces an elite retention strategy to avoid the genetic algorithm from falling into iterative degradation. At the same time, the mutation threshold of the whale algorithm is introduced to balance the global exploration and local search capabilities of the genetic algorithm. The threshold is dynamically adjusted according to the diversity and evolution stage of current population, which positively guides the evolution of the population. Finally, an improved crossover strategy is proposed to accelerate the convergence of the algorithm. The improved whale genetic algorithm is compared with genetic algorithm, whale algorithm and particle swarm algorithm on two benchmark programs. The results show that the proposed algorithm is faster for test data generation than comparison methods and can provide better coverage with fewer evaluations, and has great advantages in generating test data.

Keywords: Test Data Generation, MC/DC, Whale Genetic Algorithm, Mutation Threshold

1 Introduction

Test data generation is a key step to realize software test automation. The existing automatic testing tools for unit testing, such as C++test and Testbed, which are difficult to generate test data to meet the software requirements in key areas. Currently, most unit test data still rely on manual builds by testers. However, for complex programs, manually constructing test data requires testers to fully understand the structure and function of the program, so it is difficult to construct data that meets the requirements of the MC/DC standard in a short time. This paper studies the automatic generation algorithm of MC/DC test data based on intelligent optimization algorithm.

Under the guidance of the fitness function, the intelligent optimization algorithm transforms the generation process of test data into the process of searching for the optimal solution in the problem space[1]. Through the heuristic search algorithm, the test data that meets the test requirements can be obtained quickly. Common intelligent optimization algorithms include genetic algorithm, particle swarm optimization algorithm, simulated annealing algorithm, etc. Among them, genetic algorithm is the most widely used.

Solving the problem of test data auto-generation using genetic algorithm has always been a research hotspot. Huang aimed at the problem of premature convergence of traditional genetic algorithm and low population diversity in the later stage of iteration, used the reverse learning strategy to initialize the population, combined with the layer proximity to improve the evaluation method of individual fitness, and optimized the crossover and mutation operation by used chaotic sequence[2]. Aimed at the shortcomings of genetic algorithm, Sun introduced simulated annealing mechanism to judge whether to receive bad individuals with a certain annealing probability according to the coverage in the mutation stage[3]. Zhang proposed the improved genetic ant colony algorithm, which used the genetic algorithm to initialize the pheromone of the improved ant colony algorithm and improve the coverage of the target path[4]. Yuan proposed a generation method based on hybrid genetic algorithm to improve the adaptive crossover and mutation operators by adjusting factors to improve the local search ability of the algorithm[5].

Using genetic algorithm to solve test data generation problems, the above studies mostly focus on the improvement of the fitness function or on the improvement of the algorithm itself, ignored the relationship between the algorithm and the problem to be solved. Therefore, this paper presents an improved whale genetic algorithm, which can improve the convergence speed of the algorithm by dynamically adjusting the mutation threshold, and balance the relationship between population and individual evolution by improving the crossover strategy. It can be used in MC/DC test data auto-generation to improve the efficiency of test data generation.

2 Related Work

2.1 Modified Condition/Decision Coverage

Test data generation methods can be roughly divided into two categories: function-oriented test data generation and structure-oriented test data generation. This paper mainly studies the generation of structure-oriented test data, and uses the MC/DC as the coverage criterion.

In 1992 RTCA and EUROCAE jointly issued a document for the guidance of software development for airborne equipment. The MC/DC criterion was proposed by Boeing and is the Class A coverage standard for DO-178B[6]. As a recognized development guide, it is widely used in software verification and testing, especially in some safety-critical products such as aerospace software. The MC/DC guidelines require that the following conditions must be met[7-9].

(1) Enter each entry in the program at least once, and exit each exit at least once.

(2) The coverage of the judgment should be as wide as possible.

(3) All possible situations for a single condition in the decision occur at least once.

(4) Each condition in each judgment must independently affect the judgment result at least once.

Taking the expression A&&(B||C) as an example, its truth table is as shown in Table 1.

Table 1 Truth Table

Number	Condition A	Condition B	Condition C	Determination A&&(B||C)
1	F	F	F	F
2	F	F	T	F
3	F	T	F	F
4	F	T	T	F
5	T	F	F	F
6	T	F	T	T
7	T	T	F	T
8	T	T	T	T

It can be obtained from Table 1 that for condition A, its independent influence pairs are (2,6), (3,7), (3,8), the independent influence pair of condition B is (5,7), and the independent influence pair of condition C is (5,6). Finally, a union is obtained for the independent impact pairs of each condition, resulting in the smallest set of test cases {2,5,6,7} or {3,5,6,7} that satisfy MC/DC.

The Chilenski principle states, for the determination of each condition, the number of use cases generated by the test is [n+1,2*n], that is, the minimum number of use cases is 1 more than the number of conditions, and the maximum value is 2 times the number of conditions. In engineering application, the number of conditions with twice the number of test cases can be realized and accepted. Compared with multi-condition coverage criteria, MC/DC coverage has stronger error detection ability than other coverage criteria[10].

Using MC/DC criteria as the coverage criteria, the corresponding control flow chart is obtained by static analysis of the program under test, and the target path set meeting MC/DC criteria is generated according to the control flow chart. The test data generated to cover the target path is used as the end condition of the intelligent optimization algorithm, and the fitness function to meet the test data generation problem is constructed. It is inserted into the specific location of the program under test by the instrumentation technology. The test data is analyzed according to the fitness value returned by the program. The optimal test data is found by the limited iteration algorithm execution.

2.2 Genetic Algorithm

When the genetic algorithm is searching for optimization, it generates a certain number of individuals by initializing the population, evaluates the fitness of each, selects individuals from the population for crossover to generate offspring, mutates the offspring individuals, and repeats the above steps until the iteration is terminated. The basic principle is as shown in Fig.1.

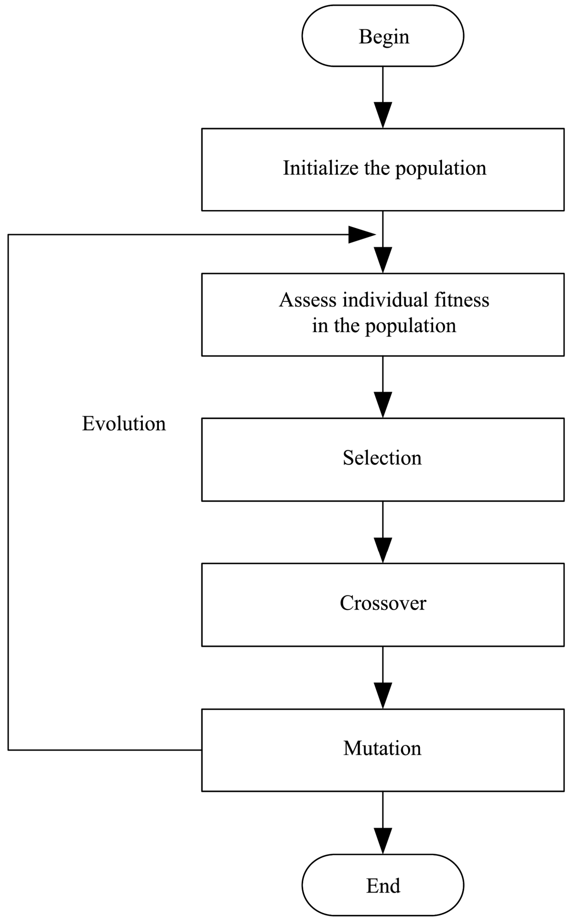


Fig.1 Basic Principles of Genetic Algorithms

(1) Selection

Select several individuals from the population with a certain probability. Generally, the selection process is a process of survival of the fittest based on fitness. Individuals with high fitness values will be judged as the next generation retained individuals, and individuals with low fitness values will be judged as abandoned individuals of the generation. The initial motivation of selection is to evaluate the fitness of a single individual, and to use an appropriate selection strategy to screen the candidate set in the population. It can be said that selection promotes the solution process of the target problem.

(2) Crossover

Crossover refers to the patchwork and recombination of chromosomes based on their internal structural elements, through the exchange of their internal elements between different individuals, thereby expanding the search capacity of the population and improving the reliability of the algorithm.

(3) Mutation

Mutation refers to probabilistically changing chromosomal genes to generate new genetic individuals.

2.3 Whale Optimization Algorithm

When the whale group hunts, on the one hand, it drives and rounds up its prey, and on the other hand, it spit out spiral-shaped bubbles for bubble attack. The basic principle is as shown in Fig.2.

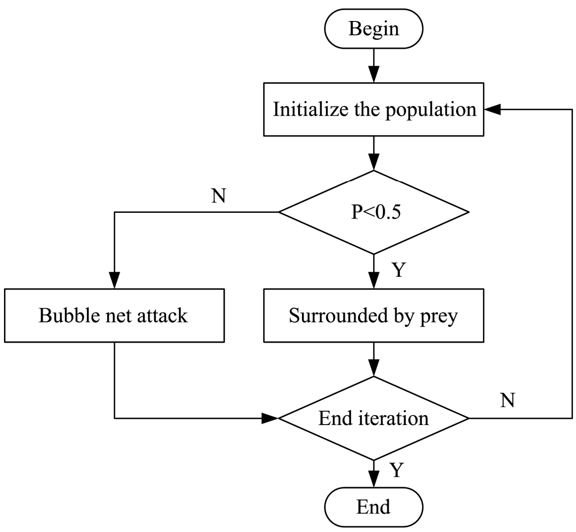


Fig.2 Basic Principles of Whale Algorithm

(1) Surround Prey

When surrounded by prey, whales choose to swim toward the optimally positioned whale or toward a random whale. See formulas 1 and 2 for updating towards a random whale swimming position.

$$X(t+1) = X_{rand} - A \times D \tag{1}$$

$$D = | CX_{rand} - X(t) | \tag{2}$$

where X_{rand} is the randomly selected whale position, A and C represent coefficient vectors, each dimension of A is a random number evenly distributed between $(-a, a)$, the initial value of a is 2, and it linearly decreases to 0 with the increase of the number of iterations. A random number of C uniformly distributed between $(0,2)$, $||$ represents the absolute value of the number.

$$A = 2 \cdot \alpha \cdot r_1 - \alpha \tag{3}$$

$$C = 2 \cdot r_2 \tag{4}$$

where r_1 and r_2 are random numbers between 0 and 1.

The whale moves towards the optimal position of the whale, see 5 and 6 for the updated formulas.

$$X(t+1) = X_{best} - A \cdot D \tag{5}$$

$$D = | CX_{best} - X(t) | \tag{6}$$

where X_{best} is the position of the current optimal whale.

When the whale chooses to swim towards the optimal individual and when to choose a random individual as the target is determined by the value of A, When $|A<1|$, the whale chooses to swim toward the optimal individual, and when $|A \geqslant 1|$, the whale chooses to swim toward a random individual.

(2) Bubble Web Attack

Whales randomly choose whether to hunt with bubble nets after finding their prey. Generates a random value, when the probability value is greater than or equal to 0.5, the whale will spiral into the prey.

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X_{best} \tag{7}$$

$$D' = | X_{best} - X(t) | \tag{8}$$

where D' is the distance from the whale to the current optimal position, b is the spiral shape constant, usually the value is 1, and l is a random number uniformly distributed between [-1, 1].

Before each action, the whale decides, based on probability P, whether to surround the prey or use a bubble net to drive it away. When $P > 0.5$, the whales use the bubble net to repel the prey and vice versa to surround the prey.

2.4 Construction of Fitness Function

The fitness function is the key element of algorithm optimization, and it is the basis for evaluating the pros and cons of the data generated by the algorithm. In the used of intelligent algorithms to solve the problem of test data generation, the design of the fitness function needs to be able to reflect the approximation of the current test data and the target test data. This approximation is expressed using branch distances. The branch distance evaluates the closeness of the test data to the target coverage branch, and its calculation is realized by branch predicates.

Table 2 Truth Table

Expression	True	False
a==b	abs(a-b)	a==b? k:0
a!=b	a!=b? k:0	a!=b? abs(a-b):0
a<b	a<b? 0:a-b+k	a<b? a-b+k:0
a<=b	a<=b? 0:a-b	a<=b? a-b:0
a>b	a>b? 0:a-b+k	a>b? a-b+k:0
a>=b	a>=b? 0:a-b	a>=b? a-b:0
a||b	min[fit(a), fit(b)]	fit(a)+fit(b)
a&&b	fit(a)+fit(b)	max[fit(a),fit(b)]

This paper focuses on the test data generation of MC/DC path. In an MC/DC path, the path fitness value should be the accumulation of branch fitness functions of all decision conditions of the path. If there are n branches on the target path of the tested program, and the expressions of their branch functions are $f_1, f_2, \ldots f_n$, the fitness function is expressed in formula 9.

$$F = f_1 + f_2 + \ldots + f_n \tag{9}$$

Through a certain mathematical transformation, it is transformed into the problem of finding the maximum value of fitness function. When the fitness function reaches the maximum value, it means that the specified MC/DC path is covered.

$$F = \left(\frac{100}{f_1+1} + \frac{100}{f_2+1} + \ldots + \frac{100}{f_n+1}\right) \div n \tag{10}$$

where f is the branch fitness and n is the number of branches.

3 Whale Genetic Algorithm

3.1 Elite Retention Strategies

In the selection operation of the genetic algorithm, the roulette selection method cannot guarantee that the optimal solution can be selected to the next generation. Therefore, this paper adds an elite retention strategy on the basis of roulette. The individual with the worst fitness value in the current population is replaced by the best individual that has appeared so far, and then the next generation of individuals is selected by roulette selection. The probability of an individual being inherited to the next generation is shown in formula 11. Compared with other individuals in the population, the optimal individual retains a higher fitness value in several previous optimization iterations. A combination of elite retention and roulette selection can effectively prevent population degradation.

$$P(x_i) = \frac{f(x_i)}{\sum_{i=1}^{n} f(x_i)} \tag{11}$$

where $f(x_i)$ is the fitness value of each individual, and n is the population size.

3.2 Adaptive Mutation Threshold

Each individual of the standard genetic algorithm will go through two stages of crossover and mutation. In this paper, the probability threshold in the whale algorithm is introduced, and a strategy for dynamically updating the mutation threshold is proposed according to the current individual fitness. When the fitness value of the individual is greater than or equal to the average fitness value of the population, the mutation threshold is fixed at 0.5, that is, the evolution strategy is selected with equal probability. Instead, the evolutionary direction is chosen based on population diversity. When the diversity of the population is low, the mutation threshold is increased, so that individuals have more possibilities to choose mutation operations. When the diversity of the population is high, the mutation threshold is reduced to allow individuals to evolve in the direction of crossover operation. At different stages of population evolution, individuals can choose an evolution method that is more suitable for the current population to speed up the convergence of the algorithm.

For the measurement of population diversity, the Euclidean Distance between individuals is added, and the Euclidean Distance between two individuals is shown in formula 12.

$$d_{ij} = \sqrt[2]{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{12}$$

where i and j range from population size n, and $i \neq j$, x_i represent the i-th genotype array, and x_{ik} represents the i-th variable stored in the t-th genotype array.

The Euclidean Distance calculation of the population is shown in formula 13.

$$D = \sum_{k=1}^{n} \sum_{j=1}^{n} d_{ij} \tag{13}$$

The average fitness of individuals in the population is expressed in formula 14.

$$f_{avg} = \frac{1}{n} \sum_{i=1}^{n} f_i \tag{14}$$

At this time, the fitness value diversity of the population is shown in formula 15.

$$F = \frac{1}{n} \sum_{\mathrm{i}=1}^{n} (f_i - f_{avg})^2 \tag{15}$$

where f_i represents the fitness value of the i-th individual in the population.

The calculation of population diversity in the new whale genetic algorithm is shown in formula 16.

$$V_{PD} = \rho \cdot D + (1-\rho) \cdot F \tag{16}$$

where $\rho (0 \leqslant \rho \leqslant 1)$ is the weight parameter, this article sets it to 0.5.

Finally, the setting of the probability threshold is shown in formula 17.

$$P = \begin{cases} P_0(1 - normalize(V_{PD})), & f_i < f_{avg} \\ P_0 & , \quad f_{\mathrm{i}} \geqslant f_{avg} \end{cases} \tag{17}$$

where P_0 is set to 0.5, $normalize()$ is the normalization function.

3.3 Coefficient Vector

The coefficient vector in the whale algorithm is introduced in the crossover stage to balance the global development and local exploration capabilities of the algorithm. When the absolute value of the coefficient vector is greater than 1, the current individual is crossed with a random individual to enhance the global search ability of the algorithm. When it is less than 1, it crosses with the current optimal individual. See formula 18 for calculating the value of A.

$$A = 2\alpha \cdot \mathrm{r} - \alpha \tag{18}$$

where r is a random vector between [0,1], and the vector $\alpha \in [2,0]$ which decreases linearly with the increase of the number of iterations, as shown in formula 19.

$$\alpha = 2\text{-}2 \cdot (t \,/\, \mathrm{MAXGENS}) \tag{19}$$

In formula 17, t is the current number of iterations, and MAXGENS is the maximum number of iterations. It can be seen from equations 16 and 17 that the value of the coefficient vector A decreases linearly with the value of α . Due to the linear relationship, the balance between algorithm search and development cannot be fully represented. In this paper, the adaptive adjustment of the convergence factor is adopted, as shown in formula 20.

$$\alpha = -\cos(\pi \cdot \frac{t}{MAXGENS} + \pi) + 1 \tag{20}$$

3.4 Process of Whale Genetic Algorithm

The Whale Genetic Algorithm determines whether individuals are crossed or mutated by the adaptive mutation threshold. The algorithm generates a random number P, which value between 0 and 1. When P is greater than the mutation threshold, the crossover operation is performed on the individuals in the population, and the number of iterations is obtained. The coefficient vector A is obtained. When the absolute value of A is greater than 1, a random individual is selected for crossover, otherwise, it is crossed with the current global optimal individual. When P is less than the mutation threshold, individuals in the population are mutated. The pseudo-code of the improved whale genetic algorithm is shown in Fig.3.

```
Algorithm 1: WGA
initialization F(0),temp;
while t≤T do
    for i=1 to M do
        Evaluate fitness of F(t);
    end
    for i=1 to M do
        Select operation to F(t);
    end
    P=rand();
    if P≤temp then
        for i=1 to M do
            Mutation operation to F(t);
        end
    else
        if |A|<1 then
            for i=1 to M/2 do
                Crossover operation to F(t) with rand;
            end
        else
            for i=1 to M/2 do
                Crossover operation to F(t) with best;
            end
        end
    end
    for i=1 to M do
        F(t+1)=F(t);
    end
    t=t+1;
end
```

Fig.3 Improved Whale Genetic Algorithm Pseudocode

4 Experiment

This paper selects two programs with different scales as the tested program to evaluate the performance of the proposed method. At the same time, it is compared with Genetic algorithm, Whale algorithm and Particle Swarm Optimization algorithm, which are most widely used in test data generation at this stage.

4.1 Benchmark Program

The first benchmark procedure for the experiments is the classic triangle classification procedure. The program takes the input three integers a, b, and c as the three sides of the triangle, and judges the type of triangle formed by the set of side values according to the size relationship of the three sides. The test program has multiple branch paths, and even if the input range is large, only a small number of input combinations can satisfy a specific branch. The control flow chart of the triangle classification program is shown in Fig.4.

By running the modified whale genetic algorithm, test data is generated for each path in Table 3, and the generated use case data is as shown in Table 4.

Table 3 Logic Path Table of Triangle Classification Program

Path	C1	C2	P1	P2	P3	Execute Branch	Result
Path1	-	-	F	-	-	a-c	Non-triangular
Path2	F1	T2	T	T	F	a-b-d-g	Isosceles Triangle
Path3	T1	F2	T	T	T	a-b-d-g	Isosceles Triangle
Path4	T1	F2	T	T	T	a-b-d-f	Equilateral Triangle
Path5	F1	F2	T	F	-	a-b-e	Ordinary Triangle

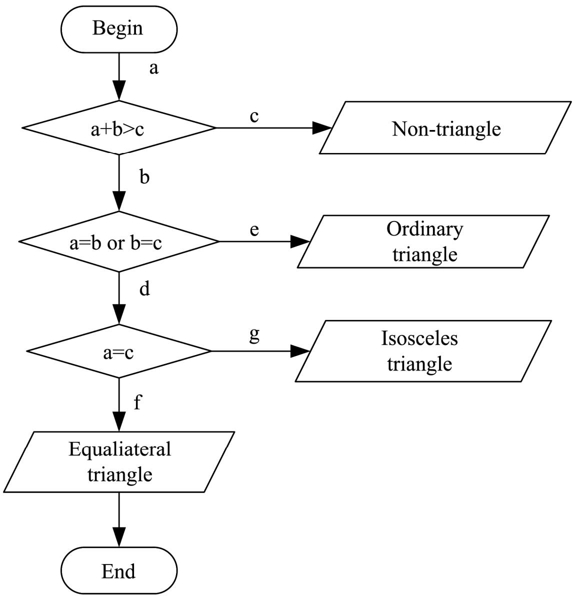


Fig.4 Triangle Classifier Branch Path Numbering Diagram

In addition to the triangle classification program, this paper also selects the program that calculates the day as the day of the year as the second benchmark test program. Compared with the triangle classification program MC/DC coverage requirements, this program is more complex and has a certain representativeness. The branch path numbers of the program to be tested are shown in Fig.5.

For the branch path shown in Fig.5, it can be obtained by generating the control flow chart and the MC/DC test case generation method. The logic coverage path of the second benchmark program is as shown in Table 5.

By running the improved whale genetic algorithm, test case data is generated for each coverage path in Table 5, and the generated use case data is shown in Table 6.

Table 4 Triangle Classification Program Test Case

Number	a	b	c	Cover Path
1	15	6	5	Path1
2	20	22	22	Path2
3	57	96	57	Path3
4	95	95	95	Path4
5	54	79	43	Path5

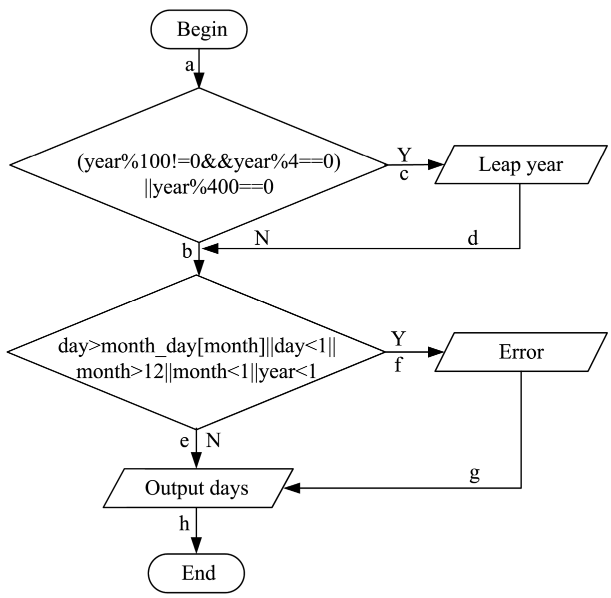


Fig.5 Calculate the Day as the Program Path Number of the Current Year

Table 5 Calculate the Day as the Day of the Year in the Program Logic Path

Path	C1	C2	C3	C4	C5	C6	C7	C8	P1	P2	Execute Branch
Path1	T1	F2	F3	T4	F5	F6	F7	F8	F	T	a-b-f-g-h
Path2	F1	T2	T3	F4	T5	F6	F7	F8	F	T	a-c-d-f-g-h
Path3	T1	F2	F3	F4	F5	T6	F7	F8	F	T	a-b-f-g-h
Path4	T1	F2	F3	F4	F5	F6	T7	F8	F	T	a-b-f-g-h
Path5	T1	F2	F3	F4	F5	F6	F7	T8	F	T	a-b-f-g-h
Path6	T1	T2	F3	F4	F5	F6	F7	F8	F	F	a-c-d-e-h

Table 6 Calculate the Day as the Day of the Current Year Program Test Case

Number	Year	Month	Day	Cover Path
1	-1	5	32	Path1
2	400	9	0	Path2
3	-1	21	7	Path3
4	-1	0	9	Path4
5	-1	4	20	Path5
6	8	6	17	Path6

4.2 Parameter Settings

In the improved whale genetic algorithm proposed in this paper, the settings of crossover probability and mutation probability directly affect the optimization result of the algorithm. Taking the first benchmark program as the program to be tested, the ablation experiment is performed on the crossover operator first, and the number of iterations and time are recorded. Then change the size of the mutation operator, and record the number and time of iterations performed by the algorithm. The population size was set to 100, the maximum number of iterations of the algorithm was set to 60, and the experiment was performed 50 times. The experimental results obtained by setting different crossover probabilities and mutation probabilities are as shown in Table 7.

From the results in Table 7, it can be seen that when the set mutation probability is small or large, the performance of the algorithm will not achieve the expected good results. If the mutation probability is too small, the probability of new genes appearing in the population is extremely low, which makes it difficult for the algorithm to jump out of the local optimum, so it takes a lot of time; when the mutation probability is too large, a large number of new genes appear in the population, which is not conducive to the convergence of the algorithm. The time spent in the process will increase. Choosing the appropriate crossover probability and mutation probability can make the algorithm have good optimization ability. Therefore, in the experiment below, set P_{c1} as to 0.90, P_{c2} as to 0.85, P_{m1} as to 0.10 and P_{m2} as to 0.096.

Table 7 Comparison of Genetic Operator Results

Parameter	Numerical Settings	Average Number of Iterations	Average Iteration Time (ms)
Crossover Probability	$P_{c1}=0.90, P_{c2}=0.85$	44	78
	$P_{c1}=0.85, P_{c2}=0.80$	47	84
	$P_{c1}=0.80, P_{c2}=0.75$	46	82
Mutation Probability	$P_{m1}=0.30, P_{m2}=0.10$	49	88
	$P_{m1}=0.10, P_{m2}=0.096$	44	78
	$P_{m1}=0.08, P_{m2}=0.096$	46	81

4.3 Experiment and Result

Taking the first benchmark program as the program to be tested, run the improved whale genetic algorithm (WGA) in this paper and compare it with genetic algorithm, whale algorithm and particle swarm optimization algorithm. We set the range of randomly generated boundary value to [1,100] and the population size is 100, the maximum number of iterations is 100, and the experiment is executed 50 times. The remaining settings of each algorithm are as shown in Table 8.

Comparing the average results of 50 experiments of genetic algorithm, whale algorithm, particle swarm algorithm and improved whale genetic algorithm, the comparison of the number of iterations generated by the optimal solution and the coverage of sentences, branches and MC/DC are as shown in Table 9.

As shown in Table 9, the average number of iterations of the improved whale genetic algorithm to generate the optimal solution is 41.34, which avoids the problem of low convergence accuracy to a certain extent compared with other algorithms. The improved whale genetic algorithm takes advantage of the strong local search ability of the whale algorithm to effectively avoid the defect of low solution quality when the genetic algorithm is large in size. It can be seen from the coverage results that the improved whale genetic algorithm has the highest data quality and the strongest algorithm stability. It should be noted here that during the operation of the algorithm, it cannot be guaranteed that the test data that satisfies the specified coverage path can be found at the end of each iteration, so the coverage rate is lower than 100%.

For the second benchmark program, set the range of year data generated by each algorithm as [-1,2022], the range of month data generated as [0,20], the range of date data generated as [0,40], the population size as 100, the maximum number of iterations as 60, and the experiment is executed 50 times. Other parameter settings of each algorithm are shown in Table 7.

Compare the average results of genetic algorithm, whale algorithm, particle swarm optimization algorithm and improved whale genetic algorithm in 50 experiments to produce the number of iterations when finding optimal solution. The results of statement, branch and MC/DC coverage are shown in Table 10.

Table 8 Parameters Setting of Algorithms

Parameter Setting	GA	WOA	PSO	WGA
Crossover Probability	0.8	-	-	$P_{c1}=0.9, P_{c2}=0.85$
Mutation Probability	0.15	-	-	$P_{m1}=0.1, P_{m2}=0.096$
probability Threshold	-	0.5	-	-
Maximum Speed	-	-	5	-
Minimum Speed	-	-	-5	-

Table 9 Comparison of Triangle Judgment Program Results

Algorithm Comparison	GA	WOA	PSO	WGA
Iterations of Optimal Solution	56.22	47.34	57.67	**41.34**
Statement Coverage	77.364	84.364	88.748	**93.372**
Branch Coverage	73.998	83.998	84.874	**89.292**
MC/DC Coverage	52	70	69	**80**

Table 10 Calculate the Comparison of Program Results on the Day of the Year

Algorithm Comparison	GA	WOA	PSO	WGA
Iterations of Optimal Solution	96.3	89.52	92.44	**84.85**
Statement Coverage	78.82	81.83	85.5	**92.92**
Branch Coverage	72.95	77.38	80.03	**86**
MC/DC Coverage	45	53	59	**66**

As shown in Table 10, the average iterations of the optimal solution generated by running the improved whale genetic algorithm is 84.85, which is lower than other algorithms. The average sentence coverage of the test data generated by the improved whale genetic algorithm is 92.92%, and the average branch coverage can reach 86%, which is better than the comparison algorithm. The average coverage of MC/DC is increased from 45% of genetic algorithm to 66%, which is also higher than whale algorithm and particle swarm optimization algorithm. The experimental results show that the test data generated by the algorithm in this paper is higher than the comparison algorithm in terms of statement, branch coverage and MC/DC coverage, and the number of iterations is less than the comparison algorithm. Therefore, this algorithm has more advantages in solving efficiency and accuracy.

5 Conclusion

Heuristic search algorithms have been widely used in the field of automatic testing. In this paper, an improved whale genetic algorithm is proposed to generate MC/DC test data. For the slow convergence speed and premature phenomenon of standard genetic algorithm, the improved algorithm combines three improved strategies: Elite retention strategy, adaptive adjustment of mutation threshold and dynamic updating of coefficient vector to effectively improve the global search ability of genetic algorithm. The algorithm is verified by the triangle classification program and the program calculating the day of the year. Compared with the standard genetic algorithm, whale algorithm and particle swarm optimization algorithm, the algorithm in this paper has more advantages in processing speed and optimization effect. However, for paths with higher complexity, the improved algorithm cannot generate an optimal solution that satisfies the coverage condition ever time. Future research will focus on this issue.

References

[1] JI. F. (2019). Research on software testing automation mechanism based on improved genetic algorithms. *Information Technology*. 2019(10). pp.88-93.

[2] Huang C H. (2021). Test Cases Automatic Generation Based on Chaotic Genetic Algorithm. *Computer & Digital Engineering*. 49(01), PP.31-35.

[3] Sun X Y. (2021). Research on Automatic Generation Algorithm of Test Cases Based on Data Flow Coverage. *University of Electronic Science and Technology of China*.

[4] Zhang X F. (2020). Research and Application of Automatic Generation of Test Cases Based on Artificial Intelligence. *Shandong University of Science and Technology*.

[5] Yuan G H. (2019). Research on test case

[6] generation based on Hybrid Genetic Algorithm. *Journal of Heilongjiang University of technology*. 19(10). pp.33-38.

[7] S K B. (2019). Validating object-oriented software at design phase by achieving MC/DC. *International Journal of System Assurance Engineering and Management*. 10(4).

[8] Ge H Q. (2011). Generating Algorithm of Recursive Blocks Matrix for Minimum Test Case Set on MC/DC.

Computer system applications. 20(07). pp.195-198.

[9] Aghamohammadi A. (2020). Statement frequency coverage: A code coverage criterion for assessing test suite effectiveness. *Information and Software Technology*.

[10] Yan X Y. (2020). Embedded software testing based on MC/DC criterion. *Digital world*. 2020(10). pp.50-52.

[11] Han Y. (2016). Using genetic algorithm to generate regression test data based on MC/DC. *Nanhua University*.

[12] Wu X Z. (2020). Optimization of vehicle parameters in cyclic condition based on adaptive genetic algorithm. *Journal of Mechanical Strength*. 42(04). pp.849-855.

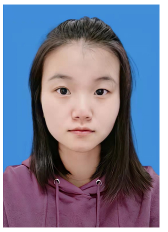
[13] Dai X H. (2021). Automated test case generation based on differential evolution with node branch archive. *Computers & Industrial Engineering*.

[14] E M. (2021). Automation of software test data generation using genetic algorithm and reinforcement learning. *Expert Systems with Applications*.

[15] Hu X Y. (2020). Research on automatic generation method of structural test data based on ant colony algorithm. *Harbin Engineering University*.

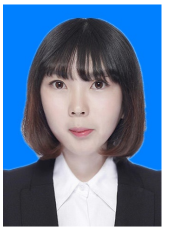
[16] Ren L Y. (2020). Research and design of automatic unit testing framework for object-oriented programs. *Beijing University of Posts and Telecommunications*.

Author Biographies



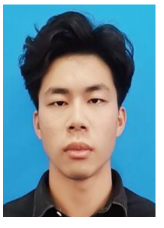
LIU Huiying is a M.Sc. candidate at North China Institute of Aerospace Engineering. Her main research interest includes software testing.

E-mail: 1115194813@qq.com



LIU Ziyang received a M.Sc. degree in engineering from Beijing University of Aeronautics and Astronautics in 2018. Now she is a teacher in North China Institute of Aerospace Engineering. Her main research interests include deep learning and intelligent optimization algorithm.

E-mail: 2210903484@qq.com



YAN Minghui is a M.Sc. candidate at North China Institute of Aerospace Engineering. His main research interest includes deep learning.

E-mail: 1227060370@qq.com