International Journal of Database Theory and Application
Vol.9, No.3 (2016), pp. 73-80
http://dx.doi.org/10.14257/ijdta.2016.9.3.08

Research of Software's Detection Data Generation Based on Improved Monkey Algorithm

Ping Chen[1] and Min Xia[2]

[1] *(Ma'anshan Teacher's College, Ma'anshan, 243041)*
[2] *(Anhui University of Technology, Ma'anshan, 243002)*
anhuicp@126.com, 2237690981@qq.com

Abstract

Generation of detection data is the most key technology at detection stage with some practical significance to improve the detection automation degree of software. Although certain achievements have been made in introducing monkey algorithm into the detection process, the algorithm itself has defects like being easy to fall into local optimum and rapid convergence speed, etc. This paper first introduces chaos algorithm and reverse learning to initialize the initial position of monkey swarms, and then makes improvement to the scrawling, watching and hopping process of monkey algorithm so as to increase the convergence precision of improved algorithm. Simulation experiments show that compared with genetic algorithm and algorithm in this paper, particle swarm algorithm and ant colony optimization has made significant progress.

Keywords*: monkey algorithm; chaos algorithm; reverse learning; detection data*

1. Introduction

In software testing, how can automatically generate test data is the key to efficiency is improved software testing. Literature studies show that algorithms are based on artificial intelligence research in the field of emphasis and difficulty, the widely used artificial intelligence algorithm with genetic algorithm and Particle Swarm Optimization algorithm. Literature [1] is a hybrid Particle Swarm algorithm HPSO, and its application in automatic test data generation, and achieved certain results. Literature [2] an improved algorithm of particle swarm optimization algorithm for AMPSO, results from algebra and convergence time are two contrastive analysis, proved that the method can not only prevent premature convergence algorithm problems, and improve the efficiency of the test data generation. Literature [3] an improved quantum genetic algorithm (IQGA) to automatically generate test data, experimental results show that the improvement of quantum genetic algorithm with time short, generate test data, coverage advantages of automatic test data generation efficiency than quantum genetic algorithm and traditional genetic algorithm. Literature [4] presents a structured chaos optimization algorithm based on Tent map, the experiments show that the algorithm generates test data faster, smaller, have higher efficiencies. Literature [5] proposed a solution space search scheme based on artificial fish-swarm algorithm in each time of iteration and in view of the current global optimal solutions for local search, and eliminated some poor individual; subsequently, contraction according the best individual solution space search area, and randomly generated some of the individual within the space. Finally, in two experiments on decision maker in a triangle, the convergence speed High accuracy. Literature [6] presents a dynamic Adaptive Particle Swarm Optimization for contraction expansion factor (AQPSO). Evolution by introducing particles of the algorithm and degree of polymerization, Contracting and expanding factor with the particle evolution factor and degree of polymerization factors varies, so as to implement the dynamic adaptability,

ISSN: 2005-4270 IJDTA

automated software test data generation proved its effectiveness and viability. Literature [7] proposed a combination of test data generation based on Particle Swarm Optimization algorithm, introduced the concept of decay threshold, elected after the end of each search process used in excellent attenuation range of particles. Compared with the traditional method, under the prerequisite of ensuring the optimization level, effectively reducing the number of optimization, improve the efficiency of the test data generated.

Based on achievements of the above research, this paper proposes introducing monkey algorithm into detection data, which is of certain significance. Monkey algorithm is featured with simple algorithm, being easy to be realized and rapid search speed. Meanwhile, it also has deficiencies like poor local search capacity and being easy to fall into early ripe. This paper makes improvement towards scrawling, watching and hopping process of monkey algorithm. And it is shown through experiment that compared with genetic algorithm and algorithm in this paper, particle swarm algorithm has certain advantages.

2. Description of Software Detection Data

Software detection is a complete process including a lot of detection data. Through setting detection data of automatic software, the entire detection process can be detected with better output of results. Set m sub-nodes on the path to be detected, and n parameters, then the m sub-functions are: $F_1 = f_1(x_1, x_2, \cdots x_n)$, $F_2 = f_2(x_1, x_2, \cdots x_n)$ … $F_m = f_m(x_1, x_2, \cdots x_n)$, so function of detection data on each path is the aggregate of each sub function: $F = \{F_1, F_2, \cdots F_n\}$ with detection data as elements of the particle swarm vector x. Firstly, generate detection data at random, and use monkey algorithm to search the optimal detection data to make the fitness function reach its maximum value.

3. Basic Algorithm Description

Monkey algorithm (in short, MA) is an intelligent algorithm to solve a large scale of optimization problems with its main idea as imitating monkeys' behavior of climbing mountains and realizing searching of the optimal solution through simulating the scrawling, watching and hopping during the process of climbing mountains. During the process of getting the solution, N is used to show the dimension of solution space and M is used to show the scale of monkey population, then the spatial location of the i monkey is as follows: $X_i = (x_{i1}, x_{i2}, \cdots x_{in})\ \ i \in [1, M]$, herein, x_{ij} refers to the actual location of each monkey in the one dimension.

3.1 Scrawling Process

The scrawling process is a process of constantly improving and gradually optimizing the value of target function, and the design process is as follows:

(1) Generate vector $\Delta x_i = (\Delta x_{i1}, \Delta x_{i2}, \cdots \Delta x_{in})$ at random, herein, α is the pace length of monkey in scrawling, which is related to the precision of the optimal solution; set Δx_i and take the value $-\alpha$ or α with the same probability 0.5.

(2) Calculate $f_{ij}^{'}(x_i) = \dfrac{f(x_i + 2\Delta x_i) - f(x_i)}{2\Delta x_{ij}} (j = 1, 2 \cdots n)$, herein, the calculated value of $f_{ij}^{'}(x_i)$ is as follows $\{f_{i1}^{'}(x_i), f_{i2}^{'}(x_i), \cdots f_{in}^{'}(x_i)\}$, which is the pseudo-gradient of each

target function.

(3) $y_i = x_{ij} + \alpha \square sign(f_{ij}^{'}(x_i)),\ j = 1, 2, \cdots n$ and $y = \{y_1, y_2 \cdots y_n\}$

(4) When y meets the requirement of restriction function, that is, with feasible solution, $y \to x_i$, otherwise, keep the value of x_i unchanged. Repeat step (1) and step (4) until the maximum scrawling times or the completion times are met.

3.2 Watching Process

Through the above scrawling process, each monkey can get to the top of their respective mountain, that is to say, obtain the local optimum temporarily. Then, observe whether there is better mountain nearby, it so, it will hop to the higher position, so it is a watching process to find a higher position. In the monkey algorithm a parameter β is defined, that is, the scope of vision of monkey algorithm, the definition is as follows:

(1) Choose adjacent points at random with the scope of vision $(x_{ij} - \beta, x_{ij} + \beta)$, and produce an y_i at random with $y = (y_1, y_2 \cdots y_n)$.

(2) If $f(y) > f(x_i)$ and the effective solution of y is met, $y \to x_i$, otherwise, expand to the scope of (1) until the suitable y is found.

(3) Repeat the action from step (1) to step (2), and find y

3.3 Hopping Process

Hopping mainly to the process of the search process will be transferred to new regions in the current zone, select all of the gravity as the fulcrum of the monkeys, each animal along the current position point to the pivot direction to their new search area for monkey hopping process is as follows:

(1) Produce a real number θ at random from $[a, b]$.

(2) Set $y_j = x_{ij} + \theta(p_j - x_{ij})$, herein $p_j = \dfrac{\sum_{l=1}^{M} (x_{lj} - x_{ij})}{M - 1}$, and the aggregate p_j of is the pivotal.

(3) If y is feasible, $y \to x_i$, or implement (1)(2) repeatedly until a new feasible solution is found.

4. Description of Improved Monkey Algorithm in Data Detection

4.1 Improvement of Initial Position

As the monkey algorithm does not describe the initial position, so after screening and filtering the initialization location of individual monkey algorithm, time to get the optimal solution can be saved to some extent. This paper used chaos algorithm to initialize monkey algorithm aiming at the initial position.

As chaos algorithm is featured with better ergodicity without repetition, so this paper uses chaos method to set the initial location, herein, y is the chaos variable and k is the iteration times.

$$y_k = \begin{cases} 2y_{k-1} & y_{k-1} \in [0, 0.5] \\ 2(1 - y_{k-1}) & y_{k-1} \in [0.5, 1] \end{cases} \quad (1)$$

Produce chaos variable within the range [a,b] and turn it into the domain of optimization problem as follows:

$$y_k^{'} = \frac{x_k - a}{b - a} \quad (2)$$

Choose monkey individual sequence with the best fitness from $y_k \cap y_k^{'}$ as the initial population.

4.2 Improvement of the Scrawling Process

From the scrawling process in Chapter 2.1, the value of α depends on the level of precision of solution. When the value of α gradually reduces, times of step increase gradually with more and more times of iteration. This paper makes the balance through positioning step so that when the value of α is relatively larger, search the local optimum can be better carried out, and when the value of α is relatively smaller, it is conducive to find the local optimal value. Set the value of α is $[\alpha_{\min}, \alpha_{\max}]$, the maximum iteration time in the algorithm is Max, and the minimum iteration time is Min with λ as the current iteration time, so α^{λ} refers to the corresponding pace of iteration, which is set according to the following formula:

$$\alpha^{\lambda} = e^{\frac{\min\lambda}{\max}} \square \alpha^{\lambda-1} \quad (3)$$

With the constant increase of iteration times, the value of λ increases gradually, but the value of $e^{\frac{\min\lambda}{\max}}$ is reduced gradually so that the α value of pace gradually reaches $\alpha_{\min}$ so as to avoid the non-convergence state of the algorithm during the searching process.

4.3 Improvement of the Watching Process

After the crawling process, monkey individuals i reach top of their mountains respectively. The smaller the value of β, the easier it is to fall into local optimum; and the larger the value of β, convergence speed of the algorithm is easy to become slow. In order to balance this situation, β is set as follows in this paper.

$$\beta^{\lambda} = \frac{\lambda}{\max + \min} \square \beta^{\lambda-1} \quad (4)$$

Herein, λ refers to the times of iteration, max and min are the maximum iteration times and minimum iteration times respectively. Therefore, scope of vision during the watching process is changed into $(x_{ij} - \beta^{\lambda}, x_{ij} + \beta^{\lambda})$, and with the increase of iteration times, the value of β gradually reaches its maximum value so as to effectively reduce the probability of falling into local optimum rapidly. Meanwhile, convergence speed of the algorithm can be maintained so as to gradually reach a balance.

4.4 Improvement of Hopping Process

In the process of hopping, transfer the current region to new region, and in monkey algorithm, pivots of monkey enter new region respectively. For x_{ij}, update it according to the following formula:

$$x_{ij}^{\lambda} = p_j + \theta(p_j - \kappa x_{ij}^{\lambda-1}) \quad (5)$$

In the formula, use parameter κ during the crawling process to update vector of monkey's location in each time of iteration λ. It can effectively avoid too great randomness of monkey during the hopping process by taking p_j as the pivotal during the hopping process, thus obtaining the location of monkey algorithm after hopping.

According to classification of detection data, each test corresponds to the classification of individual data for individual monkey algorithm, the best collection of test data is the value of optimal position of individual monkeys.

Step 1: set parameters of monkey algorithm with scale of population as M, set crawling step, scope of vision and other parameters.

Step 2: Initialize location of monkey algorithm individual through chaos algorithm.

Step 3: Calculate the location of monkey algorithm according to the improved pseudo-gradient during crawling process.

Step 4: Search and update location within the scope of visional parameter and update the new location of monkey algorithm to the optimal location.

Step 5: Make climbing process towards the monkey algorithm and use positioning step to search new position of individual monkey algorithm.

Step 6: Make watching process towards the monkey algorithm and use new vision distance to select the next distance.

Step 7: Make hopping process towards the monkey algorithm and set parameters to obtain new positions of monkeys after hopping.

Step 8: Detect whether the condition is met; if it is met, end the algorithm; otherwise, turn to step 3.

Step 9: Output the location of optimal money algorithm individuals so as to obtain the corresponding optimal detection data.

5. Simulation Experiment

Triangle discrimination issue is widely used in software test generation research data. In this paper, two types of questions to verify the efficiency of this method. The experiment with the equilateral triangle all discrimination if statements path to the program as a test subject to compare the proposed method and GA algorithm to obtain the optimal solution iterations respective needs, and find the optimal solution for its time consumed were compared.

Table 1. Iteration Times and Time of Two Methods' to Generate Detection Data of Equilateral Triangle

Scale of Population	Find Iteration Times of the Optimal Solution						Find Iteration Time of the Optimal Solution/ms	
	GA			Algorithm in this paper			GA	Algorithm in this paper
	Best	Worst	Average	Best	Worst	Average		
100	4	417	289	2	100	26	225.4	74
200	10	507	214.2	5	110	18	368.4	82
300	13	281	181	7	75	28	265.2	79
400	19	314	212	10	88	17	312.7	78
600	30	398	192	15	102	85	382.2	95

It can be seen from Table 1 that in the best case, iteration times of data needed in method of this paper is about 1/3-1/2 of that of GA, and 1/4 in the worst case. According

to the average situation, iteration times of detection data needed by method in this paper are about 1/4 of GA and the consumed time is about 1/4 of that of GA. In general, the efficiency of this method to generate test data equilateral triangle is significantly higher than GA.

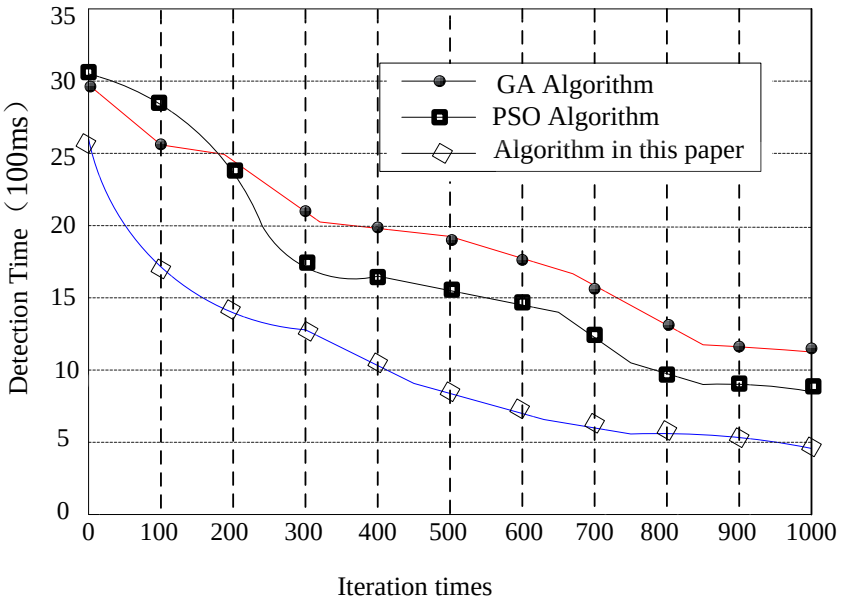

Figure 1. Comparison of Three Algorithms in Detection Time

It can be found in Figure1 that compared with commonly used genetic algorithm and particle swarm algorithm, algorithm in this paper has some advantages in terms of detection time, greatly improving detection effectiveness.

6. Conclusion

The existing structure and detection data are generated automatically, and most of them adopt genetic algorithm methods, the deficiencies of which are complex algorithm and it is not easy to set data. In this paper, a method based on monkey algorithm to automatically generate software structure detection data is proposed, and the efficiency of this algorithm to automatically generate detection data is much higher than that of genetic algorithm.

References

[1] D. Y. Hua and D. Y. Qian, "Automatic software test data generation based on hybrid particle swarm optimization", Journal of Computer Applications, vol. 35, no. 2, **(2015)**, pp. 545-549.
[2] S. Nan, Z. Y. Zhou and H. W. Tao, "Test data generation based on adaptive mutation particle swamp algorithm", Application Research of Computers, vol. 35, no. 3, **(2015)**, pp. 786-789.
[3] J. H. Jing and S. J. Mei, "Research on the Improved Quantum Genetic Algorithm Applied to Test Data Automatically Generated", Journal of Chinese Computer Systems, vol. 34, no. 2, **(2013)**, pp. 386-389.
[4] W. L, Hong, W. S. Yan and S, J. Ze, "A Data Generation Algorithm for staged Combinatorial Test", Computer Applications and Software, vol. 30, no. 3, **(2013)**, pp. 67-70.
[5] W. P. Cong and Q. Xu, "Path test data generation based on improved artificial fish swarm algorithm", Journal of Computer Applications, vol. 33, no. 4, **(2013)**, pp. 1139-1141.
[6] H. H. Mei, C. X. Wen and Z, Yan, "Automatic generation of software test data based on adaptive QPSO algorithm", China Measurement & Test, vol. 39, no. 3, **(2013)**, pp. 100-103.
[7] W. J. Feng, S. Chao and J. S. Da, "Improved algorithm for combinatorial test data generation based on particle swarm optimization", Journal of Harbin Engineering University, vol. 34, no. 3, **(2013)**, pp. 477-482.

Authors

Ping Chen, (1978.07-), male, Huaining, Anhui, master, Associate Professor, main research direction: software engineering.

Min Xia, (1977.11-), female, Ma Anshan, Anhui, Master, Lecturer, main research direction: software engineering.

