

## **A new automatic test data generation algorithm based on PSO-ACO**

Xiaomin Zhao<sup>1, a</sup>, Yiting Wang<sup>2, b</sup> and Xiaoming Ding<sup>3, c</sup>

<sup>1</sup> Southwestern University, Beibei District, Chongqing, China

<sup>2</sup> Southwestern University, Beibei District, Chongqing, China

<sup>3</sup> Southwestern University, Beibei District, Chongqing, China

<sup>a</sup>715264078@qq.com, <sup>b</sup> tianwei.yi@qq.com, <sup>c</sup> xmding@swu.edu.cn

**Keywords:** PSO; ACO; Test data generation; Software testing

### **Abstract.**

In view of the shortcomings of the test data generation algorithm including particle swarm optimization algorithm and ant colony algorithm, a new algorithm is proposed, which is based on the combination of particle swarm algorithm and parameter adjustment. This algorithm can dynamically adjust its search capabilities based on the fitness value of particles, combine the advantages of particle swarm optimization (PSO) algorithm and ant colony algorithm (ACO) to ensure the convergence and accuracy of the algorithm. Experiments show that the new algorithm can effectively improve the efficiency of test data generation.

### **Introduction**

Software testing is accompanied with the whole software development cycle, so it is one of the important means to control the cost of development and guarantee the quality of products. Test data can be generated automatically to shorten the test cycle, reduce testing costs. In the initial field of automatic test data generation method more often used include random method, symbolic execution method, program instrumentation method and iterative relaxation method. In view of the shortcomings of the test data generation algorithm, But they are poor in practical application in large projects, and the implementation efficiency is low [1].

Now there are a variety of heuristic search algorithms applied to automatic test data generation. The genetic algorithm, particle swarm optimization and ant colony algorithm are the most widely used among them. These three algorithms also have disadvantages. Scholars have done a lot of research on the disadvantages of these three algorithms: FU B successfully introduced the ant colony algorithm to the software test data automatic generation model [2]. Aiming at the problems of low convergence accuracy and easy to fall into local extreme, Dai Yuqian and Dong Yuehua proposed a hybrid particle swarm optimization algorithm HPSO [3]. Zhou Hong et al. Combine the genetic algorithm and particle swarm algorithm to form a new hybrid algorithm, and successfully applied to the automatic generation of software test data [1]. Yu Zhenyang put forward an automatic generation algorithm based on quantum particle swarm optimization (QPSO) test data in the software testing [4]. Shi Guiying and her partners proposed a new improved particle swarm optimization algorithm, effectively overcome the particle swarm algorithm prone to premature stagnation of the defect [5]. Lin Mu Gang presented a dynamic adjustment of parameters adaptive particle swarm optimization, to effectively regulate the global and local search ability of the algorithm and maintain the individuality of particles [6]. Shi and Eberhart proposed by decreasing linear inertia weight adjustment method to improve the learning ability of particles effectively. The local search ability of the algorithm and the global search ability are balanced by rationally adjusting the algorithm parameters [7].

In the above studies, we can see improvement strategies for these search algorithms can be divided into two categories: one is based on the idea of hybrid algorithm; the other is to adjust the algorithm parameters. In this article we overlay these two strategies, that is the introduction of the ant colony algorithm's "pheromone mechanism [9]" in particle swarm algorithm and dynamically adjust parameters on this basis. We propose a new automatic test data generation algorithm base on the PSO and ACO Algorithm. This algorithm makes full use of the global search capability of PSO algorithm in the search early to get the pheromone, and then use the ant colony algorithm's "pheromone mechanism" to search for an exact solution to the problem. While the particles of different fitness value particle swarm algorithm for the desired search ability of different problems, we have introduced the concept of particle relatively excellent in the new algorithm to dynamically adjust the inertia weight and learning factor. It is through the rational control algorithm convergence speed make the algorithm search results more accurate and effective. Experiments prove that the new algorithm can effectively avoid defects such as premature convergence and local extreme in the particle swarm optimization algorithm and the ant colony algorithm, which has better convergence performance.

## Text

**Particle Swarm Optimization(PSO).** Particle swarm optimization (PSO) is a swarm intelligence evolutionary algorithm [8], which is proposed by Berhart and Dr. Kennedy in 1995. Assume that there is a particle group in the  $m$ -dimensional space. Its population total number of this particle group is  $n$ , then at time  $t$ , position of a particle  $i$  is denoted as  $x_i^t=(x_{i1},x_{i2},\dots,x_{im})$ , its current speed is  $v_i^t=(v_{i1},v_{i2},\dots,v_{im})$ , while the current particle obtained in the search process history optimal solution as individual extreme  $p_i^t=(p_{i1},p_{i2},\dots,p_{im})$ , and the history optimal solution of particle group now found referred to as global extreme  $g_i^t=(g_{i1},g_{i2},\dots,g_{im})$ . Each particle updates respective positions and speed according to equation (1) and Equation (2) at time  $t + 1$ :

$$v_i^{t+1}=\omega v_i^t+c_1r_1(p_i^t-x_i^t)+c_2r_2(g_i^t-x_i^t) \quad (1)$$

$$x_i^{t+1}=x_i^t+v_i^{t+1} \quad (2)$$

Which  $i \in (1,2,\dots,n)$ ,  $\omega$  is the inertia weight used to represent the influence of the particle velocity at time  $t$  to time  $t + 1$ ;  $c_1$  and  $c_2$  are learning factors, which are used to adjust the direction of particle toward its best position and the step of particle flying to the global best position ( $c_1, c_2 \in [0,2]$ );  $r_1$  and  $r_2$  are speed constraint factors when the particle is during the location update, which are two uniform distribution random numbers in  $[0, 1]$ .

**Ant colony algorithm (ACO).** Ant colony algorithm (ACO) simulates the collective behavior of ant colony composed by a large number of ants. Suppose there are  $n$  ants, every ant on each path based on previous ant pheromones left to choose the way to go.  $\tau_{rs}(t)$  represents the total amount of pheromones on the path from node  $r$  to node  $s$  at time  $t$ , and the initial amount of pheromones on each path is  $\tau_{rs}(0)=C$  ( $C$  is a constant). The probability  $p_{rs}^k$  of the ant  $k$  transfer from node  $r$  to node  $s$  expressed as follows:

$$p_{rs}^k = \begin{cases} \frac{\tau_{rs}^\alpha \eta_{rs}^\beta}{\sum_{q \in \text{Allowed}_k} \tau_{rq}^\alpha \eta_{rq}^\beta}, & s \in \text{Allowed}_k \\ 0, & \text{else} \end{cases} \quad (3)$$

Wherein  $\text{Allowed}_k$  represents a point set which the ant  $k$  able to choose from at the next time in the current. Its traversed all points will be recorded in the taboo list  $\text{tabu}_k$ ,  $\eta_{rs}^\beta = \frac{1}{d(r,s)}$  represents expected value of the ant  $k$  moves from the node  $r$  to node  $s$ .  $\alpha$  is the importance's degree of a path's expected value and  $\beta$  is the importance's degree of the total amount of pheromones.  $\alpha$  represents the

importance's degree of the total amount of information in the path.  $\beta$  represents the importance's degree of the path expected value.

**A new automatic generation algorithm of test data based on PSO-ACO.** In the particle swarm optimization algorithm, when the particle's individual extreme is similar to the global extreme of this particle, it will be trapped in the local extreme due to premature convergence. Or the convergence speed will be slow because of improper parameter setting. For the shortcomings in the particle swarm algorithm, what the principle we based on in this paper is "fitness of particles having different values to find global optimal solution in the process of search capabilities required for different". We adjust inertia weight and learning factor reasonable and dynamically to improve the performance of particle swarm algorithm. The new particle swarm optimization algorithm is also introduced the ant colony algorithm's "pheromone mechanism": By local search in the neighborhood of the current individual extreme value  $g_i^t$  of each particle to generate  $k$  points, a new sequence  $p_{\text{new}}$  containing the current individual extreme  $g_i^t$  is obtained. The probability of  $k+1$  will be selected as the new particle according to the probability formula, So that the probability of a larger point is no longer single, which reduces the possibility of the particle swarm algorithm into local extreme value [9].

**Parameter adjustment in particle swarm optimization algorithm.** In the formula (1) and (2), we find that the size of inertia weight  $\omega$  has a direct effect on the convergence of the particle swarm optimization algorithm. The change of particle size is proportional to the growth of inertia weight. The size of the learning factor  $c_1$  and  $c_2$  determines the influence degree of individual extreme value and global polar value on the particle motion.

We also find that the more the particle position changes in the particle swarm optimization, the wider the search range, the lower the search accuracy; similarly the smaller the particle position change, the smaller the search scope, and the higher the accuracy of the search. The ultimate goal of using particle swarm optimization algorithm is to find the global optimal solution, which is the best solution. For particles with higher fitness, it is closer to the global optimal solution, which is more need is the search accuracy more than the search range of the particle; For the lower level of the particle, it is far from the optimal solution, so it is more necessary to expand the search scope and improve the convergence rate of the algorithm.

In order to better describe the particle fitness values of the relationship between inertia weight and learning factor, we introduced concept of particle relative excellent proposed by Lin Mugang [6]. The formula is:

$$\text{red}_i^k = \begin{cases} 1, & \text{if } f_{\text{worse}}^k = f_{\text{best}}^k \\ \frac{f_i^k - f_{\text{worse}}^k}{f_{\text{best}}^k - f_{\text{worse}}^k}, & \text{else} \end{cases} \quad (4)$$

Where  $\text{red}_i^k$  ( $0 \leq \text{red}_i^k \leq 1$ ) is the relative excellence of a particle  $i$  in the  $k$  generation, the fitness value of this particle is  $f_i^k$ .  $f_{\text{worse}}^k$  and  $f_{\text{best}}^k$  are the worst fitness value and fitness value of the particle swarm respectively.

From the previous paper, for particles with larger fitness values, we need to use a relatively smaller inertia weight  $\omega$ , a smaller learning factor  $c_1$  and a larger learning factor  $c_2$  to improve the local search ability and precision; For particles with smaller fitness values, we need to use a relatively large inertia weight  $\omega$ , the larger learning factor  $c_1$ , and smaller learning factor  $c_2$  to improve their global search ability and search range. Therefore, according to the concept of relative excellence, we put forward the following formula to dynamically adjust the inertia weight and learning factor of particles in the search process:

$$\omega_i^k = \omega_{\min} + (\omega_{\max} - \omega_{\min})(1 - \text{red}_i^k) \quad (5)$$

$$c_{1i}^k = c_{1\text{end}} + (c_{1\text{start}} - c_{1\text{end}})(1 - \text{red}_i^k) \quad (6)$$

$$c_{2i}^k = \omega_{2end} + (c_{2start} - c_{2end}) \text{red}_i^k \quad (7)$$

Where  $\omega_{max}$  is the maximum value of inertia weight  $\omega$ , and  $\omega_{min}$  is the minimum value of inertia weight  $\omega$ .  $c_{1start}$  is the initial values of the learning factor  $c_1$  before adjustment, and  $c_{1end}$  is the final value of the learning factor  $c_1$  after adjustment.  $c_{2start}$  is the initial values of the learning factor  $c_2$  before adjustment, and  $c_{2end}$  is the final value of the learning factor  $c_2$  after adjustment.

**Fitness function.** A particle's fitness function determines the probability that the particle will be selected. A construction of the fitness function can directly affect the efficiency of the algorithm. In this article we will use "branch function superposition method [10]" to construct the fitness function: We insert pile operation before branch point of the program, that is to insert a fitness function  $f(x_1, x_2, \dots, x_n)$  to the real valued predicate mapping,  $n$  is the number of parameters in the path to be measured. Finally it returns a real value  $\theta$  to measure the coverage of the actual execution path to the selected path. If you choose a path with  $m$  branches, the fitness function of the test program is  $F$  as follows:

$$F = \sum_{i=1}^m g(\theta_i) \quad (9)$$

$\theta_i$  and  $g(\theta_i)$  satisfy the following relations:

$$g(\theta_i) = \begin{cases} g(\theta_i), & \theta_i > 0 \\ 0, & \theta_i \leq 0 \end{cases} \quad (10)$$

**Description of the new algorithm.** In the new algorithm, based on the formula (3) we give the probability of the  $j$ -th ( $1 \leq j \leq k+1$ ) point in the new sequence  $p_{new}$  will be selected is:

$$p(j) = \begin{cases} \max \left( f(p_{new}(j)) \right), & a \geq a_0 \\ \frac{f(p_{new}(j))}{\sum_{i=1}^{k+1} p_{new}(i)}, & a < a_0 \end{cases} \quad (8)$$

Where  $a_0$  ( $0 \leq a_0 \leq 1$ ) is a given parameter, the function  $f(p_{new}(j))$  is the fitness value of the point  $p_{new}(j)$ . The basic flow chart of the new automatic generation algorithm of test data based on PSO-ACO shown in figure 1:

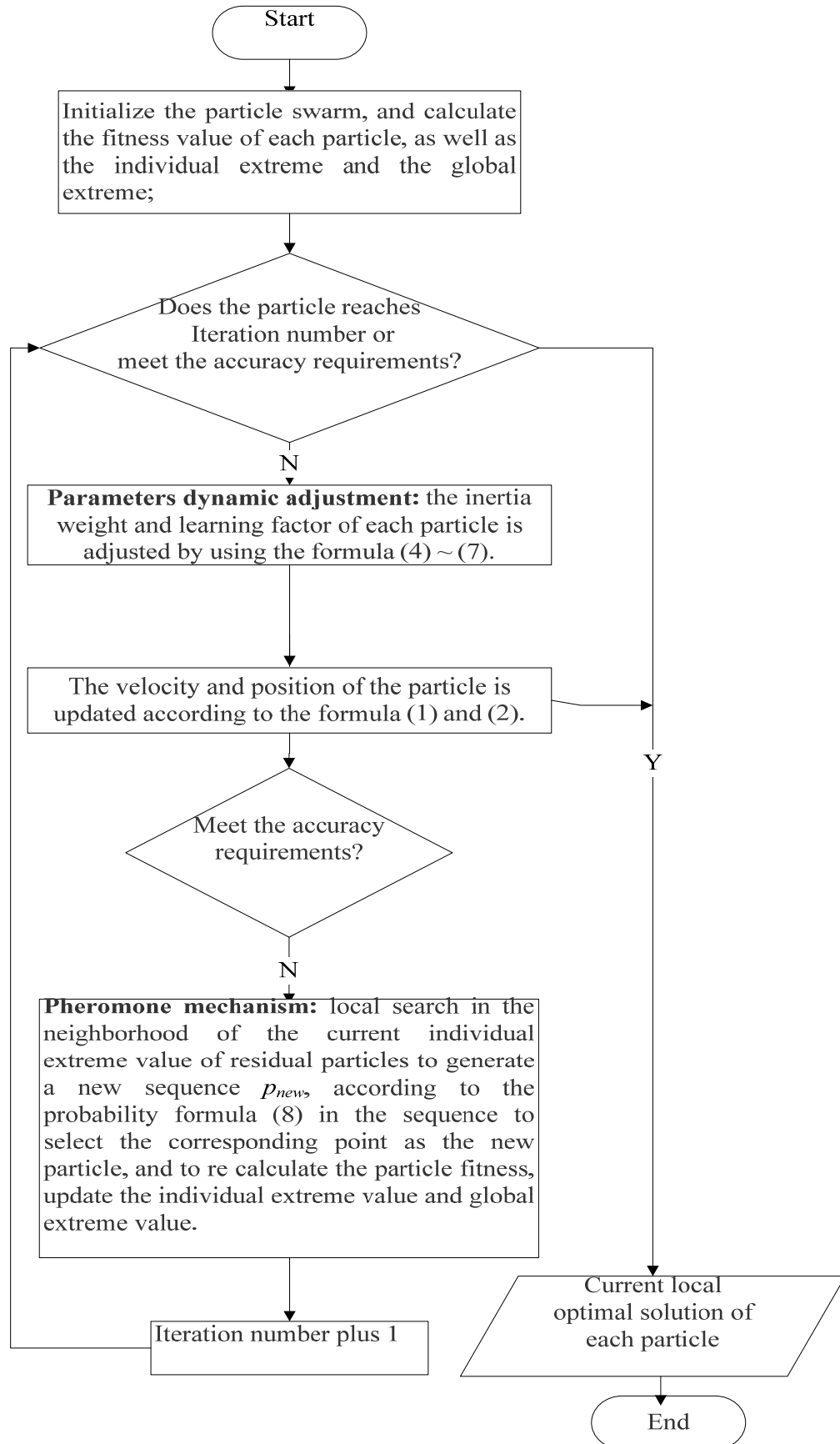


Fig. 1 Basic flow char of the new test data automatic generation algorithm based on PSO-ACO

**Equations.** In this paper, the experiment used judging program of equilateral triangle as subjects. By using the particle swarm optimization, ant colony algorithm, a new improved particle swarm optimization algorithm (IPSO) proposed by Shi Guiying[5] and the new algorithm in this article to

test data automatically generated experiments respectively. Then we compared the efficiency and diversity of those algorithms to verify the feasibility of the new algorithm.

In this experiment, the initial population size of  $n = 100$ , maximum number of iterations  $k = 20$ ; In this new algorithm, Learning factor of PSO  $c_1 = c_2 = 2$ , inertia weight maximum  $\omega_{\max} = 0.7$ , inertia weight minimum  $\omega_{\min} = 0.4$ , the learning factor  $c_1$  initial value before adjustment  $c_{1\text{start}} = 2.75$ , final value of adjusted  $c_{1\text{end}} = 1.25$ , learning factor  $c_2$  initial value before adjustment  $c_{2\text{start}} = 0.5$  and the final value adjusted  $c_{2\text{end}} = 2.25$ . Finally get the three algorithms in determining program equilateral triangle test data shown in Figure 2:

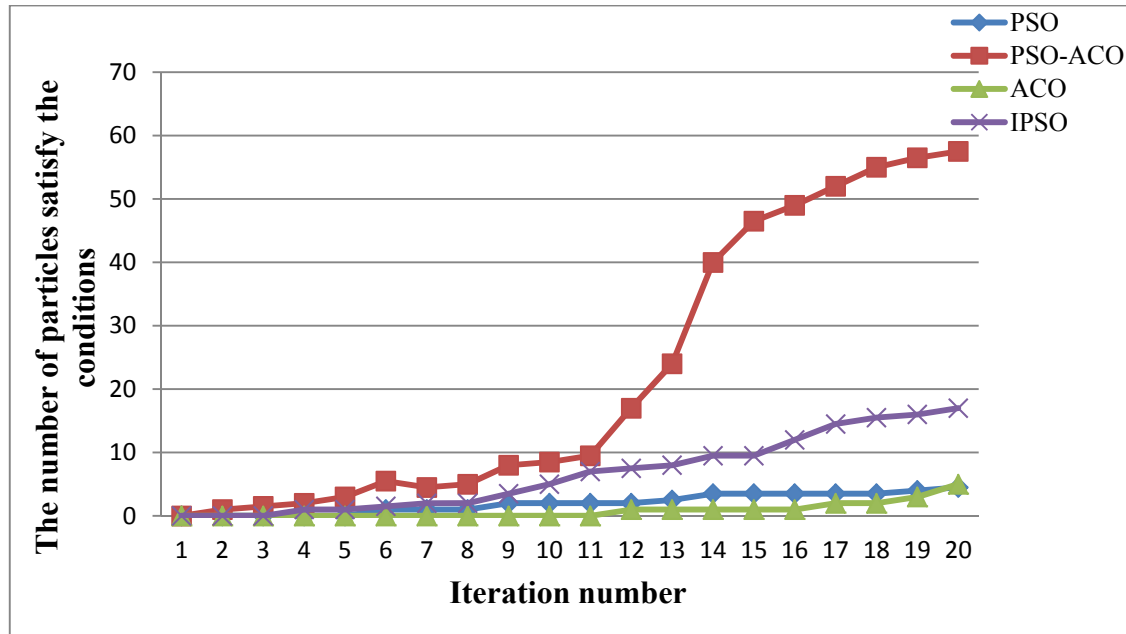


Fig. 2 Three algorithms experimental results comparison chart

We also recorded changes in particle swarm diversity. Change the maximum Iteration number  $n$  to 60, and the other initialization parameters unchanged. The diversity changes comparison chart shown in Figure 3:

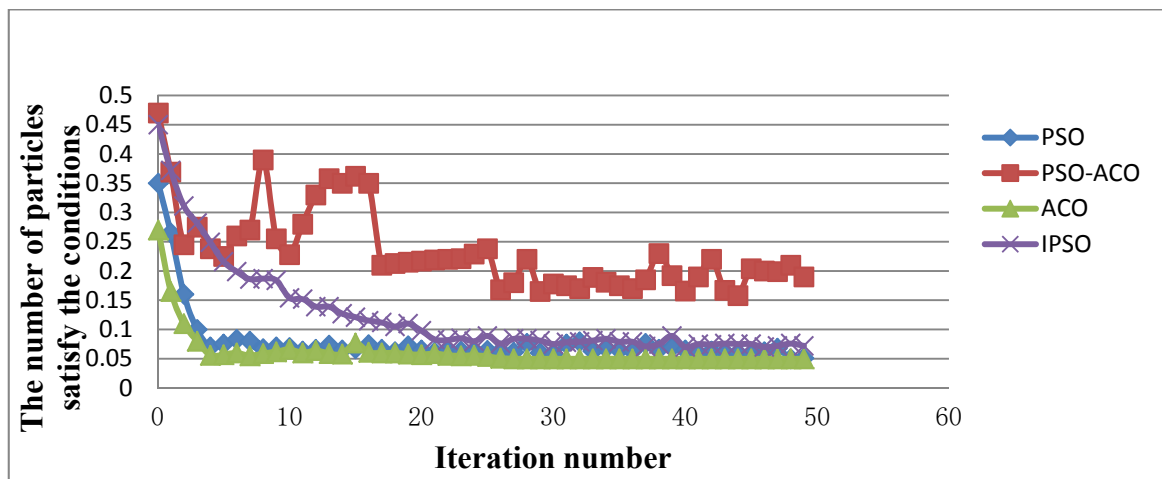


Fig. 3 Three algorithms particle swarm diversity comparison chart

In Figure 2, we can clearly see that the new algorithm based on PSO-ACO can ensure the individual's diversity, which is based on the pheromone mechanism and the dynamic adjustment of parameters. And it will not in a very short period of time enable a few individuals with high fitness to occupy most of the total population. This new algorithm is better to avoid premature and local

extreme, making the path coverage more comprehensive and reasonable. The generated test data of the new algorithm not only has more practical value, but also more reliable.

Similarly, In Figure 3, particle swarm diversity of PSO and ACO algorithm was relatively large in the initial search, but the search carried out with a sharp decline in particle swarm diversity; along with the late search the iteration proceeds, particle swarm diversity remained almost unchanged. In the algorithm proposed by Shi Guiying, the particle diversity slower decline in the initial search, which makes the convergence of the algorithm deteriorated. In contrast, the new algorithm in this article dynamically adjusts search capabilities in the different periods of algorithm. So that it has a fast convergence early in the algorithm search; with the increase in Iteration number, particle swarm diversity sex at a reduced while still maintaining the small-scale fluctuations. And the search accuracy of particle swarm is increased, the search range become smaller; particle swarm diversity leveled off late in the search, ensuring the convergence performance of the new algorithm.

## Summary

This paper presents an automatically generated algorithm of test data based on PSO-ACO. By introducing the pheromone mechanism of ACO algorithm into PSO algorithm, the new algorithm expands the search range and search accuracy. While PSO algorithm parameters were dynamic adjustment based on the fitness value, it effectively ensures the convergence of the algorithm. Experimental result shows that to a certain extent the new algorithm can avoid a premature and local extreme problems occurring in the PSO algorithm and ACO algorithm, and the generated test data is more reasonable and effective.

## References

- [1] ZHOU Hong, ZHANG Sheng, et al: Automatic generation of path test data based on GA-PSO [J]. Application Research of Computers, 2010, 27(4): 1367-1369.
- [2] FU B:Automated software test data generation based on ant colony algorithm [J]. Computer Engineering and Applications, 2007, 43(12): 97-99.
- [3] DONG Yuehua. DAI Yuqian, et al:Automatic software test data generation based on hybrid particle swarm optimization, 2015, 35(2): 545-549
- [4] YU Zhenyang:Algorithm for Automatic Generation of Test Data Based on Quantum Particle Swarm Optimization [J]. Journal of Changchun University of Science and Technology (Natural Science Edition), 2012, 35(2):173-175.
- [5] SHI Guiying, WU Yajuan, NI Hongmei:An Improved Particle Swarm Optimization[J]. Journal of Changchun University of Science and Technology (Natural Science Edition, 2011, 34(2):135-137
- [6] LJN Mugang, et al:Adaptive Particle Swarm Optimization Algorithm with Dynamically Adjusting Parameters. Natural Science Journal of Xiangtan University, 2010, 32(1):122-126.
- [7] SHI Y, EBERHART R C:A modified particle swarm optimizer[C]. Proc of the IEEE International Conference on Evolutionary Computation. NJ:IEEE Press. 1998: 69-73.
- [8] Kennedy J:Eberhert R.Particle swarm optimization [J] . IEEE International Conference on Neural Networks, 1995: 1942-1948.
- [9] Shi Haijun, Wang Zhigang, et al:Mutation Operator Particle Swarm Optimization [J]. Journal of Changchun University of Science and Technology (Natural Science Edition), 2007,30(3):81-83.
- [10] KOREL B:Automated software test data generation[J]. IEEE Transactions on Software Engineering, 1990, 16(8): 870—879.