Neurocomputing 158 (2015) 109–116

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Automatic test data generation based on reduced adaptive particle swarm optimization algorithm

Shujuan Jiang, Jiaojiao Shi *, Yanmei Zhang, Han Han

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu, China

ARTICLE INFO

Article history:
Received 23 August 2013
Received in revised form
2 October 2014
Accepted 31 January 2015
Communicated by A. Konar
Available online 11 February 2015

Keywords:
Particle swarm algorithm
Test data generation
Inertia weight
Adaptive adjustment scheme
Particle aggregation degree

ABSTRACT

Software testing aims to search a set of test data in the entire search space to satisfy a certain standard of coverage. Therefore, finding an effective approach for automatic test data generation is a key issue of software testing. This paper proposes a new approach of reduced adaptive particle swarm optimization for generating the test data automatically. First, the approach reduces the particle swarm evolution equations and gets an evolution equation without velocity. Then, the approach makes an adaptive adjustment scheme based on inertia weight for the reduced evolution equation, which is different from the methods that directly act on the particle velocity in the past. The approach directly impacts on the particle position, namely actual problem solution. Next, according to the particle fitness and the particle aggregation degree, the population will be divided into three parts and inertia weight of each part will be designed accordingly. This can balance the search capabilities of algorithm between global and local. Finally, the approach is applied to automatic test data generation. The experiments results show that our approach can enhance convergence speed of algorithm and solve the problems that particle swarm algorithm easily falls into the local optimal solution and has low search accuracy. The experiments results also turn out that our approach can improve the efficiency of generating test data automatically.

1. Introduction

Software testing is an important way to guarantee the quality and reliability of software, and automatic test data generation has always been a big challenge in software testing. The particle swarm optimization (PSO) [1, 2] algorithm is used to generate test data automatically in recent years, and this process of generating test data shows it has a great advantage.

However, when we apply the PSO algorithm to automatic test data generations, there are some disadvantages. The main defects are poor local searching capability and easily falling into the local optimal solution. Improving the convergence speed and search precision of the algorithm has an important practical significance in enhancing the efficiency and quality of automatic test data generation.

Although some studies on the population premature convergence [13, 14] were carried out, the studies that quantificationally evaluating individual convergence degree are still rarely seen. So this paper proposes a reduced adaptive particle swarm optimization (RAPSO) approach. First, it reduces the particle swarm evolution equation, and then it makes an adaptive adjustment. The approach can complete evolution process only by particle position update. Different from those that inertia weight of PSO directly acts on particle velocity, our approach directly acts on particle position and makes an impact on particle position, namely the problem solutions. This paper combines the particle fitness with the particle aggregation degree and divides population into three parts, which respectively makes a corresponding adjustment on inertia weight of each part according to the relationship among them. The results show that our approach is effective and has obvious advantages in the convergence speed and the search accuracy.

2. Basic concepts

In PSO, we suppose that the t-th generation population consists of N particles. The i-th particle position and velocity are denoted as $X_i=[x_l^t, x_2^t, \dots x_N^t]$ and $V_i=[v_l^t,\ v_2^t,\ \dots v_i^t, \dots v_N^t]$ respectively. The best point that this particle itself reached in search space so far is denoted as individual optimum *pbest*. The best point that all the particles in the population have reached in search space is denoted as global optimum *gbest*. In the evolution process, update formulas of particle velocity and position are as shown in formulas (1) and (2).

* Corresponding author.
E-mail address: sjiao888@126.com (J. Shi).

http://dx.doi.org/10.1016/j.neucom.2015.01.062
0925-2312/

$$\begin{cases} v_i^{t+1} = w\ v_i^t + c_1\ r_1(pbest - x_i^t) + c_2 r_2(gbest - x_i^t) & (1) \\ x_i^{t+1} = x_i^t + v_i^{t+1} & (2) \end{cases}$$

Where, the w is the inertia weight. The c_1 and c_2 are the learning factors which respectively represent the cognitive parameter and social parameter. Besides $c_1 > 0$, $c_2 > 0$, the r_1 and r_2 are two independent random variables that obey the U (0,1) distribution. They respectively impact cognitive and social acceleration constants (c_1 and c_2). The v_i, $v_i \in [-v_{\max}, v_{\max}]$, is set by the user according to experience. The x_i, $x_i \in [-x_{\max}, x_{\max}]$, is determined by the scope of the problem solutions.

Definition 1:. Particle Average Fitness. In PSO, we suppose that current population size is N. The fitness of i-th particle is f_i. Then the average fitness f_{avg} of the current particle is as shown in formula (3).

$$f_{avg} = \frac{1}{N}\sum_{i=1}^{N} f_i \quad (3)$$

3. RAPSO

3.1. Reduce PSO evolution equations

At present, the evolution equations are two elements based on the particle velocity and the position for all particle swarm algorithms [9,12–16]. In practice applications, the particle velocity maximum v_{max} is set depending on the users' experience generally, which not only consumes a lot of time and is hard to handle, but also impacts on particle convergence speed and accuracy. Improper velocity setting even decreases the performance and accuracy of the algorithm. So only when v_{max} is in a suitable range, can the algorithm have a good performance.

In the test data generation, the particle position x_i represents the solution of the target problem. If we set corresponding constraints, the process of algorithm implementation is to make x_i infinite approach to the optimal solution of the problem. Thus we set the value of x_i reasonably, and it is enough to take into account the direct changes of x_i. Particle velocity represents the speed when the particle runs. Its size does not represent a valid approximation to the optimal position. On the contrary, if the setting is not appropriate, the inappropriate setting may make particle deviate from the original correct evolution direction, which results in particle dispersion phenomenon and slows convergence later as well as low convergence precision. From the standard particle swarm evolution equation, evolution formula of the velocity and position are superimposed according to the formula $x = vt$. We can eliminate particle velocity and get a reduced particle swarm evolution equation without velocity as shown in formula (4).

$$x_i^{t+1} = w^* x_i^t + c_1^* r_1^* (pbest - x_i^t) + c_2^* r_2^* (gbest - x_i^t) \quad (4)$$

All parameters in formula (4) correspond to the parameters in formulas (1) and (2). The reduced particle swarm evolution equation eliminates the particle velocity parameter. And the original second-order evolution equation is reduced to a first-order, which cut the process of particle analysis and guiding particle evolution.

Inertia weight w almost plays a decisive role on convergence speed and convergence precision of algorithm. The experiments show that the algorithm is more conducive to jumping out of local minimum and facilitating the global search when inertia weight w is increasing; otherwise, the algorithm is less conducive to local accurate search. This paper reduces particle swarm evolution equations, and inertia weight w directly acts on the particle position, which is different from that the w directly acts on particle velocity in standard PSO. Our approach directly impacts on the particle position that is actual problem solution.

This paper also proposes an adaptive adjustment scheme based on inertia weight w. We dynamically adjust the value of w according to the relationship between the particle fitness and the particle aggregation degree. And we can balance the search capabilities of global and local search from the premise of guaranteeing the algorithm convergence speed is acceptable.

3.2. Adaptive adjustment scheme based on inertia weight

From what has been analyzed above, the proposed adaptive adjustment scheme based on inertia weight directly impacts on particle position in the process of PSO implementation. This approach combines the particle fitness with the particle aggregation degree and divides the population into three parts according to the relationship among them, which sets inertia weight according to the divided population. The approach is shown as follows:

Definition 2:. Particle Aggregation Degree. In PSO, we suppose that the population size is N. The i-th particle fitness is f_i. Next, we calculate the current particle average fitness as f_{avg} and find out the particles whose fitness is greater than f_{avg}. Once again we calculate those particles average fitness as f'_{avg}. We set the global optimal value of population as f_g. Then current aggregation degree δ of the particle swarm is as shown in formula (5).

$$\delta = \left| f_g - f'_{avg} \right| \quad (5)$$

Here the reason that we calculate the difference δ between f_g and f'_{avg}, rather than not simply calculate the difference between f_g and f_{avg}, is that the former does not involve those poor individuals that their fitness are less than the average fitness f_{avg}. That is to say, we can avoid the adverse effects brought by poorer individuals. This can reflect more clearly convergence degree among those individuals whose current fitness is maximum, which can more accurately describe premature convergence degree of individuals in population. In addition, the calculation workload of the two formulas is equal. The experiments show that δ is a more effective indicator of quantitative assessment about population premature convergence degree. The bigger δ is, the more particles disperse; the smaller δ is, the more particles are prone to premature convergence.

According to the proposed adaptive adjustment scheme based on inertia weight, we can make full use of the average fitness including f_{avg} and f'_{avg}. The population can be divided into three parts. And we can set the corresponding inertia weight value for each part, which can make the particle position changed according to the information of particle itself in the population. This setting does not only make the population maintained diversity and good convergence speed, but also balance global and local search capabilities. The algorithm is shown as Algorithm 1.

Algorithm 1. presents the reduced adaptive particle swarm algorithm. The algorithm takes a program under test as input and