Abstract A new adaptive mutation particle swarm optimizer, which is based on the variance of the population’s fitness, is presented in this paper. During the running time, the mutation probability for the current best particle is determined by two factors: the variance of the population’s fitness and the current optimal solution. The ability of particle swarm optimization (PSO) algorithm to break away from the local optimum is greatly improved by the mutation. The experimental results show that the new algorithm not only has great advantage of convergence property over genetic algorithm and PSO, but can also avoid the premature convergence problem effectively.

Keywords particle swarm, adaptive mutation, optimization, premature convergence

1 Introduction

Particle swarm optimization (PSO) is a novel global optimization and evolutionary algorithm proposed by Dr. Eberhart and Dr. Kennedy, who were motivated by the prey behavior of birds [1,2]. As an important tool for optimization, PSO algorithm has been successfully used in system identification [3], training artificial neural network [4], etc.

Like the other global optimization algorithms (such as genetic algorithms, GA), premature convergence also exists in PSO algorithm, especially in some complex maximum point searching problems. To solve them, the main method used, at present, is to expand the swarm size. Although it improves the performance of the algorithm to a certain degree, there are defects too: First, it cannot fundamentally get over premature convergence, and second, the computation of the algorithm will be greatly increased. This paper proposes a new adaptive mutation particle swarm optimizer (AMPSO). The mutation probability for the current best particle is determined by the variance of the population’s fitness and the current optimal solution. The experimental results show that compared with GAs and the PSO algorithm, the performance of global convergence of the new method has been improved notably; also, premature convergence can be avoided effectively.

2 PSO algorithm and its premature convergence

Similar to GAs, the PSO algorithm is also based on the population (here, we call it the particle swarm) and fitness. An individual of the particle swarm (here, we call it the particle) presents a solution. Each particle has two characters: position and velocity. The target function value corresponding to the particle position can be its fitness. In the algorithm, the fitness manifests the particle’s performance. Start with a set of particles initialized, the optimum is searched by iteration processes. During each iteration, the particle is updated through tracking two “extrema”: pBest and gBest. The former is the best position found by the particle by far, the latter is the best position in the swarm at that time.

After finding the above extrema, the scheme for updating the position and velocity of each particle is shown [1,2]:

\[
v = w v + c_1 \times \text{rand} \times (p\text{Best} - \text{Present}) + c_2 \times \text{rand} \times (g\text{Best} - \text{Present}) \tag{1}
\]
and

\[ \text{Present} = \text{Present} + V, \]  

(2)

where \( V \) is the velocity of the particle, \( \text{Present} \) is its current position, and \( \text{rand} \) is a random number between \([0,1]\). \( c_1 \) and \( c_2 \) are called learning parameters; in general, \( c_1 = c_2 = 2 \). \( w \) is the weighted parameter, and its value is between 0.1 and 0.9. In [2], typical experiments showed that, if \( w \) decreases linearly with the iterations, the algorithm performance would be greatly increased. Assume that \( w_{\text{max}} \) is the maximum weighted parameter, \( w_{\text{min}} \) is the minimum weighted parameter, \( \text{run} \) is the current iteration, and \( \text{runMax} \) is the total iterations. \( w \) is thus defined:

\[ w = w_{\text{max}} - \text{run} \frac{(w_{\text{max}} - w_{\text{min}})}{\text{runMax}}. \]  

(3)

During the update, the maximum velocity of each dimension of a particle is restricted to \( v_{\text{max}} \), whose coordinate of every dimension is also restricted to the permission scope. At the same time, \( p\text{Best} \) and \( g\text{Best} \) are updated continuously during iteration processes, and the final \( g\text{Best} \) is the optimal solution of the algorithm.

Real number coding is used in PSO algorithm. Since there are no selection, crossover, and mutation, the frame of the algorithm is relatively simple, and the operation speed is very fast. However, if some particle in the swarm finds the current optimal position, the other particles will gather close to it rapidly. If the position is a local optimum, the particle swarm will not be able to search over again in the solution space. Hence, the algorithm plunges into the local optimum, and the premature convergence phenomena happens.

We have found through many tests that regardless whether it is premature convergence or global convergence, the particles in the swarm get together at a certain special position or some special positions, which is decided on by the performance of the problem and the selection of the fitness function. It will be proved in theory in the following. The positions of the particles are consistent, which equals that every particle has the same fitness. Therefore, we can track the state of the particle swarm by the total change of the fitness of all particles in the swarm. To describe in quantity the state of the particle swarm, the definitions of the variance of the population’s fitness are given in the following, as well as that of the particle’s convergence.

**Definition 1** Assume that the number of particles (swarm size) is \( n \), \( f_i \) is the fitness of the \( i \)th particle, \( f_{\text{avg}} \) is the current average fitness of the swarm, and \( \sigma^2 \) is the variance of the population’s fitness. We can define \( \sigma^2 \) as:

\[ \sigma^2 = \sum_{i=1}^{n} \left( \frac{f_i - f_{\text{avg}}}{f} \right)^2, \]  

(4)

where \( f \) is the normalized calibration factor to confine \( \sigma^2 \). The value of \( f \) is free, but two conditions should be noted: (1) after normalization, the maximum value of \(| f_i - f_{\text{avg}} |\) of the whole particle swarm is not more than 1, and (2) \( f \) changes with the different process of the evolution. In this paper, the value of \( f \) is derived from:

\[ f = \begin{cases} \max \{ | f_i - f_{\text{avg}} | \}, & \max \{ | f_i - f_{\text{avg}} | \} > 1, \\ 1, & \text{others} \end{cases} \]  

(5)

Definition 1 shows that \( \sigma^2 \) presents the convergence degree of all the particles in the swarm. A smaller \( \sigma^2 \) presents a better convergence; contrarily, the particle swarm is still in the random searching.

**Definition 2** Assume that at time \( t \), the position of a particle is \( x(t) \), and \( p \) is the arbitrary position of the whole searching space, we define the particle’s convergence [5] as:

\[ \lim_{t \to +\infty} x(t) = p. \]  

(6)

This demonstrates that, finally, the particles settle in some fixed position \( p \) in the searching space.

**Theorem 1** If PSO algorithm plunges into premature convergence or global convergence, the particles in the swarm will gather in one or some special positions, and the variance of the population’s fitness, \( \sigma^2 \), is zero.

**Proof** By Definition 2, if a particle is in convergence, it will stay in a certain fixed position \( p \). In the following, we will discuss how to find the position \( p \). From the literature [5], the following result was given through strict mathematical derivation:

\[ \lim_{t \to +\infty} x(t) = (1 - a)y + ay', \]  

(7)

where \( a = \frac{c_1}{c_1 + c_2} \), \( c_1 \) and \( c_2 \) are learning parameters in Eq. (1), \( y \) denotes the individual extremum of this particle, while \( y' \) is the global extremum of the particle swarm. If \( c_1 = c_2 = 2 \), Eq. (7) becomes:

\[ \lim_{t \to +\infty} x(t) = \frac{y + y'}{2}. \]  

(8)

Notice that \( y \) and \( y' \) are all fixed, as Eq. (7) was derived in literature (5). If Eq. (8) were true, the convergence position \( p \) would be the midpoint between the individual extremum and the global extremum. Actually, in general, \( y \)