Neurocomputing 137 (2014) 234–240

Contents lists available at ScienceDirect

Neurocomputing

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

A diversity-guided hybrid particle swarm optimization based on gradient search

Fei Han*, Qing Liu

School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China

ARTICLE INFO

Article history:
Received 2 January 2013
Received in revised form
14 March 2013
Accepted 30 March 2013
Available online 18 February 2014

Keywords:
Particle swarm optimization
Diversity
Gradient search

ABSTRACT

As an evolutionary computing technique, particle swarm optimization (PSO) has good global search ability, but it is easy to make the swarm lose its diversity and lead to premature convergence. In this paper, a diversity-guided hybrid PSO based on gradient search is proposed to improve the search ability of the swarm. The adaptive PSO is first used to search the solution till the swarm loses its diversity. Then, the search process turns to a new PSO (DGPSOGS), and the particles update their velocities with their gradient directions as well as repel each other to improve the swarm diversity. Depending on the diversity value of the swarm, the proposed hybrid method switches alternately between two PSOs. The hybrid algorithm adaptively searches local minima with random search realized by adaptive PSO and performs global search with semi-deterministic search realized by DGPSOGS, and so its search ability is improved. The experimental results show that the proposed hybrid algorithm has better convergence performance with better diversity compared to some classical PSOs.

1. Introduction

In the past decades, feedforward neural networks (FNN) have been widely used in the field of machine learning [1–3]. Gradient descent algorithms such as backpropagation (BP) are one of the most popular learning algorithms for FNN. However, these gradient descent algorithms are easy to converge to local minima as well as converge very slowly [4–7]. To decrease the likelihood of being trapped into the local minima of the error surface, some methods incorporating prior information of the involved problems were proposed [4–7]. Although these algorithms improved their generalization performance greatly, they also increased the computational complexity as well as running time. Although their performance is improved, they are local search algorithm in essence.

As a global search algorithm, particle swarm optimization (PSO) is a population-based stochastic optimization technique developed by Kennedy and Eberhart [8,9]. Since PSO is easy to implement without requiring complex evolutionary operations compared to genetic algorithm (GA) [10,11], it has been widely used in many fields such as power system optimization, process control, dynamic optimization, adaptive control and electromagnetic optimization [12,13]. Although PSO has shown good performance in solving many optimization problems, it suffers from the problem of premature convergence like most of the stochastic search techniques, particularly in multimodal optimization problems [13].

To improve the performance of PSO, many improved PSOs were proposed. Passive congregation PSO (PSOPC) introduced passive congregation for preserving swarm integrity to transfer information among individuals of the swarm [14]. PSO with a constriction (CPSO) defined a "no-hope" convergence criterion and a "rehope" method as well as one social/confidence parameter to re-initialize the swarm [15,16]. Attractive and repulsive PSO (ARPSO) alternated between phases of attraction and repulsion according to the diversity value of the swarm, which prevented premature convergence to a high degree at a rapid convergence like adaptive PSO, [17]. Two improved ARPSOs introducing a mixed phase combined with backpropagation (BP) were proposed in [18], which had good performance on function approximation and benchmark classification problems. In fuzzy adaptive PSO (FAPSO), a fuzzy system was implemented to dynamically adapt the inertia weight of PSO, which was especially useful for optimization problems with a dynamic environment [19]. To obtain better approximation performance, double search methods, which combined adaptive PSO encoding prior constraints from the approximated functions with gradient-descent-based algorithm, were proposed [20,21]. These double search methods were designed for function approximation using FNN, and they increased computational complexity.

Although random search such as adaptive PSO converges faster than GA, it is easy to converge to local minima because of losing

* Corresponding author. Tel.: +86 511 88790321 533; fax: +86 511 88780371.
E-mail addresses: hanfei@ujs.edu.cn (F. Han), qingjing_03@126.com (Q. Liu).

http://dx.doi.org/10.1016/j.neucom.2013.03.074
0925-2312

the diversity of the swarm. On the other hand, some improved PSOs, such as PSOPC, CPSO, fussy PSO and ARPSO, may improve their search ability, but they require more time to find the best solution.

In this paper, an improved hybrid particle swarm optimization termed DGHPSOGS is proposed to overcome the problem of premature convergence. The improved method takes advantages of random search and semi-deterministic search. When the diversity value of the swarm is below a predetermined threshold, the proposed diversity-guided PSO based on gradient search called DGPSOGS is used to perform search. On the other hand, the particles update their positions by the adaptive PSO (APSO) as the diversity value of the swarm is greater than the threshold. In the hybrid method, the APSO is used to perform stochastic search by attracting the particles and the DGPSOGS is used to perform semi-deterministic search as well as repel the particles to keep the diversity of the swarm in a reasonable range. Therefore, the hybrid algorithm can decrease the likelihood of premature convergence with a reasonable convergence rate.

The rest of the paper is organized as follows. Section 2 introduces some preliminaries of PSO. The proposed method is proposed in Section 3. In Section 4, experimental results and discussion for optimization problems are given to demonstrate the efficiency and effectiveness of the proposed algorithm. Finally, the concluding remarks are offered in Section 5.

2. Particle swarm optimization

2.1. Basic particle swarm optimization

PSO is an evolutionary computation technique that searches for the best solution by simulating the movement of birds in a flock [8,9]. The population of the birds is called swarm, and the members of the population are particles. Each particle represents a possible solution to the optimizing problem. During each iteration, each particle flies independently in its own direction, which is guided by its own previous best position as well as the global best position of all the particles. Assume that the dimension of the search space is D, and the swarm is $S=(X_1, X_2, X_3, \ldots, X_{Np})$; each particle represents a position in the D dimension; the position of the i-th particle in the search space can be denoted as $X_i=(x_{i1}, x_{i2}, \ldots, x_{iD})$, $i=1, 2, \ldots, N_p$, where N_p is the number of all particles. The own previous best position of the i-th particle is called *pbest* which is expressed as $P_i=(p_{i1}, p_{i2}, \ldots, p_{iD})$. The best position of all the particles is called *gbest*, which is denoted as $P_g=(p_{g1}, p_{g2}, \ldots, p_{gD})$. The velocity of the i-th particle is expressed as $V_i=(v_{i1}, v_{i2}, \ldots, v_{iD})$. According to [8,9], the basic PSO is described as

$$V_i(t+1)=V_i(t)+c1\times rand()\times(P_i(t)-X_i(t)) +c2\times rand()\times(P_g(t)-X_i(t)) \quad (1)$$

$$X_i(t+1)=X_i(t)+V_i(t+1) \quad (2)$$

where $c1$ and $c2$ are the acceleration constants with positive values; and $rand()$ is a random number ranging from 0 to 1.

To obtain better performance, adaptive particle swarm optimization (APSO) algorithm was proposed [22], and the corresponding velocity update of particles was denoted as follows:

$$V_i(t+1)=W(t)\times V_i(t)+c1\times rand()\times(P_i(t)-X_i(t))+c2 \times rand()\times(P_g(t)-X_i(t)) \quad (3)$$

where the inertia weight $W(t)$ can be computed by the following equation:

$$W(t)=W_{\max}-t\times(W_{\max}-W_{\min})/N_{pso} \quad (4)$$

In Eq. (4), W_{max}, W_{min} and N_{pso} are the initial inertial weight, the final inertial weight and the maximum iterations, respectively.

2.2. Attractive and repulsive particle swarm optimization (ARPSO)

In [17], attractive and repulsive particle swarm optimization (ARPSO), a diversity-guided method, was proposed which was described as

$$V_i(t+1)=W(t)\times V_i(t)+dir[c1\times rand()\times(P_i(t)-X_i(t)) +c2\times rand()\times(P_g(t)-X_i(t))] \quad (5)$$

where $dir=\begin{cases}-1 & diversity<d_{low}\\ 1 & diversity>d_{high}\end{cases}$.

In ARPSO, a function was proposed to calculate the diversity of the swarm as follows:

$$diversity\ ()=\frac{1}{N_p\times|L|}\times\sum_{i=1}^{N_p}\sqrt{\sum_{j=1}^{D}(p_{ij}-\overline{p_j})^2} \quad (6)$$

where $|L|$ is the length of the maximum radius of the search space; p_{ij} is the j-th component of the i-th particle and $\overline{p_j}$ is the j-th component of the average over all particles.

In the attraction phase ($dir=1$), the swarm is attracting, and consequently the diversity decreases. When the diversity drops below the lower bound, d_{low}, the swarm switches to the repulsion phase ($dir=-1$). When the diversity reaches the upper bound, d_{high}, the swarm switches back to the attraction phase. ARPSO alternates between phases of exploiting and exploring – attraction and repulsion – low diversity and high diversity and thus improves its search ability [17].

3. The proposed hybrid algorithm

3.1. The diversity-guided PSO based on gradient search (DGPSOGS)

Since the negative gradient always points in the direction of the steepest decrease in the cost function, the nearest local minimum (maybe global minimum) will be reached eventually. In the new PSO, the particles fly to the global minimum according to their gradient of the fitness function. In the case of a function with a single minimum (unimodal function) the gradient descent algorithm converges faster than stochastic search algorithms because stochastic search algorithms are a waste of computational effort doing a random search. For multimodal functions, the gradient descent algorithm will converge to the local minimum closest to the starting point [13]. When the swarm is trapped in the local minimum, the swarm will lose its diversity. To make the swarm jump out of the local minimum, the particles in the swarm should be repelled by each other, thus improving the diversity of the swarm. For quickly finding the global minimum as well as avoiding being trapped into local minimum, the particle velocity update equation in DGPSOGS is proposed as follows:

$$V_i(t+1)=W(t)\times V_i(t)+c1\times rand() \times\left(\frac{-\partial f(X_i(t))/\partial X_i(t)}{\|-\partial f(X_i(t))/\partial X_i(t)\|}\right)-c2\times rand()\times\frac{(P_g(t)-X_i(t))}{diversity(t)+\xi} \quad (7)$$

where $f()$ is the fitness function; and ξ is a predetermined small positive number for fear that the denominator is equal to zero.

The second term on the right side of Eq. (7) guides the particles to search in the negative gradient direction. The third term on the right side of Eq. (7) is inversely proportional to the diversity of the swarm, $diversity(t)$. The smaller the diversity of the swarm, the greater the value of the third term, which results in the particles repelling each other more greatly and improving the diversity of