Applied Soft Computing 23 (2014) 76–90

Contents lists available at ScienceDirect

Applied Soft Computing

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

An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space

Xuewen Xia[a,b,*], Jingnan Liu[a], Zhongbo Hu[c]

[a] National Engineering Research Center for Satellite Positioning System, Wuhan University, Wuhan 430079, China
[b] School of Software, East China Jiaotong University, Nanchang 330013, China
[c] School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China

ARTICLE INFO

Article history:
Received 1 August 2013
Received in revised form 29 April 2014
Accepted 9 June 2014
Available online 18 June 2014

Keywords:
Particle swarm optimization
Detecting strategy
Shrink search space
Local learning
Subregions
Premature convergence

ABSTRACT

To improve the performance of the standard particle swarm optimization (PSO) which suffers from premature convergence and slow convergence speed, many PSO variants introduce lots of stochastic or aimless strategies to overcome the convergence problem. However, the mutual learning between elites particles is omitted, although which might be benefit to the convergence speed and, prevent the premature convergence. In this paper, we introduce DSLPSO, which integrates three novel strategies, specifically, tabu ***d***etecting, ***s***hrinking and ***l***ocal learning strategies, into PSO to overcome the aforementioned shortcomings. In DSLPSO, search space of each dimension is divided into many equal subregions. Then the tabu detecting strategy, which has good ergodicity for search space, helps the global historical best particle to detect a more suitable subregion, and thus help it jump out of a local optimum. The shrinking strategy enables DSLPSO to take optimization in a smaller search space and obtain a higher convergence speed. In the local learning strategy, a differential between two elites particles is used to increase solution accuracy. The experimental results show that DSLPSO has a superior performance in comparison with several other participant PSOs on most of the tested functions, as well as offering faster convergence speed, higher solution accuracy and stronger reliability.

Introduction

Particle swarm optimization (PSO), which was introduced by Kennedy and Eberhart in 1995 [1,2], is a population-based optimization technique inspired by the emergent motion of swarms such as birds flocking and fish schooling. As a swarm intelligence algorithm, a PSO system shares some common characteristics with other evolutionary algorithms (EA). However, unlike EA, in which some genetic operations including selection, crossover and mutation operators are adopted to manipulate new solutions (called individuals), PSO searches for new solutions by particles' flying through the problem space according to their own experiences and neighborhoods' best experience.

During the last decades, PSO gained increasing popularity because it can obtain promising results for optimization problems by simple implementation [3]. Due to the merit, PSO and its variants have been applied in many research areas [4–7]. However, those experiments have shown that the standard PSO algorithm suffers from premature convergence when solving complex multimodal problems [8]. Furthermore, PSO is also a population-based iterative algorithm which causes PSO computationally be inefficient as measured by the number of function evaluations (FEs) required. These drawbacks have restricted wider application of PSO [9].

A major concern in PSO today is how to overcome the shortcomings and to improve its applicability. A lot of theoretical analysis have been done on PSO [10–14], and extensive studies have revealed that PSO's capability of finding a global optimum mainly depends upon its two characteristics. The first one is the ability to preserve the diversity of swarm, the aim of which is to avoid premature convergence, and thereby to enhance PSO's exploration ability. The other one is the capability of local search, which called exploiting ability, by which PSO can improve the accuracy of solutions. Aiming at the issues, many studies have been carried on during the past decades to accelerate convergence speed of PSO and to prevent the swarm falling into a local optimum [15–17], the detail of which will be discussed in Section "Related works".

Although a lot of efforts have being spent on improving these abilities, it is seen to be difficult to achieve both goals

* Corresponding author at: National Engineering Research Center for Satellite Positioning System, Wuhan University, Wuhan 430079, China.
Tel.: +86 18170871418.
E-mail address: xwxia@whu.edu.cn (X. Xia).

http://dx.doi.org/10.1016/j.asoc.2014.06.012
1568-4946/

simultaneously. For example, the comprehensive-learning PSO (CLPSO) [18] has a desirable performance on preserving the diversity, which causes it more suitable for solving multimodal problems. However, it is not the best choice to use CLPSO optimizing unimodal problems due to the slow convergence speed. On the contrary, UPSO [19] is more suitable for unimodal problems than multimodal problems. Although one algorithm cannot offer better performance than all the others on every kind of problem, these variants PSO have provided diverse effective strategies to solve different problems. Based on the state-of-art, three strategies, specifically, tabu ***d***etecting, ***s***hrinking and ***l***ocal learning are proposed to improve the performance of PSO. The novel PSO variant, called DSLPSO, is proposed in this paper.

In DSLPSO, four new features are adopted to improve its performance. First, search space of each dimension is divided into many equal subregions. Based on it, the dominant searching regions can be obtained according to the statistics of some superior particles. Second, a novel detecting strategy is adopted to attract particles to jump out of local optima traps. Third, the search space is shrunk according to the statistical information of particles' historical best position. Fourth, a simple local search process based on the differential between the global historical best particle and the second historical best particle is executed at the end of the search.

The rest of this paper is organized as follows. Section "Related works" describes the framework of standard PSO and reviews some PSO variants. The detail of DSLPSO algorithm is presented in Section "DSLPSO". Section "Experimental verification and comparisons" experimentally compares the DSLPSO with various existing PSO algorithms using a set of benchmark functions. Furthermore, in order to verify performance of DSLPSO in engineering applications, DSLPSO is also used to optimize a set of PID controller parameters. Finally, discussion and further investigations on the DSLPSO are introduced in Section "Conclusions".

Related works

PSO

Similar to other swarm intelligence heuristic algorithms, PSO is a population-based stochastic optimization technique. While optimizing a D-dimension problem, every individual in PSO is treated as a volumeless particle in a D-dimensional hyperspace; and the optimization process is described as particles' flying. During the searching process, the ith particle in PSO has a velocity vector and a current position vector which are represented as $V_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$ and $X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$, respectively. The position vector X_i is regarded as a candidate solution while the velocity vector V_i is treated as the particle's search direction and step. During the process of optimization (or flying), the ith particle decides its trajectory according to its personal historical best position vector $pbest_i = [pb_{i1}, pb_{i2}, \ldots, pb_{iD}]$ (the position giving the best objective function value or fitness value) and the global (or neighbor's) historical best position vector $gbest = [gb_1, gb_2, \ldots, gb_D]$. The update rules of a particle's velocity and position are very simple, which are defined as follows.

$$v_{ij}^{t+1} = \omega \cdot v_{ij}^t + c_1 \cdot r_1 \cdot (pb_{ij}^t - x_{ij}^t) + c_2 \cdot r_2 \cdot (gb_j^t - x_{ij}^t) \quad (1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^t \quad (2)$$

where ω is inertia weight; c_1 and c_2 are known as acceleration coefficients that determine the relative learning weight for $pbest_i$ and $gbest$, which called self-cognitive and social influence, respectively; r_1 and r_2 are two random numbers generated in the interval [0, 1] uniformly; v_{ij}^t and x_{ij}^t represent the ith particle's velocity and position on jth dimension at time t, respectively.

The inertia weight ω controls the degree that the velocity of a particle at time t influences its velocity at time t+1. As a basic parameter in PSO, ω plays an important role in regulating swarm's convergence speed and global searching ability [15,20]. Many researches have shown that a larger value of ω is beneficial to global searching ability (exploration ability) but is detrimental to the accuracy of solution (exploitation ability). On the contrary, a smaller value of ω makes PSO to obtain a more accurate solution with a very lower convergence speed. Different strategies which adapting time-varying inertia weight ω have been introduced in most PSO variants[20,21] to offsetting the contradiction between the exploration ability and the exploitation ability of PSO. For instance, a larger value of ω at the beginning of the search process can accelerate the convergence while a lesser value of ω at the end of the search process can improve the accuracy of a solution. The commonest updating rule of ω is defined as follows:

$$\omega^t = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T} \cdot t \quad (3)$$

where ω^t is the inertia weight at time t; the upper value and the lower value of ω is commonly predefined as $\omega_{max} = 0.9$ and $\omega_{min} = 0.4$, respectively; T is a maximum number of search iterations. In this case, ω^t decreases linearly from 0.9 to 0.4 during the process of optimization. Without loss of generality, the work in this paper only concentrates on PSO which utilize the same strategy for updating ω.

The flying velocity v, defined as the search step-length of a particle, characterizes the particle's ability of traversing a solution space. A larger value of v can accelerate the optimization process, but it is also very easy to omit the optimal solution. On the contrary, a smaller value of v is beneficial to improve the accuracy of solutions; in contrast, it may cause a particle to never to jump out of a local optimum. To control v within a reasonable range, a positive value V_{max} is used to clamp it within a rational range. If $|v|$ exceeds V_{max}, then it is set to $sign(v) \cdot V_{\max}$. In general, V_{max} is set within the interval of 10–20% of the search space [22]. Without loss of generality, V_{max} is set as 20% of the search space in this research.

The acceleration coefficients c_1 and c_2 determine the influence of two attractors (the personal historical best position $pbest$ and the global historical best position $gbest$ of a swarm) to a particle, respectively. A particle's flying process can be regarded as a mutation operator near the particle's historical optimal position if $c_1 \neq 0$ and $c_2 = 0$. In this condition, the primary factor that influences a particle's velocity is its own historical best position. As a result, PSO can maintain the diversity of the population and thereby avoid premature convergence. However, the speed of global convergence is very slow because of little information exchange occurs within the swarm. On the contrary, PSO can obtain a higher convergence speed if $c_1 = 0$ and $c_2 \neq 0$. The reason is that the global historical best position is the sole attractor for all particles. Nevertheless, the great attractor may cause the algorithm easily to fall into a local optimum while optimizing the multimodal functions. Previous studies have shown that c_1 and c_2 are preferably set in the interval [1.5, 2.0] [15,16].

PSO variants and improvements

There are still many deficiencies in PSO algorithm, such as low accuracy of resolution, premature convergence and curse of dimensionality, though PSO has been successfully applied in various fields, especially in some complex, large scale, nonlinear and non-differentiable optimization problems. In order to solve above-mentioned drawbacks, different strategies have been introduced to improve the performance of the standard PSO during the past decades. These modifications, which give birth to more advanced