Applied Soft Computing 29 (2015) 169–183

Contents lists available at ScienceDirect

Applied Soft Computing

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

Dynamic multi-swarm particle swarm optimizer with cooperative learning strategy

Xia Xu [a,*], Yinggan Tang [a], Junpeng Li [a], Changchun Hua [a], Xinping Guan [a,b]

[a] Institute of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China
[b] Institute of Electronic, Information, and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 1 March 2014
Received in revised form 17 August 2014
Accepted 2 December 2014
Available online 8 January 2015

Keywords:
Particle swarm optimizer
Dynamic multi-swarm particle swarm optimizer
Cooperative learning strategy

ABSTRACT

In this article, the dynamic multi-swarm particle swarm optimizer (DMS-PSO) and a new cooperative learning strategy (CLS) are hybridized to obtain DMS-PSO-CLS. DMS-PSO is a recently developed multi-swarm optimization algorithm and has strong exploration ability for the use of a novel randomly regrouping schedule. However, the frequently regrouping operation of DMS-PSO results in the deficiency of the exploitation ability. In order to achieve a good balance between the exploration and exploitation abilities, the cooperative learning strategy is hybridized to DMS-PSO, which makes information be used more effectively to generate better quality solutions. In the proposed strategy, for each sub-swarm, each dimension of the two worst particles learns from the better particle of two randomly selected sub-swarms using tournament selection strategy, so that particles can have more excellent exemplars to learn and can find the global optimum more easily. Experiments are conducted on some well-known benchmarks and the results show that DMS-PSO-CLS has a superior performance in comparison with DMS-PSO and several other popular PSO variants.

1. Introduction

Particle swarm optimization (PSO) is a population based stochastic optimization algorithm which was originally introduced by Kennedy and Eberhart [1,2]. This algorithm is motivated by the emergent motion of the foraging behavior of a flock of birds. PSO consists of a swarm of particles. Each particle represents a potential solution, which is a point in the multi-dimensional search space. The global optimum of PSO is regarded as the location of food. Each particle has a fitness value and a velocity to adjust its flying direction according to the experiences of the particle itself and its neighbors. PSO is simple in implementation and has good convergence properties when compared to evolutionary algorithms [3]. The advantages of PSO have encouraged PSO to become one of the most population optimization techniques. Now, PSO has been successfully extended to many application areas such as function optimization [4], artificial neural network training [5–7], fuzzy system control [8–11], power system [12,13] and image processing [14].

Although PSO is considered to be a robust algorithm and results in a fast convergence rate in many applications, it suffers from the premature convergence problem. PSO can be easily trapped into local optima when solving multimodal problems with a huge number of local minima. In the original PSO, the algorithm exhibits a fast-converging behavior, since all particles learn from the global best particle when updating velocities and positions. But the global best particle located at a local optimum may trap the whole swarm and lead to premature convergence [15]. For this reason, the balance between exploration (global investigation of the search place) and exploitation (the fine search around a local optimum) throughout the course of a run is a challenge to the success of PSO algorithms, especially for complex multimodal problems. In order to deal with this problem, a number of PSO variants have been developed. These approaches include tuning the control parameters such as inertia weights, social learning coefficients [16–19], designing different neighborhood topologies [20–22], hybridizing PSO with auxiliary search techniques [23–27].

Multi-swarm technique has attracted increasing attention during the last decade [28–31]. It is one of the effective methods maintaining diversity of swarm. In [28], a MSCPSO based on four sub-swarms is presented, which exchange information among themselves to evaluate overall fitness as the basis of the fitness adaptive equation. In [29], MCPSO is presented, which is a

* Corresponding author. Tel.: +86 335 8072979; fax: +86 335 8072979.
E-mail address: xuxiajnz@163.com (X. Xu).

http://dx.doi.org/10.1016/j.asoc.2014.12.026
1568-4946/

master–slave model that consists of one master swarm and several slave swarms. The master swarm updates the particle states based on both its own experience and that of the most successful particles in the slave swarms. In [30], a multi-swarm particle swarm optimization algorithm with a center learning strategy (MPSOCL) is presented with employing a new communicational scheme that each particle within one multi-swarm updates its flying direction combining historical experience from all multi-swarms with the present center position. In [31], a chaotic multi-swarm particle swarm optimization (CMS-PSO) is proposed by modifying the generic PSO with the help of the chaotic sequence for multi-dimension unknown parameter estimation and optimization by forming multiple cooperating swarms.

To consider the cooperation among sub-swarms for the multi-swarm technique, a dynamic multi-swarm particle swarm optimizer (DMS-PSO) is introduced [32]. In DMS-PSO, the whole population is divided into many small sub-swarms. These sub-swarms are regrouped frequently by using a certain regrouping schedule, and thus information can be exchanged among the sub-swarms. DMS-PSO achieves great diversity of population. However, the frequent regrouping operation results in the deficiency of exploitation. To overcome this drawback, several improvements have been made. In [33], DMS-PSO is combined with Quasi-Newton method to improve its local searching ability. In [34], the harmony search (HS) algorithm is merged into each sub-swarm of the DMS-PSO. DMS-PSO-HS can make good use of the information in past solutions more effectively. In [35], DMS-PSO is hybridized with modified multi-trajectory search (MTS) and the sub-regional HS (DMS-PSO-SHS), which makes the harmonies search in a larger potential space among different sub-populations.

Although the improvements of DMS-PSO have achieved good performance through many experiments, the cooperation among sub-swarms is just confined to the regrouping operation. No matter DMS-PSO-HS or DMS-PSO-SHS, the search capability of sub-swarms is exploited more deeply. However, it is not an efficient way to make best use of the search information for the whole swarm. For example, if some of the sub-swarms are trapped into local minima, the exchanged information among them will be the local optimal information when the regrouping operation arrives. The reason is that the sub-swarms have not cooperated with each other sufficiently. Consequently, the sub-swarms with "wrong track" may go farther with not being corrected timely. The ability of the periodic regrouping operation to exchange particles' information is constrained or inadequate. There may be a deficiency of exploration.

In order to achieve a good balance between the global exploration and the local exploitation, a new cooperative learning strategy is hybridized with DMS-PSO. The new strategy is used to exchange information among sub-swarms before the regrouping operation. For each sub-swarm, each dimension of the two worst particles learns from the better particle of two randomly selected sub-swarms using tournament selection strategy. Each particle can make full use of the best information of the sub-swarms to update its position. Meanwhile, the information among sub-swarms can be exchanged further to achieve a large search space. Owing to the further communication, DMS-PSO-CLS can make better use of the beneficial information and result in a good balance between the global exploration and the local exploitation. It is expected to bring better learning efficiency to PSO and hence better global optimization performance. Its advantages will be demonstrated by comparison with traditional PSOs and other improved PSOs.

The article is organized as follows. Standard PSO and the original DMS-PSO algorithm are introduced in Section 2. DMS-PSO-CLS is described in Section 3. Section 4 presents the benchmark functions used for experiments, experimental settings for each algorithm, and the experimental study. Conclusions are presented in Section 5.

2. Review of standard PSO and DMS-PSO

2.1. Standard PSO

Potential solution of the optimization problem can be described as a point in D-dimensional space for an optimization problem has D variables to optimize. Each particle has a velocity vector to determine its direction and a fitness value to measure its corresponding optimization state. The position and velocity in D-dimensional search space are adjusted according to the current optimal particle.

The process can be converted into a mathematical problem as follows. Suppose that sz particles are used to search the solution. The ith particle in D-dimensional space is represented as $\mathbf{x}_i = (x_i^1, x_i^2, \cdots, x_i^d, \cdots, x_i^D)$, where $x_i^d \in [x_{\min}, x_{\max}]$, $d \in [1, D]$. The velocity corresponding to the ith particle is $\mathbf{v}_i = (v_i^1, v_i^2, \cdots, v_i^d, \cdots, v_i^D)$, where $v_i^d \in [v_{\min}, v_{\max}]$. The velocity and location update strategy of the ith particle are given below:

$$v_i^d \leftarrow v_i^d + c_1 \cdot \mathrm{rand1}_i^d \cdot (pbest_i^d - x_i^d) + c_2 \cdot \mathrm{rand2}_i^d \cdot (gbest^d - x_i^d) \quad (1)$$

$$x_i^d = x_i^d + v_i^d \quad (2)$$

where c_1 and c_2 are the acceleration constants. c_1 represents the weight that the ith particle tracks its own historical optimum value $\mathbf{pbest}_i$. Figuratively speaking, it shows the understanding of itself. Similarly, c_2 represents the weight that the ith particle tracks the whole group's optimum value **gbest**. All particles use the same values c_1 and c_2. $\mathbf{pbest}_i$ and **gbest** are updated all the time according to each particle's fitness value. $\mathrm{rand1}_i^d$ and $\mathrm{rand2}_i^d$ are two random numbers in [0, 1].

To control the flying velocity, an inertia weight or a constriction factor is introduced by Shi and Eberhart [16]. It is modified to be Eq. (3).

$$v_i^d \leftarrow w \cdot v_i^d + c_1 \cdot \mathrm{rand1}_i^d \cdot (pbest_i^d - x_i^d) + c_2 \cdot \mathrm{rand2}_i^d \cdot (gbest^d - x_i^d) \quad (3)$$

where w usually decreases linearly from 0.9 to 0.4 during the iterative process [36].

Substantially, PSO is divided into two versions. The above formula is global PSO, another version is local PSO. For local PSO, each particle adjusts its position and velocity according to its historical best position $\mathbf{pbest}_i$ and the best position achieved so far from its group $\mathbf{lbest}_i$. The velocity update strategy is described as follows:

$$v_i^d \leftarrow w \cdot v_i^d + c_1 \cdot \mathrm{rand1}_i^d \cdot (pbest_i^d - x_i^d) + c_2 \cdot \mathrm{rand2}_i^d \cdot (lbest_i^d - x_i^d) \quad (4)$$

2.2. DMS-PSO

DMS-PSO is a local version of PSO with a new neighborhood topology. The population is divided into some small sized sub-swarms. They search for better positions in the search space using their own members. However, the sub-swarms are dynamic and they are regrouped frequently by using a regrouping schedule, which is a periodic exchange of information. Particles from different sub-swarms are regrouped to a new configuration through the random regrouping schedule. In this way, the search space of each small sub-swarm is expanded and better solutions are possible to be found by the new small sub-swarms [37].

Kennedy claimed that PSO with large neighborhoods would perform better on simple problems and PSO with small neighborhoods might perform better on complex problems [38]. A very small population size (e.g., 3∼5) for DMS-PSO is enough when solving relatively complex problems, which is also one of its significant features [32].