Computers & Operations Research 60 (2015) 91–110

Contents lists available at ScienceDirect

Computers & Operations Research

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

Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization

Quande Qin [a,b,c,d], Shi Cheng [e,f,*], Qingyu Zhang [a,b], Yiming Wei [c,d], Yuhui Shi [g]

[a] Department of Management Science, Shenzhen University, Shenzhen, China
[b] Research Institute of Business Analytics and Supply Chain Management, Shenzhen, China
[c] School of Management and Economics, Beijing Institute of Technology, Beijing, China
[d] Center for Energy and Environmental Policy Research, Beijing Institute of Technology, Beijing, China
[e] Division of Computer Science, The University of Nottingham Ningbo, China
[f] International Doctoral Innovation Centre, The University of Nottingham Ningbo, China
[g] Department of Electrical & Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China

ARTICLE INFO

Available online 25 February 2015

Keywords:
Global optimization
Learning strategy
Opposition-based learning
Orthogonal design
Particle swarm optimization
Economic load dispatch problems

ABSTRACT

In the canonical particle swarm optimization (PSO), each particle updates its velocity and position by taking its historical best experience and its neighbors' best experience as exemplars and adding them together. Its performance is largely dependent on the employed exemplars. However, this learning strategy in the canonical PSO is inefficient when complex problems are being optimized. In this paper, Multiple Strategies based Orthogonal Design PSO (MSODPSO) is presented, in which the social-only model or the cognition-only model is utilized in each particle's velocity update, and an orthogonal design (OD) method is used with a small probability to construct a new exemplar in each iteration. In order to enhance the efficiency of OD method and obtain more efficient exemplar, four auxiliary vector generating strategies are designed. In addition, a global best mutation operator including non-uniform mutation and Gaussian mutation is employed to improve its global search ability. The MSODPSO can be applied to PSO with the global or local structure, yielding MSODPSO-G and MSODPSO-L algorithms, respectively. To verify the effectiveness of the proposed algorithms, a set of 24 benchmark functions in 30 and 100 dimensions are utilized in experimental studies. The proposed algorithm is also tested on a real-world economic load dispatch (ELD) problem, which is modelled as a non-convex minimization problem with constraints. The experimental results on the benchmark functions and ELD problems demonstrate that the proposed MSODPSO-G and MSODPSO-L can offer high-quality solutions.

1. Introduction

An unconstrained optimization problem, without loss of generality, can be formulated as the following minimization model: $\min f(\vec{x}), \quad \vec{x} = [x_1, x_2, \ldots, x_D]^T$
where D is the number of the parameters to be optimized. Due to many real world optimization problems becoming increasingly complicated, the features of these problems such as noisy, high-dimensional, non-differentiable, discontinuous, and non-convex pose severe challenges to the traditional optimization techniques. This fact led researchers to develop various meta-heuristics algorithms, which have been demonstrated as powerful optimizers for solving complex optimization problems [1]. Particle swarm optimization (PSO) is one of the most popular swarm intelligence algorithms.

PSO was originally developed to emulate the flocking behavior of birds and fish schooling [2,3]. Since its inception in 1995, PSO has gained a lot of attentions because of its easy implementation and good performance in solving a wide spectrum of real world problems from the diverse fields of science and engineering [4–6]. However, a major problem associated with the PSO is its premature convergence which results in locally optimal solutions frequently when solving complex problems. To overcome these limitations, much effort has been made and a number of improved PSO algorithms have been developed. These improved PSO variants can be generally categorized into five groups. The first adjusts the configuration parameters to balance global and local search abilities. [7–9]. The second aims to increase diversity by designing population topologies [10,11]. The third is the PSO hybridized with other meta-heuristics optimization operators to enhance performance [12,13]. The fourth introduces multiple swarms to improve the global search ability [14,15]. The fifth

* Corresponding author at: Division of Computer Science, The University of Nottingham Ningbo, China.
E-mail addresses: qinquande@gmail.com (Q. Qin), shi.cheng@nottingham.edu.cn (S. Cheng), q.yu.zhang@gmail.com (Q. Zhang), yuhui.shi@xjtlu.edu.cn (Y. Shi).

http://dx.doi.org/10.1016/j.cor.2015.02.008
0305-0548/

designs new efficient learning strategy to overcome the drawbacks of its learning strategy [16–20].

When searching for a global optimum in a search space, PSO relies on its learning strategy to guide its search direction. Originally, each particle updates its velocity and position by taking its own historically best experience and its neighbors' best experience as exemplars, and adds them together. There are two problems in this learning strategy: one is that this learning strategy lacks a mechanism to ensure the diversity of the swarm and momentums to escape from the local optima [21]; the other is that it is not an efficient way to make the best use of the search information and a particle may suffer from that "two steps forward, one step back" phenomenon [14]. Therefore, designing new efficient learning strategies may be the most promising way to improve the performance of PSO. Nowadays, some PSO variants have been developed in this direction [16–20].

Orthogonal design (OD), an experimental design method, offers an ability to discover the best combination levels for different factors with a reasonably small number of experiments [22]. Nowadays, OD has been applied to enhance the meta-heuristics algorithms, such as genetic algorithm [23,24], differential evolution [25], artificial bee colony [26], biogeography-based optimization [27] and particle swarm optimization [20,28,29]. The OD method used in the PSO algorithm can be divided into two categories. Firstly, the OD method is applied to determine the optimal parameters combinations. The PSO algorithm takes an "update on one dimension, evaluation on all dimensions" strategy [30]. Thus, the values on some dimensions of a relatively bad solution may be closer to the optimal solution than those of a good solution. Secondly, the OD method uses the information from the search experience and constructs a more promising exemplar. An orthogonal learning strategy for PSO (OLPSO) is proposed to utilize more useful information from a particle's best experience and its neighbors' best experience via the orthogonal design [20]. This strategy can guide particles to fly in a better direction toward the optimal solution. Experimental results show that OLPSO has good performance. However, with the increase of iterations, a particle's best position and its neighbors' best position are apt to be similar. This phenomenon will reduce the efficiency of the orthogonal learning strategy in OLPSO. Moreover, OLPSO is not suitable for solving noisy problems and the cost of computation is relatively high.

In this paper, we will introduce a new OD based learning strategy and propose a Multiple Strategies based Orthogonal Design PSO (MSODPSO). Compared with the OLPSO algorithm, MSODPSO has the following new features:

1. In our proposed PSO algorithm, the OD based learning strategy involves a particle's own experience and an auxiliary vector randomly selected from four choices which were generated by different strategies. Moreover, these four auxiliary vectors have different characteristics. Theoretically, it is more effective to explore the search space than the fixed two-parent choices [23].
2. The OD based operator, which is time-consuming, may be triggered many times in one iteration in OLPSO. Our proposed algorithm will execute this operator with a small probability. This reduces the cost of computation in some degree.
3. On the basis of analyzing the characteristics of cognition-only and social-only model of PSO, we adopt a new velocity update method which combines the cognition-only model and the social-only model in this OD based learning strategy.
4. In order to enhance the global search capability, a global best mutation strategy, which hybridizes non-uniformly mutation and Gaussian mutation, is used.

Organization of the rest of this paper is as follows: Section 2 briefly introduces the basic PSO algorithm. Section 3 describes related background including the orthogonal design and opposition-based learning. The proposed algorithm, MSODPSO, is elaborated in Section 4. In Section 5, comprehensive experimental studies are conducted on 24 benchmark functions with 30-dimension and 100-dimension to verify the effectiveness of MSODPSO. In Section 6, The MSODPSO algorithm is utilized to solve a real-world economic load dispatch problem, which is a non-smooth and non-convex problem. Finally, conclusions are given in Section 7.

2. Particle swarm optimization algorithm

In PSO, each candidate solution, also called a "particle", is composed of two parts: the position and the velocity. Similar to the genetic algorithm, PSO is a derivative-free, population-based global search algorithm. The basic PSO model consists of a swarm of particles that fly through the D-dimensional search space. The ith particle is associated with two factors: a velocity vector $\vec{v}_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$ and a position vector $\vec{x}_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$, where $i \in \{1, 2, \dots, \mathrm{NP}\}$, NP is the swarm size. $x_{id} \in [l_d, u_d]$, $d \in \{1, 2, \dots, D\}$, where l_d and u_d are the lower and upper bounds of the dth dimension of the search space, respectively. The velocity and the position of each particle are initialized with random vectors within corresponding ranges. PSO finds good solutions by simply adjusting the trajectory of each particle towards its own previously best position and towards the best position of its neighbors found so far. The new velocities and the positions of the particles for the next iteration are updated according to the following two equations:

$$v_{id}^{t+1} = wv_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{n_i d}^t - x_{id}^t) \tag{1}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{2}$$

where $c_1 r_1 (p_{id}^t - x_{id}^t)$ and $c_2 r_2 (p_{n_i d}^t - x_{id}^t)$ are known as the cognition component and social component, respectively; w is the inertia weight which determines how much the previous velocity is preserved [31]; the superscript t denotes the iteration number; $\vec{p}_i = [p_{i1}, p_{i2}, \dots, p_{iD}]$ is the historically best position that has been found by particle i, $\vec{p}_{n_i} = [p_{n_i 1}, p_{n_i 2}, \dots, p_{n_i D}]$ is the historically best position that has been found by the ith particle's neighbors so far; c_1 and c_2 are acceleration coefficients; r_1 and r_2 represent two independently random numbers uniformly distributed within the interval $[0, 1]$. Generally, a maximum velocity $v_{\max}$ is specified to control excessive roaming of particles outside the user defined search space. If $|v_{id}|$ exceeds $v_{\max}$, then it is assigned to $\mathrm{sign}(v_{id}) v_{\max,d}$ [32,33].

The PSO algorithm that updates velocities using Eq. (1) is called full model. In addition, Kennedy [34] introduced two special PSO models, the cognition-only model and the social-only model, which are defined by omitting components of Eq. (1). The cognition-only model is obtained by dropping the social component in the full model. The velocity of the ith particle is changed to the following equation:

$$v_{id}^{t+1} = wv_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) \tag{3}$$

Dropping the cognition component defines the social-only model. The velocities update of social-only model uses Eq. (4)

$$v_{id}^{t+1} = wv_{id}^t + c_2 r_2 (p_{n_i d}^t - x_{id}^t) \tag{4}$$

According to the size of the neighborhood of each particle, PSO algorithms are commonly classified into global version of PSO and local version of PSO. In the global version of PSO (GPSO), a single best historical experience of the entire swarm, denoted as $\vec{p}_g$, is shared by all particles in the whole population. Thus, $\vec{p}_g$ is used to replace $\vec{p}_{n_i}$ in the global version of PSO. In a local version of PSO (LPSO), each particle in the swarm includes a limited number of social neighbors which are defined by some topological structure, such as the ring structure and the Von Neumann structure [30]. Generally, GPSO converges faster but is likely to get trapped into