Knowledge-Based Systems 139 (2018) 23–40

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Chaotic dynamic weight particle swarm optimization for numerical function optimization

Ke Chen[a], Fengyu Zhou[a,*], Aling Liu[b]

[a] School of Control Science and Engineering, Shandong University, Jinan 250061, Shandong, PR China
[b] Yuanshang Central Middle School, Laixi, Qingdao 266600, Shandong, PR China

ARTICLE INFO

Article history:
Received 9 January 2017
Revised 5 October 2017
Accepted 6 October 2017
Available online 9 October 2017

Keyword:
Particle swarm optimization
Chaotic map
Dynamic weight
Optimization

ABSTRACT

Particle swarm optimization (PSO), which is inspired by social behaviors of individuals in bird swarms, is a nature-inspired and global optimization algorithm. The PSO method is easy to implement and has shown good performance for many real-world optimization tasks. However, PSO has problems with premature convergence and easy trapping into local optimum solutions. In order to overcome these deficiencies, a chaotic dynamic weight particle swarm optimization (CDW-PSO) is proposed. In the CDW-PSO algorithm, a chaotic map and dynamic weight are introduced to modify the search process. The dynamic weight is defined as a function of the fitness. The search accuracy and performance of the CDW-PSO algorithm are verified on seventeen well-known classical benchmark functions. The experimental results show that, for almost all functions, the CDW-PSO technique has superior performance compared with other nature-inspired optimizations and well-known PSO variants. Namely, the proposed algorithm of CDW-PSO has better search performance.

1. Introduction

In recent years, optimization problems are frequently encountered in many real-world applications such as statistical physics [1], computer science [2], artificial intelligence [3], pattern recognition and information theory [4,5], etc. It is widely believed that it takes too much time to solve actual optimization problems with traditional optimization techniques, and searching for optimal solutions is extremely hard. So, there has been a growing interest in developing and investigating different optimization methods [6,7] in the past twenty years, especially meta-heuristic optimization methods such as particle swarm optimization (PSO) [8], sine cosine algorithm (SCA) [9], moth-flame optimization algorithm (MFO) [10], ant colony optimization (ACO) [11], firefly algorithm [12], and the artificial bee colony (ABC) method [13]. These optimization algorithms have been adopted by researchers and are well suited for optimizing tasks such as function optimization [14], feature selection [15,16], logic circuit design [17,18] and artificial neural networks [19,20].

With the development of nonlinear dynamics, chaotic theory has been widely applied to various domains. In this case, one of the most famous applications is the introduction of the chaotic concept into optimization algorithms [21]. At present, the chaotic concept has been successfully combined with several nature-inspired methods such as biogeography-based optimization (BBO) [22], gravitational search algorithm (GSA) [23], harmony search (HS) [24], and krill herd algorithm (KH) [25]. To date, there is no clear theory to explain the use of the chaotic sequences to replace certain parameters that change the performance of meta-heuristic optimization algorithms. However, empirical studies indicate that the chaotic theory owns a high-level of mixing ability, and thus, a chaotic map replaced the algorithm parameter. The generated solutions may have higher diversity and mobility; therefore, the chaotic map should be adopted in many studies, especially in meta-heuristic approaches.

Particle swarm optimization (PSO) is a nature-inspired and global optimization algorithm originally developed by Kennedy et al. [26], which mimics the social behaviors of individuals in bird and fish swarms. The PSO method is robust, efficient, and straightforward; it can be implemented in any computer language very easily [27–29]. Therefore, this method has been applied to search optimum values in many fields. In some research, the performance of PSO has already been compared with other similar population-based optimization techniques such as genetic algorithm (GA) [30], differential evolution (DE) algorithm [31], and the bat algorithm (BA) [32]. The results show that the original PSO can find better values.

However, basic PSO may get trapped in local optimum and premature convergence when solving complex optimization prob-

* Corresponding author.
E-mail address: zhoufengyu@sdu.edu.cn (F. Zhou).

https://doi.org/10.1016/j.knosys.2017.10.011
0950-7051/

lems [33–35]. So to improve the performance of the original PSO on complex optimization problems, many different types of topologies have been designed to improve PSO performance. Liang et al. [36] proposed comprehensive learning PSO (CLPSO), which incorporates a novel learning strategy into PSO. Compared with other PSO variants, CLPSO shows better performance in solving multimodal problems, but the convergence rate of CLPSO on most problems is much lower than other methods. Zhan et al. [37] incorporated an orthogonal learning strategy into PSO and proposed an orthogonal learning PSO (OLPSO). Compared with other PSO algorithms and some well-known evolutionary methods, the experimental results illustrate that OLPSO has higher efficiency; however, for a few problems, OLPSO is easily trapped in the local optimal. Parsopoulos and Vrahatis [38] introduced local and global variants of PSO to combine their exploration and exploitation abilities. Liu et al. [39] introduced differential evolution to PSO and proposed a hybrid method called PSO-DE. Based on grouping and reorganization techniques, Liang and Suganthan [40] proposed dynamic multi-swarm PSO to improve PSO for avoiding local optima. To overcome premature convergence, Peram et al. [41] developed the fitness-distance-ratio-based PSO (FDR-PSO). Mendes et al. [42] proposed a fully informed PSO. Ratnaweera et al. [43] introduced self-organizing and time-varying acceleration coefficients to provide the required momentum for particles to search the global optimum solution. In the above-mentioned PSO variants, it is difficult to simultaneously achieve both fast convergence speed and escape from the local optima. Undeniably, many PSO improvements can enhance the performance of PSO, but the results of these efforts are still unsatisfactory in addressing the many challenges.

In this paper, in order to strengthen PSO performance and further balance the exploration and exploitation process, a chaotic dynamic weight particle swarm optimization called CDW-PSO is proposed. In CDW-PSO, there are two major highlights. Firstly, in the chaotic map, the future is completely determined by the past. In addition, the chaotic map has the properties of ergodicity, randomness and sensitivity. It can perform downright searches at higher speed compared with stochastic searches that mainly rely on probabilities. Therefore, the chaotic map is introduced to tune the inertia weight ω. Secondly, the exploration and exploitation capabilities contradict each other. So to achieve better optimization performance, the exploration and exploitation abilities must be balanced effectively. According to the analysis of the PSO algorithm, three major changes are proposed by introducing dynamic weight, an acceleration coefficient and the best-so-far position to modify the search process. To verify the validity and feasibility of the proposed CDW-PSO method, seventeen well-known classical benchmark functions are adopted to test CDW-PSO search performance. Firstly, CDW-PSO is compared with PSO, C-PSO and DW-PSO, and the dimension is set to 30 and 50 in turn. The experimental results show that the CDW-PSO method is effective. Secondly, CDW-PSO is compared with other well-known intelligent algorithms, and the dimension is set to 30. The simulation results show that the CDW-PSO method acquires better results in relatively all classical benchmark functions tested and is more stable in most. Finally, to further verify the performance of the CDW-PSO approach, it is compared with the other excellent PSO variants while the dimension is set to 30. The results indicate that the CDW-PSO method outperforms other PSO variants for most classical benchmark functions. In summary, the proposed CDW-PSO algorithm is successful, the CDW-PSO has a faster convergence speed and better search performance, it can simultaneously find the global optimal values or close to the theoretical optima values effectively, and the algorithm can escape trapping in local minima better than other relevant algorithms for relatively all functions.

The remainder of this paper is presented as follows: the original PSO algorithm is presented in Section 2. Section 3 describes the proposed CDW-PSO algorithm, which shows better search performance and fast convergence speed. The simulation results and comparison of the methods are shown in Section 4. Finally, conclusions are drawn and future work is presented in Section 5.

2. The original PSO algorithm

Particle swarm optimization is a biological-inspired optimization inspired by social behaviors of the individuals in bird and fish swarms. This technique is very robust, straightforward, and efficient; it is population-based on a stochastic optimization algorithm. The PSO algorithm updates the positions and velocities of the members of the swarm by adaptively learning from good experiences. In the original PSO algorithm, a particle represents a potential solution. When searching in the D-dimensional space, each particle i has a position vector $\boldsymbol{X_i^d} = [x_{i1}, x_{i2}, \cdots, x_{iD}]$ and a velocity vector $\boldsymbol{V_i^d} = [v_{i1}, v_{i2}, \cdots, v_{iD}]$ to calculate its current state, where D is the dimensions of the solution space. Moreover, particle i will keep its personal previous best position vector $\boldsymbol{pbest_i} = [pbest_i^1, pbest_i^2, \cdots, pbest_i^D]$. The best position discovered by the whole population is denoted as $\boldsymbol{gbest} = [gbest^1, gbest^2, \cdots, gbest^D]$. The position $\boldsymbol{X_i^d}$ and velocity $\boldsymbol{V_i^d}$ are initialized randomly, and updates of D-dimensional of the i particle are calculated as follows:

$$V_i^d = V_i^d + c_1 \cdot r_1 \cdot \left(pbest_i^d - X_i^d\right) + c_2 \cdot r_2 \cdot \left(gbest^d - X_i^d\right) \tag{1}$$

$$X_i^d = X_i^d + V_i^d \tag{2}$$

where c_1 and c_2 are the acceleration parameters, which are set to 2.0 commonly; r_1 and r_2 are two uniform distributed values in the range [0,1].

$V_{\max}$ and $V_{\min}$ parameters may be set for the velocity values determined for each particle to control the velocity within a reasonable range. In this study, $V_{\max}$ and $V_{\min}$ are set as 10% of the upper and lower values, respectively.

Shi and Eberhart introduced an inertia weight in order to balance exploitation and exploration abilities [44]. Using inertia weight ω, Eq. (1) is modified to Eq. (3) as follows:

$$V_i^d = \omega \cdot V_i^d + c_1 \cdot r_1 \cdot (pbest_i^d - X_i^d) + c_2 \cdot r_2 \cdot (gbest^d - X_i^d) \tag{3}$$

Shi and Eberhart also proposed a linearly decreasing inertial weight within [0,1]. In [44], an update mechanism is made as follows:

$$\omega = \omega_{\max} - \frac{M_j}{M_{\max}} \cdot (\omega_{\max} - \omega_{\min}) \tag{4}$$

where M_j and $M_{\max}$ are the current iteration and maximum iteration, respectively; $\omega_{\max}$, $\omega_{\min}$ are defined by the user.

3. The CDW-PSO algorithm

The PSO, proposed in 1995, has been widely applied to many real-world optimization problems, and many different PSO variants have been introduced in the literature. However, the challenges of premature convergence and easy trapping in the local optimum solution still persist [33–35]. Therefore, a chaotic dynamic weight particle swarm optimization called CDW-PSO is proposed to improve the optimization performance of the original PSO. In the CDW-PSO algorithm, the two major highlights are described below.

Firstly, from Eq. (3) the inertia weight ω is shown to be vitally important in deciding the convergence speed. Here, ω linearly decreases with increasing iterations, reducing the convergence speed and accuracy of the PSO algorithm. In the PSO algorithm, ω does not need to continue to linearly decrease. In fact, the chaotic map has the properties of ergodicity, non-repetition and sensitivity; it can perform downright searches at higher speeds