Applied Soft Computing 29 (2015) 386–394

Contents lists available at ScienceDirect

Applied Soft Computing

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

Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight

Chengwei Yang, Wei Gao*, Nengguang Liu, Chongmin Song

The School of Civil and Environmental Engineering, The University of New South Wales, Sydney, NSW 2032, Australia

A R T I C L E I N F O

Article history:
Received 5 November 2013
Received in revised form 6 January 2015
Accepted 6 January 2015
Available online 20 January 2015

Keywords:
Low-discrepancy sequence
Particle swarm optimization
$1/\pi^2$ order nonlinear function
Nonlinearly decreasing inertial weight
Constant acceleration coefficients

A B S T R A C T

Particle swarm optimization (PSO) is a population-based stochastic optimization algorithm motivated by intelligent collective behavior of some animals such as flocks of birds or schools of fish. The most important features of the PSO are easy implementation and few adjustable parameters. A novel PSO method called LHNPSO, with low-discrepancy sequence initialized particles and high-order ($1/\pi^2$) nonlinear time-varying inertia weight and constant acceleration coefficients, is proposed in this paper. The initial population particles are generated by using the Halton sequence to fill the search space efficiently. Nonlinear functions with orders varied within big ranges are employed to adjust the inertial weight, cognitive and social parameters. Based on the sensitivity analysis of PSO performance to the changes of the orders of these nonlinear functions, $1/\pi^2$ order nonlinear function is selected to adjust the time-varying inertia weight and the two acceleration coefficients are set to be constants. A set of well-known benchmark optimization problems is then used to investigate the performance of the proposed LHNPSO algorithm and facilitate the comparison with other three types of PSO algorithms. The results show that the easily implemented LHNPSO can converge faster and give a much more accurate final solution for a variety of benchmark test functions.

1. Introduction

With the inspiration motived by the research results on the modeling and the simulations of the behavior reflected in flocks of birds, Kennedy and Eberhart proposed the particle swarm optimization (PSO) algorithm [1]. The algorithm is a stochastic population-based method and is regarded as a global search strategy. In the PSO, particles move through the problem space with a specified velocity in search of optimal solution. Each particle maintains a memory which helps it keep the track of its previous best position and the global best position. The most important advantages of PSO are that it is easy to implement and has few parameters to adjust [2]. Commonly, the adjustable parameters are inertial weight, cognitive and social parameters.

PSO algorithm has attracted considerable attention and has been used in many research areas over the past decades [3–11]. Generally, convergence speed and ability to find global optima are basic criteria for assessing the performance of an optimization method. A large number of improved PSO algorithms [12–22] have been proposed to achieve the two goals, that is, faster convergence speed and avoiding premature or local optima.

The convergence speed of a PSO algorithm is dependent on the inertia weight and two acceleration coefficients. Random [23], linear time-varying [24], nonlinear time-varying [25,26] and adaptive strategies [27–29] have been used to adjust the inertia weight to improve the convergence speed. A review and summary of various inertia weight modification mechanisms reported in literature can be found in reference [2]. Nonlinear time-varying and adaptive inertia weights normally have better performance than others. Moreover, the principle and implementation of nonlinearly decreasing inertia weights are simpler than those of adaptive ones. Currently, third and less order nonlinear functions have been investigated extensively for adjusting inertia weight, however, the performance of higher order nonlinear time-varying weight needs to be further studied.

The concept for initializing the particles of most PSO algorithms is same as that for generating random numbers in the traditional Monte-Carlo simulation method which could be regarded as the original stochastic optimization method. In contrast to traditional Monte-Carlo methods using pseudo random numbers, the quasi-Monte Carlo method produces deterministic sequences of well-chosen points that provide the best-possible spread in the change ranges of variables [30]. These deterministic sequences are

* Corresponding author. Tel.: +61 293854123; fax: +61 293856139.

http://dx.doi.org/10.1016/j.asoc.2015.01.004
1568-4946/

often referred to as low-discrepancy sequences filling the sample area efficiently and uniformly [31], which have been successfully used to solve globally optimal problems [30–33].

The major objective of this paper is to propose an improved PSO algorithm to benefit by the advantages of high-order nonlinear time-varying inertia weight and low-discrepancy sequence. This paper is organized as follows. Section 2 reviews the classical PSO and some commonly accepted improved version of PSO. In Section 3, different-order nonlinear functions, varying from 1/18 to 9, are tested to adjust the inertia weight and two acceleration coefficients, and the corresponding parametric studies on the orders are also implemented. The improved PSO with adjusted coefficients are further test with different population sizes. Section 4 presents an improved PSO method with low-discrepancy sequence initialized particles and high-order ($1/\pi^2$) nonlinear dynamic varying inertia weight (LHNPSO), and experimental results by using well-known benchmark test functions. Section 5 gives a brief conclusion about this study.

2. Review of classical PSO and its variants

2.1. PSO

PSO algorithm starts with a population of particles randomly initialized in the search space. Each particle represents a potential solution. The algorithm searches the optimal solution by moving the positions of particles in the search space. The position and velocity of the ith particle are represented by n-dimensional vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$, respectively. The particle moves its current position toward the global optimum based on two items, that is, the best position encountered by this particle and the best position visited by the whole swarm. The velocity and position of the particle are updated according to the following formulations

$$v_i(k+1) = wv_i(k) + c_1 r_1 \circ (x_i^{Pb} - x_i(k)) + c_2 r_2 \circ (x^{Gb} - x_i(k)) \quad (1)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (2)$$

where k is the iteration number, $v_i(k)$ denotes the velocity of the ith particle at the kth iteration, $x_i(k)$ represents the position vector of the ith particle at the kth iteration, vector x_i^{Pb} is the best position visited by the ith particle, vector x^{Gb} is the global best location found by the whole swarm until the current iteration. w is the inertia weight controlling the influence of the previous velocity on the current one. In this paper, the maximum velocity is 20% of the search range divided by the size of particles. c_1 is the cognitive parameter, and c_2 is the social parameter. The two acceleration coefficients and represent dependent settings which indicate the degree of confidence in the best solution found by the individual particle and by the entire swam, respectively. r_1 and r_2 are two random vectors consist of random numbers uniformly distributed in the interval [0, 1]. The symbol $\circ$ in Eq. (1) denotes the Hadamard product.

2.2. LPSO and LPSO-TVAC

The global convergence of a PSO algorithm is dependent on the degree of local/global exploration controlled by the two acceleration coefficients, meanwhile, the relative rate of convergence is affected by the inertia weight parameter. Research results have shown that for a fixed/constant inertia value there is a significant reduction in the algorithm convergent rate. In the earlier optimization stage, a large inertia weight is required in order to search the design space thoroughly. When the most promising areas of the design space have been discovered and the convergence rate starts to slow down, the inertia weight should be reduced, in order for the particles' momentum to decrease allowing them to concentrate in the best design areas. Therefore, Shi and Eberhart [34] proposed an improved PSO algorithm with a linear time-dependent value of the inertia weight (LPSO) to accomplish the aforementioned strategy

$$w(k+1) = w_{max} - \frac{w_{max} - w_{min}}{k_{max}} \cdot k \quad (3)$$

where k is the iteration number staring from iteration zero; k_{max} is the maximum number of allowable iteration; w_{max} and w_{min} are the maximum and minimum values of the inertia weight, respectively. Usually the value of the inertia weight varies between 0.4 and 0.9.

Similarly, the PSO algorithm with linearly decreasing inertia weight and time-varying acceleration coefficients (LPSO-TVAC) has been proposed by Ratnaweera et al. [35]. In the early stage, a large c_1 and small c_2 allow the particles to move around the whole search space instead of moving toward the population best. In the latter stage, a small c_1 and a large c_2 allow the particles to converge into the global optimum. c_1 and c_2 are computed by

$$c_1(k+1) = c_{1i} - \frac{c_{1i} - c_{1f}}{k_{max}} \cdot k \quad (4)$$

$$c_2(k+1) = c_{2i} - \frac{c_{2i} - c_{2f}}{k_{max}} \cdot k \quad (5)$$

where c_{1i}, c_{1f}, c_{2i} and c_{2f} are initial and final values of the cognitive and social parameters, respectively. The best reported results were achieved when $c_{1i} = c_{2f} = 2.5$ and $c_{1f} = c_{2i} = 0.5$ [20].

Although numerous improved PSO algorithms have been presented and applied in many research areas, solving optimization problems with high accuracy and rapid convergence speed is still an important task. Both LPSO and LPSO-TVAC achieve better convergence and accuracy than classical PSO, and also keep the developed formulations structurally simple and easily understandable, making these variants become handy tools to solve optimization problems in a wide range of subjects. In other words, these two variants preserve one of the major merits of PSO, easy implementation. Because of this, in this paper, we manage to further improve PSO based on the ideas from LPSO and LPSO-TVAC, which will be explained at a greater extent in the next section. PSO, LPSO and LPSO-TVAC are all adopted as the reference methods.

3. Sensitivity analysis of time-varying inertia weight and acceleration coefficients to the order of nonlinear functions

In contrast to linear time-varying inertia weight and acceleration coefficients, in this study, nonlinear functions are proposed to adjust them. Non-linear high-order function has advantage to update PSO parameters. For instance, a faster reduction of the inertia weight can be achieved in the early stage, which will increase the rate of convergence. In the surrounding area of the optimum, the reduction of the inertia weight becomes slower, which will be helpful for capturing the solution.

$$w(k+1) = w_{max} - (w_{max} - w_{min})\left(\frac{k}{k_{max}}\right)^{\alpha} \quad (6)$$

$$c_1(k+1) = c_{1i} - (c_{1i} - c_{1f})\left(\frac{k}{k_{max}}\right)^{\beta} \quad (7)$$

$$c_2(k+1) = c_{2i} - (c_{2i} - c_{2f})\left(\frac{k}{k_{max}}\right)^{\gamma} \quad (8)$$

Actually, constant and linear time-varying parameters are special cases of what are described by the above formulations. Inertia weight, cognitive and social parameters are constants when $\alpha = \beta = \gamma = 0$, and they are linearly time-varying parameters when $\alpha = \beta = \gamma = 1$.

To investigate the influences produced by α, β and γ on the performance of PSO, the values taken for them are listed in Table 1. In literature, commonly, a set of well-known nonlinear benchmark