Applied Soft Computing 28 (2015) 345–359

Contents lists available at ScienceDirect

Applied Soft Computing

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

Non-parametric particle swarm optimization for global optimization

Zahra Beheshti, Siti Mariyam Shamsuddin*

UTM Big Data Centre, Universiti Teknologi Malaysia, Skudai, 81310 Johor, Malaysia

ARTICLE INFO

Article history:
Received 30 March 2014
Received in revised form 15 October 2014
Accepted 9 December 2014
Available online 18 December 2014

Keywords:
Optimization problems
Particle swarm optimization
Global and local optimum
Non-parametric particle swarm optimization

ABSTRACT

In recent years, particle swarm optimization (PSO) has extensively applied in various optimization problems because of its simple structure. Although the PSO may find local optima or exhibit slow convergence speed when solving complex multimodal problems. Also, the algorithm requires setting several parameters, and tuning the parameters is a challenging for some optimization problems. To address these issues, an improved PSO scheme is proposed in this study. The algorithm, called non-parametric particle swarm optimization (NP-PSO) enhances the global exploration and the local exploitation in PSO without tuning any algorithmic parameter. NP-PSO combines local and global topologies with two quadratic interpolation operations to increase the search ability. Nineteen (19) unimodal and multimodal nonlinear benchmark functions are selected to compare the performance of NP-PSO with several well-known PSO algorithms. The experimental results showed that the proposed method considerably enhances the efficiency of PSO algorithm in terms of solution accuracy, convergence speed, global optimality, and algorithm reliability.

1. Introduction

PSO [1] is a population-based algorithm inspired by the social behavior of bird flocking or fish schooling. In the algorithm, a member in the swarm, particle, represents a potential solution which is a point in the search space. The global optimum is regarded as the location of food. Each particle adjusts its flying direction according to the best experiences obtained by itself and the swarm in the solution space. The algorithm has a simple concept and is easy to implement. Hence, it has received much more attention to solve real-world optimization problems [2–7], nevertheless, PSO may easily get trapped in local optima and shows a slow convergence rate when solving the complex and high dimensional multimodal objective functions [8].

A number of variant PSO algorithms have been proposed in the literature to overcome the problems. The algorithms have improved the performance of PSO in different ways using various types of topologies, selecting parameters, combining with other search techniques and so on.

A local (ring) topological structure PSO (LPSO) [9] and Von Neumann topological structure PSO (VPSO) [10] were proposed by Kennedy and Mendes to avoid trapping into local optima. According to Kennedy [9,11], PSO with a small neighborhood might have a better performance on complex problems, while PSO with a large neighborhood would perform better on simple problems. Suganthan [12] applied a dynamically adjusted neighborhood where the neighborhood of a particle gradually increases until it includes all particles. Dynamic multi-swarm PSO (DMS-PSO) [13] was suggested by Liang and Suganthan where the neighborhood of a particle gradually increases until it includes all particles. Hu and Eberhart [14] applied a dynamic neighborhood where m nearest particles in the performance space is chosen to be its new neighborhood in each generation. Mendes et al. [15] presented the fully informed particle swarm (FIPS) algorithm that uses the information of entire neighborhood to guide the particles for finding the best solution. Parsopoulos and Vrahatis combined the global and local versions together to form the unified particle swarm optimizer (UPSO) [16]. Gao et al. [17] used PSO with a stochastic search technique and chaotic opposition-based population initialization to solve complex multimodal problems. The algorithm, CSPSO, finds new solutions in the neighborhoods of the previous best positions to escape from local optima.

The fitness-distance-ratio-based PSO (FDR-PSO) was introduced by Peram et al. [18]. In the algorithm, each particle moves toward nearby particle with higher fitness value. Liang et al. [8] developed comprehensive learning particle swarm optimization (CLPSO) that focused on avoiding the local optima by encouraging each particle to learn its behavior from other particles on different dimensions.

In another research, a selection operator was firstly proposed for PSO by Angeline [19]. Other researchers applied apart from crossover [20], and mutation [21] operations from GA into PSO. An

* Corresponding author. Tel.: +60123710679.
E-mail addresses: bzahra2@live.utm.my (Z. Beheshti), mariyam@utm.my (S.M. Shamsuddin).

http://dx.doi.org/10.1016/j.asoc.2014.12.015
1568-4946/

adaptive fuzzy particle swarm optimization (AFPSO) [22] proposed to adjust the parameters in PSO based on fuzzy inferences.

Beheshti et al. proposed the median-oriented PSO (MPSO) [23] based on the information from the median particle. Also, they introduced centripetal accelerated PSO (CAPSO) [24] according to Newton's laws of motion to accelerate the learning procedure and convergence rate of optimization problems. Other variant PSO algorithms have been recently developed based on different techniques [25–28].

Although the aforementioned algorithms have obtained satisfactory results in many optimization problems; there are still some disadvantages. For example, LPSO presents a slow convergence rate in unimodal functions [23,24]. CLPSO is not a good choice for solving unimodal problems [8]. Also, the majority of the algorithms require several parameters to tune, and setting the parameters can be a challenging for optimization problems. Moreover, some of the algorithms have a better performance than the PSO but their structures are not as simple as PSO.

To overcome the drawbacks, this study introduces a non-parametric particle swarm optimization (NP-PSO) algorithm. The proposed method performs a global and local search over the search space with a fast convergence speed using two quadratic interpolation operations. There is no need to tune any algorithmic parameter in the NP-PSO algorithm. It means that all PSO parameters are removed in the proposed algorithm.

The remainder of this study is organized as follows. In Section 2, a brief overview of PSO is provided. The proposed algorithm, NP-PSO in more details is described in Section 3. In Section 4, NP-PSO is used to solve several unimodal and multimodal benchmark functions and its performance is compared with some PSO algorithms in the literature. Finally, conclusions and further research directions are presented in Section 5.

2. Particle swarm optimization (PSO)

PSO is a population-based meta-heuristic algorithm that applies two approaches of global exploration and local exploitation to find the optimum solution. The exploration is the ability of expanding search space, where the exploitation is the ability of finding the optima around a good solution. The algorithm is initialized by creating a swarm, i.e., population of particles (N), with random positions. Every particle is shown as a vector, $(\vec{X}_i, \vec{V}_i, \vec{P}_{best_i})$, in a *D-dimensional* search space where $\vec{X}_i$ and $\vec{V}_i$ are the position and velocity, respectively. $\vec{P}_{best_i}$ is the personal best position found by the ith particle:

$$\vec{X}_i = (x_i^1, x_i^2, \ldots, x_i^D) \quad \text{for} \quad i = 1, 2, \ldots, N. \tag{1}$$

$$\vec{V}_i = (v_i^1, v_i^2, \ldots, v_i^D) \quad \text{for} \quad i = 1, 2, \ldots, N. \tag{2}$$

$$\vec{P}_{best_i} = (pbest_i^1, pbest_i^2, \ldots, pbest_i^D) \quad \text{for} \quad i = 1, 2, \ldots, N. \tag{3}$$

The best position obtained by the swarm, $\vec{P}_g$, is obtained to update the next particle velocity.

$$\vec{P}_g = (p_g^1, p_g^2, \ldots, p_g^D). \tag{4}$$

Based on $\vec{P}_{best_i}$ and $\vec{P}_g$, the next velocity and position of the ith particle are computed using (5) and (6), respectively as follows:

$$v_i^d(t+1) = w \times v_i^d(t) + C_1 \times rand_1 \times (pbest_i^d(t) - x_i^d(t)) + C_2 \times rand_2 \times (p_g^d(t) - x_i^d(t)), \tag{5}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1), \tag{6}$$

where $v_i^d(t+1)$ and $v_i^d(t)$ are the next and current velocity of the ith particle respectively. w is inertia weight, C_1 and C_2 are acceleration coefficients, $rand_1$ and $rand_2$ are random numbers in the interval [0,1]. $x_i^d(t+1)$ and $x_i^d(t)$ are the next and current position of the ith particle.

Also, $|v_i^d(t+1)| < v_{\max}$ and $vmax$ is set to a constant bounded based on the search space bound. In (5), the second and the third terms are called cognition and social term, respectively. The two models applied to choose $\vec{P}_g$ are known as $\vec{P}_{gbest}$ (for global topology) and $\vec{P}_{lbest}$ (for local topology) models. In global topology, the position of each particle is affected by the best-fitness particles of the entire population in the search space; in the local model, each particle is influenced by the best-fitness particles in its neighborhood. In this study, the local topology is called LPSO.

A large value of w is more appreciate of the global exploration; while a small value facilities a local exploitation. Shi and Eberhart [29] proposed a linearly decreasing inertia weight. They also designed fuzzy methods to nonlinearly change the inertia weight [30]. Ratnaweera et al. [31] proposed the HPSO-TVAC algorithm, which used linearly time-varying acceleration coefficients. In the algorithm, a larger C_1 and a smaller C_2 set at the beginning, gradually reversing their relationship throughout the search. By analyzing the convergence behavior of the PSO, a PSO variant with a constriction factor was presented by Clerc and Kennedy [32] as follows:

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \tag{7}$$

$$\varphi = C_1 + C_2 = 4.1, \tag{8}$$

$$v_i^d(t+1) = \chi \left[v_i^d(t) + C_1 \times rand_1 \times (pbest_i^d(t) - x_i^d(t)) + C_2 \times rand_2 \times (p_g^d(t) - x_i^d(t)) \right], \tag{9}$$

where C_1 and C_2 are set to 2.05. χ is mathematically equivalent to the inertia weight (w) as Eberhart and Shi pointed out [33].

3. NP-PSO – The proposed method

NP-PSO tends to overcome the disadvantages of PSO by avoiding local optima, accelerating the convergence speed and removing algorithmic parameters. According to [23,24], PSO has shown a better performance than LPSO in unimodal problems and LPSO provides a good results in multimodal. Hence, both local and global topologies are applied in NP-PSO. Also, the search of new area is

Table 1
Dimensions, ranges, and global optimum values of test functions used in the experiments.

Test function	Dimension (n)	[Range]^n	Xopt	Fopt
$F_1(x)$	10/30	[−100,100]n	0	0
$F_2(x)$	10/30	[−10,10]n	0	0
$F_3(x)$	10/30	[−100,100]n	0	0
$F_4(x)$	10/30	[−5,5]n	1	0
$F_5(x)$	10/30	[−600,600]n	0	0
$F_6(x)$	10/30	[−5.12,5.12]n	0	0
$F_7(x)$	10/30	[−5.12,5.12]n	0	0
$F_8(x)$	10/30	[−100,100]n	0	0
$F_9(x)$	10/30	[−10,10]n	0	0
$F_{10}(x)$	10/30	[−32.767,32.767]n	0	0
$F_{11}(x)$	10/30	[−100,100]n	0	300
$F_{12}(x)$	10/30	[−100,100]n	0	0
$F_{13}(x)$	10/30	[−32.767,32.767]n	0	0
$F_{14}(x)$	10/30	[−5,5]n	1	0
$F_{15}(x)$	10/30	[−100,100]n	0	1300
$F_{16}(x)$	10/30	[−100,100]n	0	2100
$F_{17}(x)$	10/30	[−100,100]n	0	2800
$F_{18}(x)$	10/30	[−100,100]n	0	2900
$F_{19}(x)$	10/30	[−100,100]n	0	3000