Applied Soft Computing 23 (2014) 333–345

Contents lists available at ScienceDirect

Applied Soft Computing

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

A novel particle swarm optimization algorithm with Levy flight

Hüseyin Haklı *, Harun Uğuz

Department of Computer Engineering, Selcuk University, 42075 Konya, Turkey

ARTICLE INFO

Article history:
Received 5 August 2013
Received in revised form 20 June 2014
Accepted 22 June 2014
Available online 28 June 2014

Keywords:
Particle swarm optimization
Levy flight
Levy distribution
Optimization

ABSTRACT

Particle swarm optimization (PSO) is one of the well-known population-based techniques used in global optimization and many engineering problems. Despite its simplicity and efficiency, the PSO has problems as being trapped in local minima due to premature convergence and weakness of global search capability. To overcome these disadvantages, the PSO is combined with Levy flight in this study. Levy flight is a random walk determining stepsize using Levy distribution. Being used Levy flight, a more efficient search takes place in the search space thanks to the long jumps to be made by the particles. In the proposed method, a limit value is defined for each particle, and if the particles could not improve self-solutions at the end of current iteration, this limit is increased. If the limit value determined is exceeded by a particle, the particle is redistributed in the search space with Levy flight method. To get rid of local minima and improve global search capability are ensured via this distribution in the basic PSO. The performance and accuracy of the proposed method called as Levy flight particle swarm optimization (LFPSO) are examined on well-known unimodal and multimodal benchmark functions. Experimental results show that the LFPSO is clearly seen to be more successful than one of the state-of-the-art PSO (SPSO) and the other PSO variants in terms of solution quality and robustness. The results are also statistically compared, and a significant difference is observed between the SPSO and the LFPSO methods. Furthermore, the results of proposed method are also compared with the results of well-known and recent population-based optimization methods.

Introduction

In recent years, many nature-inspired algorithms have been developed to solve complex and difficult non-linear problems. One of the reasons of this is that it take too much time to solve real world problems with traditional optimization methods, and that they cannot be solved effectively. Nature-inspired algorithms which are also known as swarm intelligence algorithms have been developed through inspiration from the behaviors of the living things in the nature. For instance, artificial bee colony optimization algorithm was developed by being motivated from bee colonies [1], ant colony optimization simulates the behavior of real ants between nest and food sources [2]. Being inspired social behaviors of a flock of fishes or birds, PSO was first proposed in 1995 by Eberhart and Kennedy [3].

PSO is a robust stochastic optimization and population-based technique based on the movement and intelligence of swarms. Easy-to-perform methods and negligible parameter settings have recently made the algorithm very popular and started to be applied in many fields.

By virtue of these advantages, PSO is effectively used for function optimization [4], filter design [5,6], fuzzy PID control [7], predicting power allocation [8], feature selection [9,10], artificial neural networks [11], image segmentation [12], scheduling and sequencing problems [13,14], logic circuit design [15,16], human tremor analysis [17], other scientific, engineering problems, etc.

In order to overcome the problem that the basic PSO is lack of producing good results due to its deficiency in velocity control mechanism and to ensure the balance between exploration and exploitation, Shi and Eberhart [18] added the inertia weight to the velocity update procedure in the basic PSO. Inertia weight enabled the PSO algorithm work more effectively by ensuring the balance between global search and local search.

In the original PSO algorithm, update procedures may be performed according to the best value found by each particle until the iteration at that moment (*pbest*) and the best value found by all particles until the iteration at that moment (*gbest*). The principle behind the PSO is that each particle owns the learning ability from itself (*pbest*) and its best neighbor (*gbest*). The PSO performs velocity change through being affected by both local and global conditions. Although this circumstance, since particles resemble each other after a certain number of iterations (loss of diversity), velocity changes drop to very little values and lead to loss of global

* Corresponding author. Tel.: +90 332 223 37 28; fax: +90 332 241 06 35.
E-mail address: hhakli@selcuk.edu.tr (H. Haklı).

http://dx.doi.org/10.1016/j.asoc.2014.06.034
1568-4946/

search ability. This causes trapping of the PSO in local minima, one of its biggest problems. There are many studies in the literature aimed at preventing this problem (such as change of velocity updates or using in hybridization with other algorithms). Liang et al. [19] diversified the swarm and targeted to prevent early convergence by making velocity update using *gbest*, or particle's best or *pbests* of different particles and selecting one of them randomly instead of learning from *pbest* and *gbest* of the particles in the original PSO. Xinchao [20] changed the velocity update procedure to prevent loss of diversity, and proposed perturbed particle swarm algorithm based on the perturbed *gbest* updating strategy. In another study, different velocity update techniques were combined, and it was ensured to continue use of the technique by which update is made better [21]. Tsoulos [22] added stopping rule, similarity check and a conditional application of some local search method modifications to enhance velocity and effectiveness of PSO. Unlike the velocity update changes, the PSO that performed local search well was used in hybridization with other nature-inspired algorithms, thus, hybrid algorithms performing both global search and local search were proposed [23–25]. Many other PSO variants such as ILPSO, GPSO, Orthogonal PSO, etc. were proposed to solve the premature convergence problem of PSO.

To strengthen global search of PSO and overcome the problem of being trapped in local minima as in the above mentioned methods, PSO was combined with Levy flight in this study. A Levy flight is a class of random walk, which is generalized Brownian motion to include non-Gaussian randomly distributed step sizes for the distance moved [26]. There are many natural and artificial facts that can be depicted by Levy flight, such as fluid dynamics, earthquake analysis, the diffusion of fluorescent molecules, cooling behavior, noise, etc. [27]. Levy flight was also used by Pereyra and Hadj in the field of Ultrasound in Skin Tissue [28], and by Al-teemy [26] in Ladar Scanning. Levy flight also took an important part in many fields in computer sciences besides these fields. Levy flight is used for Internet Traffic Models by Terdik and Gyres [29], Delay and Disruption Tolerant Network by Chen [27], Multi Robot Searching procedure by Sutantyo et al. [30] and Rhee [31] utilized Leyv walk on human mobility fields.

Moreover, Levy flight that resembles food searching path of many animals like albatross, bumblebees and deer was added to nature-inspire algorithms to ensure improvement of the algorithms [32,33]. Yang and Deb [34] used Levy flight distribution to create new cuckoo in Cuckoo Search. Also, Yang [35] introduced a new version of Firefly Algorithm-FA, Levy-flight Firefly algorithm (LFA), which combined Levy-flight with the search strategy via the Firefly for improving the randomization of FA. In their Evolution Algorithm, Lee and Yao [36] created 4 different states of β parameter of Levy flight and 4 candidate solutions, and took the one that gave the best result among these candidate values and used it to perform the mutation procedure. Also, Levy flight was used as diversification tools for ant colony optimization [37–39].

In this study, the long jumps are performed through Levy distribution, and more effective use of the search space compared to the PSO is ensured. A limit value is determined for each particle, and in case the particles could not improve self-solutions as much as the limit value given, the particles are redistributed with Levy flight such that *gbest* would be affected, and being trapped in local minima is prevented. As in many studies previously conducted to improve the PSO, in the proposed method, it is ensured with random walks that PSO performs global search more effective. It is intended to be more consistent by ensuring its being affected by *gbest* while performing these random walks.

When compared the proposed method that are examined on various types of benchmark functions with the SPSO, it is observed to be effective particularly for solving multimodal functions and as the dimensions increased, and to converge earlier. The results are also statistically compared with non-parametric Wilcoxon test, and a significant difference is seen between the LFPSO and the SPSO methods.

The rest of the paper is divided as follows. In PSO and Levy Flights, original PSO algorithm and Levy flight method are presented. The proposed approach is detailed in The Proposed Algorithm LFPSO. In Experiments and Results, the experimental results and comparison of the methods are presented. In Results and Discussion provides discussion of the present work. As a final, the paper is concluded with the future works.

PSO and Levy flights

Original PSO algorithm

The PSO algorithm has been proposed through inspiration from social behaviors of the individuals in bird and fish swarms [3]. Individuals in the swarms are referred to as particles, and each particle consists of *D-dimensional* values. For a *D-dimensional* state, position and velocity expressions of particle *i* are represented as follows.

$X_i = \{X_{i1}, X_{i2}, X_{i3}, \ldots, X_{iD}\}$ and $V_i = \{V_{i1}, V_{i2}, V_{i3}, \ldots, V_{iD}\}$

Intelligent interaction among the swarm is provided with best value of each particle (*pbest*) and best value of all particles (*gbest*) until at the current iteration. For a *D-dimensional* search space, *pbest* of particle *i* is represented as $pbest = \{P_{i1}, P_{i2}, P_{i3}, \ldots, P_{iD}\}$, *gbest* is represented as $gbest = \{G_1, G_2, G_3, \ldots, G_D\}$. Since PSO will perform update procedures according to these values, *pbest* values for each particle and the *gbest* value, which is the best value for the entire swarm, should be kept. PSO consists of two stages as beginning and calculation. In the beginning stage, all particles are distributed randomly in the search space within the determined boundaries. In calculation stage, velocities and positions of the particles are updated. Velocity of a particle is calculated as follows [3]:

$$V_{i,d}^{t+1} = V_{i,d}^{t} + c_1 rand_1 (pbest_{i,d}^{t} - X_{i,d}^{t}) + c_2 rand_2 (gbest_{d}^{t} - X_{i,d}^{t}) \qquad (1)$$

where $V_{i,d}^{t+1}$ is velocity of particle *i* at iteration *t+1* with respect to the *d*th dimension, $X_{i,d}^{t}$ is position value of the *i*th particle with respect to the *d*th dimension, *c1* is cognitive weighting factor, *c2* is social weighting factors are acceleration coefficients, *r1* and *r2* values are stochastic components of the algorithm, which are in the interval [0, 1]. *c1* and *c2* values which are generally determined identical and as 2 are set as desired, and it may be ensured that particles affected more either locally or globally. While the fact that acceleration coefficients take big values causes the particles to move away from each other and separate, their taking small values causes limitation of the movements of the particles, and not being able to scan the solution space adequately [40].

V_{max} and V_{min} parameters may be set for the velocity values determined for each particle to prevent occurrence of big changes on the particles or constant limit excesses. In this study, V_{max} and V_{min} was set as 20% of the upper and lower limits.

Inertia weight was added to PSO by Shi and Eberhart in 1998 [18] to provide the balance between exploitation and exploration:

$$V_{i,d}^{t+1} = w^{t} V_{i,d}^{t} + c_1 rand_1 (pbest_{i,d}^{t} - X_{i,d}^{t}) + c_2 rand_2 (gbest_{d}^{t} - X_{i,d}^{t}) \qquad (2)$$

Inertia weight controls effect of previous velocity increases of the particles on the velocity value, and takes part in providing the balance between global search and local search. When the inertia weight takes large values, global search is more suitable and a small inertia weight facilitates local search [19]. Shi and Eberhart [18] proposed a linearly decreasing inertia weight over the course of