Computers and Mathematics with Applications 66 (2013) 122–134

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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

On the behavior and performance of chaos driven PSO algorithm with inertia weight

Michal Pluhacek [a,*], Roman Senkerik [a], Donald Davendra [b], Zuzana Kominkova Oplatkova [a], Ivan Zelinka [b]

[a] Tomas Bata University in Zlin, Faculty of Applied Informatics, Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic

[b] VŠB – Technical University of Ostrava, Department of Computer Science, Faculty of Electrical Engineering and Computer Science, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic

ARTICLE INFO

Keywords:
Evolutionary algorithms
Particle swarm
Optimization
Swarm intelligence
Chaos

ABSTRACT

In this paper, the utilization of chaos pseudorandom number generators based on three different chaotic maps to alter the behavior and overall performance of PSO algorithm is proposed. This paper presents results of testing the performance and behavior of the proposed algorithm on typical benchmark functions that represent unimodal and multimodal problems. The promising results are analyzed and discussed.

1. Introduction

The particle swarm algorithm (PSO) is one of the new and promising evolutionary optimization algorithms (EAs)—a class of soft computing methods that is inspired by nature. In recent years, various EA's have been designed and implemented with promising results in many areas of complex optimization [1–6].

More recently, some studies indicated that using chaotic number generators might improve the performance of EAs on such tasks as PID controller design [7–9] or fuzzy modeling of an experimental thermal-vacuum system [10].

Several studies have already dealt with the possibilities of integration of chaotic systems into the PSO algorithm and the performance of such algorithms [11–13]. This research extends the previous experiments [12] and investigates the impact of using different chaotic maps on the behavior of PSO algorithm especially in terms of convergence speed and premature convergence risk. Three different chaotic systems (maps) are used and their impact compared in this study.

The aim is to find a link between specific chaotic system and specific behavior of the PSO algorithm.

In the first part of this paper, the PSO algorithm is explained. The following sections are aimed at the description of the used chaotic systems and benchmark test functions. Results analysis and a conclusion follow afterwards.

2. Particle swarm optimization algorithm

The PSO algorithm is based on the natural swarm behavior of birds and fish. It was initially proposed by Eberhart and Kennedy in 1995 [1] as an alternative to other EAs, such as Genetic Algorithms (GA) [4] and Differential Evolution (DE) [5].

Each particle in the population represents a possible solution of the optimization problem defined by its cost function. In each generation, a new location (combination of cost function parameters) of the particle is calculated based on its previous location and velocity (or "velocity vector" as velocity may be different for each dimension).

* Corresponding author. Tel.: +420 576032830.
E-mail addresses: pluhacek@fai.utb.cz (M. Pluhacek), senkerik@fai.utb.cz (R. Senkerik), donald.davendra@vsb.cz (D. Davendra), oplatkova@fai.utb.cz (Z. Kominkova Oplatkova), ivan.zelinka@vsb.cz (I. Zelinka).

0898-1221/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.camwa.2013.01.016

One of the disadvantages of the PSO algorithm is the rapid acceleration of particles, which causes them to abandon the defined area of interest. For this reason, several modifications of the PSO were introduced to handle this problem. The main principles of the PSO algorithm and its modifications are detailed in [1–3,14,15].

Within this research, the chaos driven PSO strategy with inertia weight was utilized [14]. This strategy was first introduced in 1998 [14] in order to improve the local search capability of PSO. The selection of inertia weight strategy of PSO was based on numerous previous experiments [8,9,12,13]. Several modifications of inertia weight strategy are well described in [15]. In this study, linear decreasing inertia weight [14,15] is used in order to keep the algorithm simple enough to observe and highlight the influence of chaotic systems. Default values of all PSO parameters were chosen according to the recommendations given in [1–3,14,15].

Inertia weight is designed to influence the velocity of each particle differently over time [14,15]. In the beginning of the optimization process, the influence of inertia weight factor w is minimal. As the optimization continues, the value of w is decreasing, thus the velocity of each particle is decreasing, since w is always the number <1 and it multiplies the previous velocity of particle in the process of new velocity value calculation. Inertia weight modification PSO strategy has two control parameters w_{start} and w_{ends}. A new w for each generation is given by Eq. (1), where i stand for current generation number and n for the total number of generations.

$$w = w_{\text{start}} - \frac{((w_{\text{start}} - w_{\text{end}}) * i)}{n}. \tag{1}$$

A chaos driven pseudorandom number generator is used in the main PSO formula (Eq. (2)) that determines new "velocity" and thus the position of each particle in the next generation (or migration cycle).

$$v(t+1) = w \cdot v(t) + c_1 \cdot Rand \cdot (pBest - x(t)) + c_2 \cdot Rand \cdot (gBest - x(t)) \tag{2}$$

where:

$v(t+1)$—New velocity of a particle.
$v(t)$—Current velocity of a particle.
c_1, c_2—Priority factors.
pBest—Best solution found by a particle.
gBest—Best solution found in a population.
$x(t)$—Current position of a particle.
Rand—Random number, interval $\langle 0, 1\rangle$. Chaos number generator is applied only here.

The new position of a particle is then given by Eq. (3), where $x(t+1)$ is the new position:

$$x(t+1) = x(t) + v(t+1). \tag{3}$$

3. Chaotic maps

This section contains the description of discrete chaotic maps used as the chaotic pseudorandom generator for PSO. In this research, direct output iterations of the chaotic map were used for the generation of real numbers for the main PSO formula that determines new velocity, thus the position of each particle in the next generation (see Eq. (2) in Section 2). The initial concept of embedding chaotic dynamics into evolutionary algorithms is given in [16].

3.1. Dissipative standard map

The Dissipative standard map is a two-dimensional chaotic map [15]. The parameters used in this work are $b = 0.6$ and $k = 8.8$ based on previous experiments [12] and suggestions in literature [15]. The Dissipative standard map is given in Fig. 1. The map equations are given in Eqs. (4) and (5).

$$X_{n+1} = X_n + Y_{n+1} \pmod{2\pi} \tag{4}$$

$$Y_{n+1} = bY_n + k \sin X_n \pmod{2\pi}. \tag{5}$$

3.2. Lozi map

The Lozi map is a simple discrete two-dimensional chaotic map. The Lozi map is depicted in Fig. 2. The map equations are given in Eqs. (6) and (7). The parameters used in this work are: $a = 1.7$ and $b = 0.5$ as suggested in [15,17,18].

$$X_{n+1} = 1 - a|X_n| + bY_n \tag{6}$$

$$Y_{n+1} = X_n. \tag{7}$$

3.3. Arnold's Cat map

The Arnold's Cat map is a simple two-dimensional discrete system that stretches and folds points (x, y) to $(x+y, x+2y) \bmod 1$ in phase space. The map equations are given in Eqs. (8) and (9). This map uses parameter $k = 0.1$ [12,15]. The