International Journal of Innovative
Computing, Information and Control
Volume **5**, Number **4**, April **2009**

ICIC International ©2009 ISSN 1349-4198
pp. 951–960

CHAOTIC PERFORMANCE-DEPENDENT PARTICLE SWARM OPTIMIZATION

Zhihua Cui, Xingjuan Cai and Jianchao Zeng

Division of System Simulation and Computer Application
Taiyuan University of Science and Technology
Taiyuan 030024, P. R. China
cuizhihua@gmail.com
cai_xing_juan@sohu.com; zengjianchao@263.net

Received November 2007; revised April 2008

Abstract. *Performance-dependent particle swarm optimization (PDPSO) is a new variant of PSO by incorporating the differences among individuals. Different from the traditional centralized parameter settings, PDPSO established the relationship between parameter selection and particles's performance. However, due to the introduction of strong selection pressure and its biological assumption limitation, PDPSO is easily stuck into a local optimum when solving high-dimensional multi-model numerical optimization problems. To overcome this shortcoming, four different chaotic sequences are incorporated into PDPSO to enhance the exploration capability. In this new algorithm, the chaotic sequences are mapped onto one coordinate of the velocity vector of a randomly selected particle to enhance the escaping capability from a local optimum. Four famous unconstraint benchmark functions are used to test the performance. Simulation results show this variant can improve the performance significantly.*
Keywords: Chaotic sequences, Performance-dependent particle swarm optimization, Premature convergence

1. **Introduction.** As a novel population-based swarm intelligent technique, particle swarm optimization (PSO) [1][2] simulates the animal social behaviors such as birds flocking, fish schooling, etc. Due to the simple concept and ease implementation, it has gained much attention and many improvements have been proposed [3][4][5][6][7].

In a PSO system, multiple candidate solutions coexist and collaborate simultaneously. Each solution, called a "particle", flies in the problem space according to its own "experience" as well as the experience of neighboring particles.

Different from other evolutionary computation algorithms, in PSO, each particle utilizes two information index: velocity and position, to search the problem space. The velocity information predicts the next moving direction, as well as the position vector is used to detect the optimum area. In standard particle swarm optimization, the velocity vector is updated as follows:

$$v_{jk}(t+1) = wv_{jk}(t) + c_1 r_1(p_{jk}(t) - x_{jk}(t)) + c_2 r_2(p_{gk}(t) - x_{jk}(t)) \quad (1)$$

where $v_{jk}(t)$ and $x_{jk}(t)$ represent the k^{th} coordinates of velocity and position vectors of particle j at time t, respectively. $p_{jk}(t)$ means the k^{th} dimensional value of the best position vector which particle j had been found, as well as $p_{gk}(t)$ denotes the corresponding coordinate of the best position found by the whole swarm. Inertia weight w, cognitive coefficient c_1 and social coefficient c_2 are three parameters controlling the size of velocity vector. r_1 and r_2 are two random numbers generated with normal distributions within