*Engineering Optimization*
Vol. 43, No. 1, January 2011, 19–37

# Chaotic particle swarm optimization with neural network structure and its application

Y. Sun[a,b]*, Z. Wang[a,c], G. Qi[a] and B.J. van Wyk[a]

[a]*Department of Electrical Engineering, Tshwane University of Technology, French South African Technical Institute in Electronics, Pretoria 0001, South Africa;* [b]*Mathématiques et Sciences et Technologies de l'Information et de la Communication, Université Paris-Est, Paris 94010, France;* [c]*Department of Electrical and Mining Engineering, University of South Africa, Pretoria, South Africa*

(*Received 2 September 2009; final version received 20 January 2010*)

A new particle swarm optimization (PSO) algorithm having a chaotic Hopfield Neural Network (HNN) structure is proposed. Particles exhibit chaotic behaviour before converging to a stable fixed point which is determined by the best points found by the individual particles and the swarm. During the evolutionary process, the chaotic search expands the search space of individual particles. Using a chaotic system to determine particle weights helps the PSO to escape from the local extreme and find the global optimum. The algorithm is applied to some benchmark problems and a pressure vessel problem with nonlinear constraints. The results show that the proposed algorithm consistently outperforms rival algorithms by enhancing search efficiency and improving search quality.

**Keywords:** chaos; particle swarm optimization; Hopfield neural network; bifurcation; convergence

## 1. Introduction

Particle Swarm Optimization (PSO) was proposed by Kennedy *et al*. (1995). PSO is based on a metaphor of social interaction such as birds flocking or fish schooling to search a space by adjusting the trajectories of individual vectors, called 'particles', conceptualized as moving points in a multidimensional space. The random weights of the control parameters are used to cause the particles to stochastically move towards a successful region in a higher dimensional space. Particles iteratively adjust their speed and direction based on their personal best positions and the best position in the swarm. PSO has been successfully applied to optimize a wide range of problems (Hu *et al*. 2004, Huang *et al*. 2005, Clerc 2006, Nedjah 2007, Song *et al*. 2007). However, PSO algorithms are easily trapped in local sub-optimal points when applied to problems with many local extremes (Eberhart *et al*. 1998, Shi *et al*. 1998). Usually, the particle velocities need to be limited to control their trajectories. Clerc *et al*. (2002) have significantly improved the convergence tendencies of particle swarm systems by introducing a constriction coefficient and by resorting to random weights to control the search space of the particle trajectories. A large number of PSO improvements

*Corresponding author. Email: sunyanxia@gmail.com

ISSN 0305-215X print/ISSN 1029-0273 online
© 2011 Taylor & Francis
DOI: 10.1080/03052151003702604
http://www.informaworld.com

are based on combining PSO with other techniques such as chaos or genetic algorithms. Liu *et al*. (2005) and Fan *et al*. (2008) applied chaos to PSO to avoid PSO getting trapped in local minima.

PSO cannot guarantee the convergence of particle trajectories caused by random weights. It is well known that although chaos is generated by a deterministic nonlinear system it appears pseudo random. Using a chaotic system to replace the effect of random weights of the original PSO might be convenient for analysis while maintaining stochastic search properties. In this article, chaos is combined with PSO for optimization.

Based on the above analysis, a Chaotic Particle Swarm Optimization (shortened to CPSO), which introduces chaos into PSO, is proposed. It combines the ergodic ability of chaos and the fast convergence ability of PSO to increase the variety of particles in the swarm and to improve its global optimization ability through the evolution of the population. When it is applied to some benchmarks and the pressure vessel optimization problem, the particle final states converge to the optimal region and global convergence is guaranteed.

## 2. Preliminaries

Many optimization problems can be represented as the following optimization problem:

$$\begin{aligned} &f(X_i) = \min(g(X_i)), \quad X_i = [x_i^1, x_i^2, \ldots, x_i^n], \\ &\text{subject to } p_j(X_i) \leq 0,\ j = 1, 2, \ldots, k. \end{aligned} \tag{1}$$

Here $g(\cdot)$ is the objective function without constraints; $X_i(t)$ denotes the position vector of particle $i$ consisting of $n$ variables. Every candidate solution of $f(X_i)$ is called a 'particle'. $p_j(X_i)$ is the $j$th constraint. The formulation of the constraints in Equation (1) is not restrictive, since an inequality constraint of the form $p_j(X_i) > 0$ can also be represented as $-p_j(X_i) \leq 0$, and an equality constraint, $p_j(X_i) = 0$, can be represented by two inequality constraints $p_j(X_i) \leq 0$ and $-p_j(X_i) \leq 0$.

The canonical particle swarm algorithm works by iteratively searching in a region and is concerned with the best previous success of each particle, the best previous success of the particle swarm, the current position and the velocity of each particle (Kennedy *et al.* 1995). The particle searches the domain of the problem, according to

$$V_i(t+1) = \omega V_i(t) + c_1 R_1 (P_i(t) - X_i(t)) + c_2 R_2 (P_g(t) - X_i(t)), \tag{2}$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{3}$$

where $V_i = [v_i^1, v_i^2, \ldots, v_i^n]$ is the velocity of particle $i$; $X_i = [x_i^1, x_i^2, \ldots, x_i^n]$ represents the position of particle $i$; $P_i(t)$ represents the best previous position of particle $i$ (indicating the best discoveries or previous experience of particle $i$); $P_g(t)$ represents the best previous position among all particles indicating the best discovery or previous experience of the social swarm; $\omega$ is the inertia weight that controls the impact of the previous velocity of the particle on its current velocity and is sometimes adaptive (Liu *et al.* 2005); $R_1$ and $R_2$ are two random weights whose components $r_1^j$ and $r_2^j (j = 1, 2, \ldots, n)$ are chosen uniformly within the interval [0, 1] which increase the searching ability, but cannot guarantee the convergence of the particles; and $c_1$ and $c_2$ are positive constant parameters.

As every particle can be seen as a model of a single fish or a single bird, the position chosen by the particle can be regarded as a state of a neural network with a random synaptic connection. According to Equations (2) and (3), the position components of particle $i$ can be regarded as the

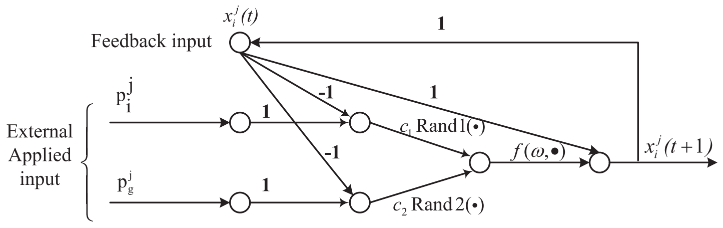

Figure 1. Particle structure.

outputs of a neural network as shown in Figure 1. In Figure 1, Rand1$(\cdot)$ and Rand2$(\cdot)$ are two independently and uniformly distributed random variables with range [0, 1], which refer to $r_1^j$ and $r_2^j$, respectively. $p_i^j$ and $p_g^j$ are components of $P_i$ and $P_g$, respectively. $f(\omega, \cdot)$ is a function of Equation (2). The particles migrate toward a new position according to Equations (2) and (3). This process is recurrent until a defined stopping criterion is met.

*Remark 1* Figure 1 has three characteristics: (1) the structure has feedback which is characteristic of the Hopfield neural network (Hopfield *et al.* 1985); (2) the structure has an externally applied input which is reminiscent of a back-propagation neural network (Rumelhart *et al.* 1986); (3) it exhibits stochastic-like chaos (Aihara *et al.* 1990).

## 3. Dynamics of the simple PSO

Clerc *et al.* (2002) reformulate Equation (2) as

$$v_i^j(t+1) = \omega v_i^j(t) + \phi(p_i' - x_i^j), \tag{4}$$

where

$$p_i' = \frac{c_1 r_1^j p_i^j + c_2 r_2^j p_g^j}{c_1 r_1^j + c_2 r_2^j}, \quad \phi = c_1 r_1^j + c_2 r_2^j, \quad (j = 1, 2, \ldots, n).$$

Setting $p_i'$ as a constant value $p$, the system reduces to

$$\left.\begin{aligned} v(t+1) &= \omega v(t) + \phi(p - x(t)),\\ x(t+1) &= x(t) + v(t+1), \end{aligned}\right\} \tag{5}$$

where $\phi$ is a constant.

If define $y(t) = p - x(t)$, Equation (5) becomes

$$\left.\begin{aligned} v(t+1) &= \omega v(t) + \phi y(t),\\ y(t+1) &= -\omega v(t) + (1-\phi) y(t), \end{aligned}\right\} \tag{6}$$

with $\omega = 1$. Equation (6) has been analysed for various values of $\phi$ and it has been proven that a non-random particle trajectory is cyclical or quasi-cyclical when $\phi \in (0, 4)$ (Clerc *et al.* 2002). The bifurcation of Equation (6) with $\omega = 1$ is shown in Figure 2. The corresponding Lyapunov exponents $\lambda_1$ and $\lambda_2$ both are 0 and the initial values of $v(t)$ and $y(t)$ have a great effect on the range of $y$. For every value of $\phi$, the values of $y$ are only recorded after 100 iterations (*i.e.* after transient behaviour has been discarded as shown in Figure 2). It can be seen from Figure 2 that the bifurcation becomes pseudo-chaotic as $\phi$ approaches 4. When a random weight is added to Equation (6), its dynamics will become more complex and it is difficult to analyse.

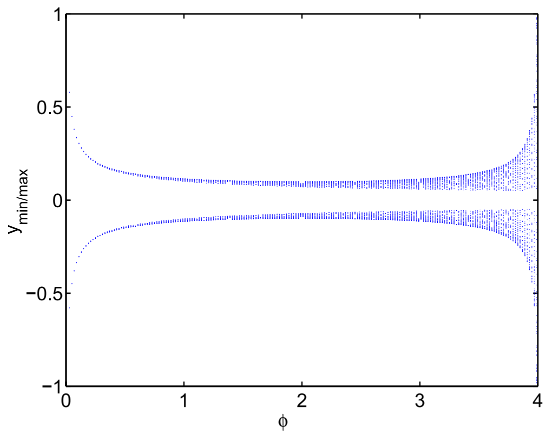

Figure 2. Bifurcation of single particle.

## 4. A model of the chaotic particles

As the PSO was discovered through the simulation of a simplified social model (Kennedy *et al*. 1995), the particle model should also be based on a social model. The sociologist Wilson (1975) made the following remark: ‘In theory at least, an individual member of a school of fish can profit from the discoveries and the previous experience of all other members of the school during the search for food. This advantage can become decisive, outweighing the disadvantages of the competition for the food items, whenever the resource is unpredictably distributed in the patches’. This means that the social sharing of information amongst the conspeciators offers an evolutionary advantage. This hypothesis was fundamental to the development of the PSO. Although human social behaviour is abstract, beliefs and attitudes are also adjusted to conform with those of social peers (Kennedy *et al*. 1995). The beneficial previous experience of the social swarm is important for optimization. There are many patterns to reflect the influences amongst particles in the swarm, and the effect is different from each other Mendes *et al*. 2004). For simplicity, only the best experience among particles is used in this article. Not only does the social influence have an effect on PSO, but also the particle’s own experience contributes information to PSO as the ‘habit’ of the particle. It would therefore be beneficial for the model of a particle to include the following two parts: the best position found by all particles and the best position of the individual particle.

Artificial neural networks are composed of simple artificial neurons mimicking biological neurons. The HNN has a property that as each neuron in an HNN updates, an energy function is monotonically reduced until the network stabilizes. One can therefore map an optimization problem to an HNN such that the cost function of the problem corresponds to the energy function of the HNN and the result of the HNN thus suggests a low cost solution to the optimization problem. The HNN might therefore be a good choice to model the particle behaviour.

Since Hopfield *et al*. (1985) applied their neural network to the travelling salesman problem, neural networks have provided a powerful approach to a wide variety of optimization problems. However, the Hopfield neural network (HNN) is often trapped in local minima. A number of modifications were proposed to make HNN escape from local minima. Some modifications are based on chaotic neural networks (Aihara *et al*. 1990) and simulated annealing (Kirkpatrick *et al*. 1983) to solve the global optimization problem (Chen *et al*. 1995). The convergence of such neural networks was discussed by Wang *et al*. (1997, 1998) and Chen *et al*. (1999).

A Hopfield network is a recurrent neural network having a synaptic connection pattern and there is an underlying Lyapunov energy function for the dynamics (Hopfield 2009). When started in any initial state, the state of the system evolves to a final state that is a (local) minimum of a Lyapunov energy function. The Lyapunov energy function decreases in a monotone fashion, and is bounded from below. Because of the existence of an elementary Lyapunov energy function for the dynamics, the only possible asymptotic result is a state on an attractor.

There are two popular forms of the model: binary neurons with discrete time that is updated one at a time, and graded neurons with continuous time. In this article, the second kind of model is used. The dynamics of an $n$-neuron continuous Hopfield neural network is described by

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{-u_i}{\tau} + \sum_j T_{ij} f_i[u_i(t)] + I_i. \tag{7}$$

The internal state of neuron $i$ for $u_i \in (-\infty, \infty)$, determines the output of neuron $i$:

$$x_i(t) = f_i[u_i(t)]. \tag{8}$$

Here:

| | |
|---|---|
| $i$ | $= 1, 2, \ldots, n$; |
| $\tau$ | is a positive constant; |
| $I_i$ | is the external input (*e.g.* sensory input or bias current) to neuron $i$ and is sometimes called the 'firing threshold' when replaced with $-I_i$; |
| $u_i$ | is the mean internal potential of the neuron which determines the output of neuron $i$; |
| $T_{ij}$ | is the strength of synaptic input from neuron $i$ to neuron $j$; |
| $f$ | is a monotone function that converts internal potential into firing rate input of the neuron; |
| $T$ | is the matrix of $T_{ij}$. |

When $T$ is symmetric, the Lyapunov energy function is given by

$$J = -\frac{1}{2}\sum_{ij} T_{ij} x_i x_j - \sum_i I_i x_i + \frac{1}{\tau}\sum_i \int_0^{x_i} f_i^{-1}(Z)\,\mathrm{d}Z, \tag{9}$$

where $f_i^{-1}$ is the inverse of the gain function $f_i$. There is a significant limiting case of this function when $T$ has no diagonal elements and the input–output relation becomes a step, going from 0 to a maximum firing rate (for convenience, scaled to 1). The third term of this Lyapunov function is then zero or infinite. With no diagonal elements in $T$, the minima of $J$ are all located at corners of the hypercube $0 \leq x_i \leq 1$. In this limit, the states of the continuous variable system are stable. This property of the HNN is discussed and incorporated into the CPSO in the remainder of this article.

In order to approach $P_g$ and $P_i$, the HNN model should include at least two neurons. For simplicity, each particle position component has two neurons whose outputs are $x_i^j(t)$ and $x_{ip}^j(t)$. In order to transform the problem into variables such that the desired optimization corresponds to the minimization of the energy function, the objective function should first be determined. As $x_i^j(t)$ and $x_{ip}^j(t)$ should approach $p_g^j$ and $p_i^j$, respectively, $(x_i^j(t) - p_g^j)^2$ and $(x_{ip}^j(t) - p_i^j)^2$ can be chosen as two parts of the energy function. The third part of the energy function, $(x_i^j(t) - x_{ip}^j(t))^2$, accompanies $(x_{ip}^j(t) - p_i^j)^2$ to cause $x_i^j(t)$ to tend towards $p_i^j$. The proposed HNN energy function is therefore given by

$$J_i^j(t) = A(x_i^j(t) - p_g^j)^2 + B(x_{ip}^j(t) - p_i^j)^2 + C(x_i^j(t) - x_{ip}^j(t))^2, \tag{10}$$

where $A$, $B$ and $C$ are positive constants.

According to Figure 1, PSO applies random weights $R_1$ and $R_2$ to simulate birds flocking or fish searching for food. An intelligent particle by implication exhibits chaos-like behaviour. Aihara *et al.* (1990) proposed a kind of chaotic neuron, which includes relative refractoriness in the model to simulate chaos in a biological brain. Wang *et al.* (1997) presented a convergence theorem for the HNN with arbitrary energy functions and discrete-time dynamics for discrete neuronal input–output functions. The theorem states that if one has a network of neurons with discrete input–output neuronal response functions, then for any change of state in any neuron $i$ the energy is guaranteed to decrease $\triangle J_i^j < 0$, if $f(\cdot)$ is a monotonically increasing function and the network is updated according to the following rules:

(1) the network is updated asynchronously, and
(2) the equations

$$x_i + \triangle x_i(t) = f(u_i(t) + \triangle u_i(t)) \tag{11}$$

$$\triangle u_i(t) = \frac{-w\triangle J_i(t)}{\triangle x_i(t)} \tag{12}$$

have non-zero solutions for $\triangle x_i^j$ and $\triangle u_i^j$ when the state of neuron $i$ is updated and remains unchanged otherwise. Here $u_i(t)$ is the input of the neurons, $x_i(t)$ is the state of the neurons, and $\omega > 0$ is the updating rate. In this article $\omega$ is set to 1. With Equations (11) and (12) satisfied, the convergence of the energy function Equation (10) is guaranteed.

For simplicity, the neuron input–output function is chosen as a linear saturation function, given by Equations (13) and (16). Equations (14) and (17) are the Euler approximation of the input of the continuous Hopfield neural network (Wang *et al.* 1998). Equations (15) and (18) are obtained by simultaneously solving Equations (10), (12), (13) and (16). The self-coupling terms $z(t)(x_i^j(t) - I_0)$ and $z(t)(x_{i_p}^j(t) - I_0)$ are added to Equations (14) and (17) to cause chaotic dynamics. The dynamics of component $j$ of particle $i$ is described by

$$x_i^j(t+1) = \begin{cases} 1 & \text{if } ku_i^j(t+1) > 1 \\ ku_i^j(t+1) & \text{if } ku_i^j(t+1) \in [-1, 1] \\ -1 & \text{if } ku_i^j(t+1) < -1, \end{cases} \tag{13}$$

$$u_i^j(t+1) = u_i^j(t) + \triangle u_i^j(t+1) - z(t)(x_i^j(t) - I_0), \tag{14}$$

$$\triangle u_i^j(t+1) = \frac{-[2A(x_i^j(t) - p_g^j) + 2C(x_i^j(t) - x_{i_p}^j(t))]}{1 + kA + kC}, \tag{15}$$

$$x_{i_p}^j(t+1) = \begin{cases} 1 & \text{if } ku_{i_p}^j(t+1) > 1 \\ ku_{i_p}^j(t+1) & \text{if } ku_{i_p}^j(t+1) \in [-1, 1] \\ -1 & \text{if } ku_{i_p}^j(t+1) < -1, \end{cases} \tag{16}$$

$$u_{i_p}^j(t+1) = u_{i_p}^j(t) + \triangle u_{i_p}^j(t+1) - z(t)(x_{i_p}^j(t) - I_0), \tag{17}$$

$$\triangle u_{i_p}^j(t+1) = \frac{-[2B(x_i^j(t) - p_i^j) - 2C(x_i^j(t) - x_{i_p}^j(t))]}{1 + kB + kC}, \tag{18}$$

$$z(t+1) = (1 - \beta)z(t). \tag{19}$$

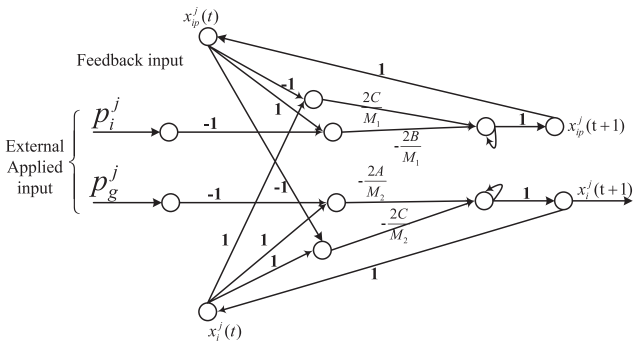

Figure 3. Structure of CPSO, where $M_1 = 1 + kB + kC$, $M_2 = 1 + kA + kC$.

Here:

$k$ is a positive constant;
$z(t)$ is self-feedback connection weight (the refractory strength);
$\beta$ is the damping factor of the time-dependent $z(t)$, $(0 \le \beta \le 1)$;
$I_0$ is a positive parameter.

According to Equations (13) and (19), the structure of the CPSO is provided as Figure 3.

Here, the asynchronous updating arrangement is used for the particle dynamic model, implying that the updating time of Equations (13) and (14) and Equations (16) and (17) is different. They use the same time indices because the particle does not distinguish between asynchronous update sequences.

According to Equations (13)–(19), the following procedure can be used to implement CPSO.

(1) Initialize the parameters of CPSO and swarm by assigning a random position in the problem hyperspace to each particle.
(2) Evaluate the fitness function for each particle.
(3) For each individual particle, compare the particle's fitness value with its $p_i^j$. If the current value is better than the $p_i^j$ value, then set this value as the $p_i$ and the current particle's position, $x_i$, as $p_i$.
(4) Identify the particle that has the best fitness value. The value of its fitness function is identified as $p_g^j$ and its position as $p_i$. When iterations are less than a certain value, reset $z = z(0)$ to keep the particles chaotic to prevent premature convergence.
(5) Asynchronously update the positions of all the particles using Equations (13)–(19) and change only one of the two states at each iteration.
(6) Repeat steps (2)–(5) until a stopping criterion is met (*e.g.* maximum number of iterations or a sufficiently good fitness value).

As can be seen from Equations (13) and (16), the particle position component $x_i^j$ is located in the interval $[-1, 1]$. The optimization problem variable interval must therefore be mapped to $[-1, 1]$ and vice versa using

$$x^j = -1 + \frac{2(x_j - a_j)}{b_j - a_j}, \quad j = 1, 2, \ldots, n \tag{20}$$

and

$$x_j = a_j + \frac{1}{2}(x^j + 1)(b_j - a_j), \quad j = 1, 2, \ldots, n. \tag{21}$$

Here, $a_j$ and $b_j$ are the lower boundary and the upper boundary of $x_j(t)$, respectively, and only one particle is analysed for simplicity.

## 5. Dynamics of CPSO

### 5.1. *Convergence of the particle swarm*

According to Equation (10), the interaction among particles in a swarm is derived from the best previous position amongst all particles. By applying asynchronous updating to analyse the particle model, the following theorem is derived.

THEOREM 5.1 *When using the energy function Equation* (10), *and the dynamics given by Equations* (13)–(19), *the particles converge on an equilibrium.*

*Proof* According to Equation (12),

$$\triangle x_i^j(t+1)\triangle u_i^j(t+1) = -\omega \triangle J_i^j(t+1). \tag{22}$$

Moreover, $\triangle x_i^j(t+1)\triangle u_i^j(t+1) > 0$ since the neuron input–output function $f$ monotonically increases. When $\triangle x_i^j(t+1) \neq 0$, the energy function $J$ monotonically decreases. When the particle state finally becomes steady, then

$$u_i^j(t+1) = u_i^j(t), \quad u_{i_p}^j(t+1) = u_{i_p}^j(t). \tag{23}$$

As $t \to \infty$,

$$z(t+1) = 0. \tag{24}$$

By substituting Equations (23) and (24) into Equations (14) and (17),

$$\triangle u_i^j(t+1) = 0, \quad \triangle u_{i_p}^j(t+1) = 0. \tag{25}$$

With Equations (15), (18) and (25) satisfied, the final convergence equilibria of the particle are

$$x_{i_e}^j(t+1) = \frac{(AB+AC)p_g^j + BCp_i^j}{AB+BC+AC}, \tag{26}$$

$$x_{ip_e}^j(t+1) = \frac{(AB+BC)p_g^j + ACp_i^j}{AB+BC+AC}. \tag{27}$$

Theorem 5.1 is therefore verified. ■

THEOREM 5.2 *The particles converge to the sphere with centre point $p_g^j$ and radius*

$$R = \frac{BC}{AB+BC+AC}(\max \|p_g^j - p_i^j\|)$$

*(if it is in a two-dimensional plane, the particles are finally in a circle).*

*Proof* According Theorem 5.1, the $i$th particle converges to the only equilibrium $X_{ie}(x_{i_e}^j(t+1), x_{ip_e}^j(t+1))$. From Equations (15) and (18), it is easy to see $p_g^j$ and $p_i^j$ determine the particle final states. The distance from $p_g^j$ to the final equilibrium $X_{ie}$ is $\| p_g^j - x_{i_e}^j(t+1)\|$. By substituting Equation (25) into $\| p_g^j - x_{i_e}^j(t+1)\|$, then

$$\| p_g^j - x_{ie}^j \| = \left\| p_g^j - \frac{(AB+AC)p_g^j + BCp_i^j}{AB+BC+AC} \right\| = \frac{BC}{AB+BC+AC} \| p_g^j - p_i^j \|. \tag{28}$$

Theorem 5.2 is therefore verified. ■

It is easy to show that the particle model given by Equations (10) and (13)–(19) has only one equilibrium as $t \to \infty$, *i.e.* $z(t) = 0$. Hence, as $t \to \infty$, $X_i$ belongs to the hyper-sphere whose origin is $P_g$ and the radius is $R$. As can be seen from Equations (26) and (27), $x_{ie}^j \approx p_g^j$ if $A \gg B$, $A \gg C$ or $(AB + AC) \gg C$.

## 5.2. *Dynamics of the simplest chaotic particle swarm*

In this section, the dynamics of the simplest particle swarm model is analysed. Equations (10) and (13)–(19) describe the dynamical model of a single particle with subscript $j$ ignored.

According to Equation (28) and Theorem 5.2, parameters $A$, $B$ and $C$ control the final convergent radius. According to trial and error, the parameters $A$, $B$ and $C$ can be chosen in the range [0.005, 0.05]. The parameter $z(0)$ controls the time of the chaotic period. If $z(0)$ is too small, the system will quickly escape from chaos and the performance will degrade. The parameter $I_0 = 0.2$ is standard in the literature on chaotic neural networks. The simulation shows that the model is not sensitive to the parameters. The values of the parameters in Equations (10) and (13)–(19) are set to

$$A = 0.02, \quad B = C = 0.01, \quad z(0) = 0.7, \quad I_0 = 0.2, \quad p_i = 0.5, \quad p_g = 0.5. \tag{29}$$

All the parameters are fixed except $z(t)$ which is varied. Figure 4 shows the time evolution of $x_i(t)$, $z(t)$ and the Lyapunov exponent $\lambda$ of $x_i(t)$. The Lyapunov exponent $\lambda$ characterizes the rate

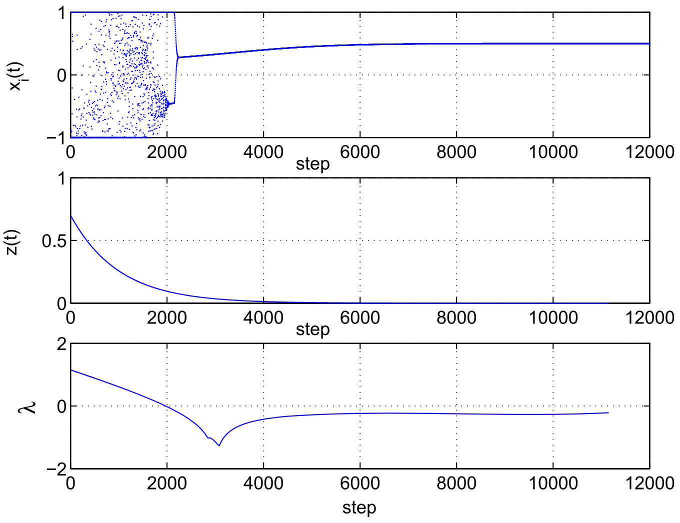

Figure 4. Time evolutions of $x_i(t)$, $z(t)$ and the Lyapunov exponent $\lambda$.

of separation of infinitesimally close trajectories. A positive Lyapunov exponent is usually taken as an indication that the system is chaotic. Here, $\lambda$ is defined as

$$\lambda = \lim_{n \to +\infty} \frac{1}{n} \sum_{t=0}^{n-1} \ln \left| \frac{\mathrm{d}x_i(t+1)}{\mathrm{d}x_i(t)} \right|. \tag{30}$$

The asynchronous updating arrangement should be changed to synchronous updating to calculate the Lyapunov exponent. At about 2000 steps, $z(t)$ decays to a small value and $x_i(t)$ departs from chaos which corresponds with the change of $\lambda$ from positive to negative. From Figure 4, the convergence process of a simple particle position follows a nonlinear bifurcation making the particle converge to a stable fixed point from a strange attractor. For the proposed CPSO model and parameters, the particle position approaches 0.5 as seen in Figure 4(a) which meets the result of Theorem 5.1.

PROPOSITION 5.1 *Since the energy function change $\triangle J$ (whether it is positive or negative) is not monotonic, the proposed chaotic particle swarm optimization model can escape from a local extremum.*

*Proof* Since the states of a single particle update asynchronously, only the state $x_i^j(t)$ can be considered without loss of generality. Referring to Equation (10),

$$\begin{aligned} \triangle J_i^j(t+1) &= J_i^j(t+1) - J_i^j(t) \\ &= A[x_i^j(t+1) - p_g^j]^2 + B[x_{ip}^j(t) - p_i^j]^2 + C[x_i^j(t+1) - x_{ip}^j(t)]^2 \\ &\quad - A[x_i^j(t) - p_g^j]^2 + B[x_{ip}^j(t) - p_i^j]^2 + C[x_i^j(t) - x_{ip}^j(t)]^2 \\ &= \triangle x_i^j(t+1)[(A+C)x_i^j(t+1) + (A+C)x_i^j(t) - 2Ap_g^j - 2Cx_{ip}^j(t)]. \end{aligned} \tag{31}$$

From Equation (15), one obtains

$$-2Ap_g^j - 2Cx_{ip}^j(t) = -(1 + kA + kC)\Delta u_i^j(t+1) - (2A + 2C)x_i^j(t). \tag{32}$$

Substituting Equation (32) into $\triangle J_i^j(t+1)$ yields

$$\begin{aligned} \triangle J_i^j(t+1) &= \Delta x_i^j(t+1)[(A+C)\Delta x_i^j(t+1) - (1 + kA + kC)\Delta x_i^j(t+1)\Delta u_i^j(t+1)] \\ &= \Delta x_i^j(t+1)\{(A+C)[k\Delta u_i^j(t+1)z(t)(x_i^j(t) - I_0)] - (1 + kA + kC)\Delta u_i^j(t+1)\} \\ &= -\Delta x_i^j(t+1)\Delta u_i^j(t+1) - (kA + kC)z(t)(x_i^j(t) - I_0)\Delta x_i^j(t+1). \end{aligned} \tag{33}$$

From Equation (33), it can be observed that

(1) when $z(t) = 0$, $\triangle J_i^j(t+1) = -\Delta x_i^j(t+1)\Delta u_i^j(t+1) < 0$;
(2) when $z(t) \neq 0$ whether $\triangle J_i^j(t+1)$ is positive or not cannot be determined, implying that the state can escape from a local extremum, completing the proof.

■

*Remark 2* The model is a deterministic HNN-based CPSO which differs from existing PSO algorithms with stochastic parameters. Its search orbits exhibit an evolutionary process of inverse period bifurcation from chaos to periodic orbits then to sink. As chaos is ergodic and the particle is in a chaotic state at the beginning (*e.g.* in Figure 4), the particle can escape when trapped in a local extremum. Note from Equation (33) that $\triangle J_i^j(t+1)$ is not always positive or negative. The proposed CPSO model will therefore in general not suffer from premature convergence problems.

## 6. Numerical simulation

To demonstrate the efficiency of the proposed technique, three famous benchmarks are chosen as test problems. The proposed CPSO, PSO (Kennedy *et al.* 1995) with parameters settings $\omega = 1, c_1 = c_2 = 2$, and standard PSO (SPSO) (Clerc 2004) with parameters settings $\omega = 0.729$, $c_1 = c_2 = 1.49445$ are applied with a population size of 20 and maximum iteration 15,000. The CPSO parameters are chosen as

$$A = 0.02, \quad B = C = 0.01, \quad \beta = 0.001, \quad z(0) = 0.7,$$
$$z(t+1) = (1-\beta)z(t), \quad k = 15, \quad I_0 = 0.2.$$

The position of every particle is initialized with a random value.

### 6.1. *Rastrigin function*

The Rastrigin function with two variables is given by

$$f(X) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \quad -1 < x_i < 1, \; i = 1, 2. \tag{34}$$

The global minimum is $-2$ and the minimum point is $(0, 0)$. There are about 20 local minima arranged in a lattice configuration for this Rastrigin function.

The time evolutions of the Rastrigin function for the proposed CPSO, PSO (Kennedy *et al.* 1995) and SPSO (Clerc 2004) are shown in Figure 5. The global minimum at $-2$ is obtained by the best particle with $(x_1, x_2) = (0, 0)$ for the proposed CPSO and the SPSO (Clerc 2004). It is clear in Figure 5 that $f(x)$ using SPSO (Clerc 2004) decays more rapidly than that from the proposed CPSO. However, there are minor oscillations caused by the stochastic influence (van dan Bergh 2001) and the main disadvantage of the SPSO is that the particles may follow wider cycles and may not converge when the individual best performance $P_i$ is far from the neighborhood's best performance $P_g$ (two different regions) (del Valle 2008). Since there are two variables in the Rastrigin function, the final convergent particle states are shown in the plane in Figure 6. In this article, '$+$' denotes the best experience of each particle and '$\circ$' denotes the final state of the particle. The '$\circ$' in Figure 6 correspond to the '$*$' for the original PSO in Figure 7. According

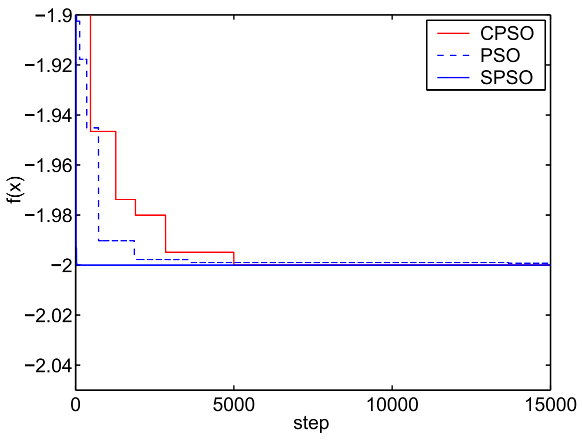

Figure 5. Time evolutions of the Rastrigin function for the proposed CPSO, PSO (Kennedy *et al.* 1995) and SPSO (Clerc 2004).

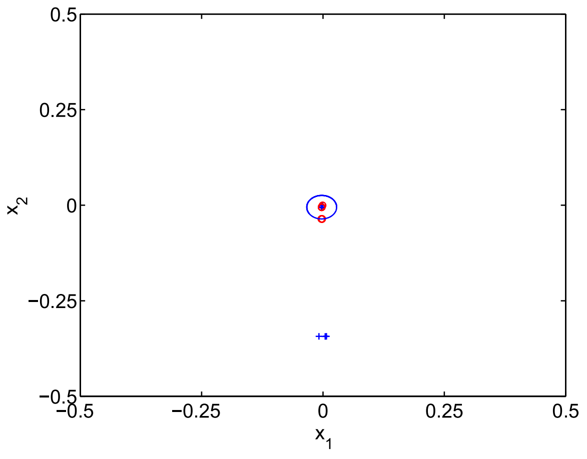

Figure 6. The particle final states and previous best experience using the proposed CPSO for the Rastrigin function (in this article, '+' denotes the best experience of each particle and '∘' denotes the final state of the particle).

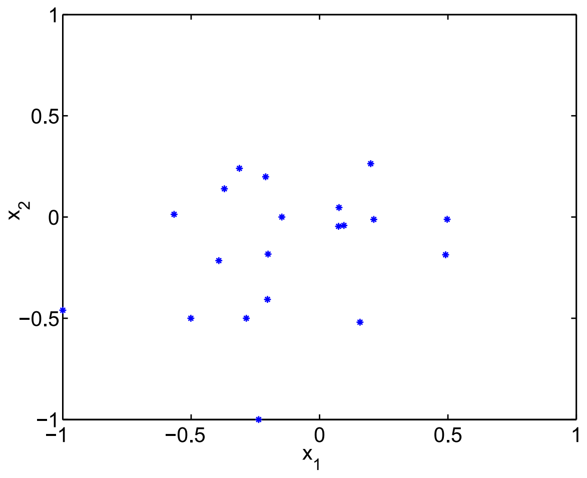

Figure 7. The particle final states achieved from the original PSO (Kennedy *et al.* 1995) for the Rastrigin function.

to Theorem 5.2 the particles converge on a circle as shown in Figure 6. The centre point is (0,0) and the radius is 0.0496. The global convergence of the particles is guaranteed for the CPSO. When the original PSO (Kennedy *et al.* 1995) is used to optimize the Rastrigin function, the final particle states are shown in Figure 7. It is easy to find that the particle final states are ruleless. By comparing the results obtained by the proposed CPSO in Figure 6 with the original PSO in Figure 7, it can be seen that the particles of the proposed CPSO are finally attracted to the best experience of all the particles and convergence is guaranteed, which is not the case for the original PSO.

The average best function value and the deviation of 50 independent runs of the proposed CPSO for the Rastrigin test function are $f(x) = -1.9983 \pm 0.0017$. By comparing the results $-1.9067 \pm 0.0930$ from CPSO (Krishna *et al.* 2006) and $-1.9979 \pm 0.0021$ using CPSO (Tang *et al.* 2009), it is seen that the result using the proposed CPSO is better and more stable than those using the other two CPSO algorithms. Moreover, the proposed CPSO can guarantee the convergence of the particle final states.

### 6.2. *The Branin function*

The Branin function is

$$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10, \tag{35}$$

where

$$-5 \le x_1 \le 10, \quad 0 \le x_2 \le 15.$$

The global minimum is approximately 0.398 and it is reached at three points (−3.142,12.275), (3.142, 2.275) and (9.425,2.425).

The time evolutions of the Branin function for the proposed CPSO, PSO (Kennedy *et al.* 1985) and SPSO (Clerc 2004) are shown in Figure 8. The proposed CPSO and SPSO (Clerc 2004) can achieve good results according to Figure 8.

Figure 9 shows the particle final states and previous best experience using CPSO. The global minimum is approximately 0.398 at the points (3.137, 2.274), (9.419, 2.495) and (−3.148, 12.283).

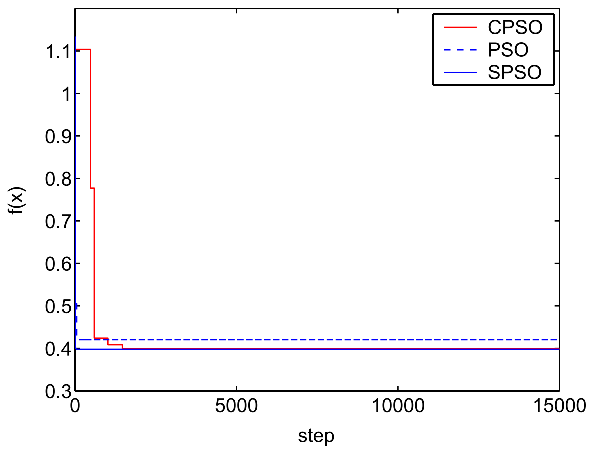

Figure 8. Time evolutions of the Branin function for the proposed CPSO, PSO (Kennedy *et al.* 1995) and SPSO (Clerc 2004).

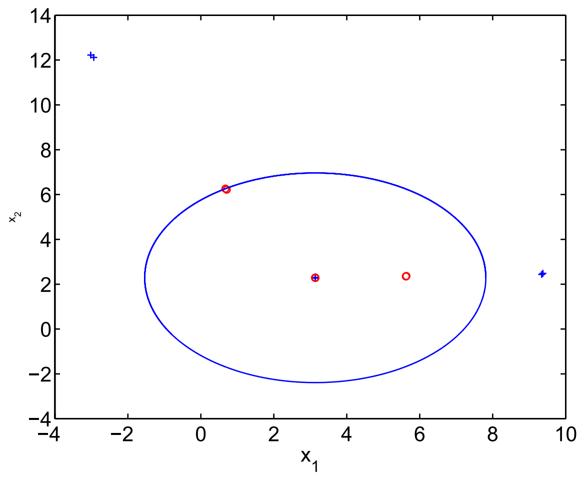

Figure 9. The particle final states and previous best experience using the proposed CPSO for the Branin function.

All the ∘'s converge in or on the circle as shown in Figure 9. The centre point is (3.137, 2.274) and the radius is 4.6751. The global convergence of the particles is guaranteed.

The average best function value and the deviation of 50 independent runs of the proposed CPSO for the Branin test function are $f(x) = 0.4006 \pm 0.00256$. Comparing the results $0.4076 \pm 0.0097$ from CPSO (Krishna *et al.* 2006), $0.4019 \pm 0.0038$ from CPSO (Tang *et al.* 2009), $0.4960 \pm 0.3703$ from PSO (Kennedy *et al.* 2001), and $0.4021 \pm 0.0153$ from GA (Goldenberg 1999), the best result from the proposed CPSO is better than those from the other algorithms.

## 6.3. *The Hartmann function*

The Hartmann function when $n = 3, 6$; $q = 4$ is given by

$$f(x) = -\sum_{i=1}^{q} c_i \exp\left[-\sum_{j=1}^{n} a_{ij}(x_j - p_{ij})^2\right], \tag{36}$$

where

$$S = \{x \in R^n | 0 \le x_j \le 1, 1 \le j \le n\}$$

and $S$ is the range of the parameter $x_j$. Tables 1 and 2 show the parameter values for the Hartmann function when $n = 3$, $q = 4$ and $n = 6$, $q = 4$, respectively. When $n = 3$, $X_{\min} = (0.114, 0.556, 0.882)$, $f(x_{\min}) = -3.86$. When $n = 6$, $X_{\min} = (0.201, 0.150, 0.477, 0.275, 0.311, 0.657)$, $f(x_{\min}) = -3.32$.

Figures 10(a) and 10(b) show the time evolutions of the Hartmann function when $n = 3, q = 4$ and $n = 6$, $q = 4$, respectively, for the proposed CPSO, PSO (Kennedy *et al.* 1995) and SPSO (Clerc 2004). From Figure 10, it can be found that the proposed CPSO and SPSO (Clerc 2004) achieve good performance. All particle final states that are denoted by '∘' are in or on the circle as shown in Figures 11(a) and 11(b). The centre point is (0.1146, 0.5557, 0.8525) and radius is 0.0821 in Figure 11(a) and the centre point is (0.2015, 0.1498, 0.4769, 0.2752, 0.3118, 0.6484) and the radius is 0.3466 in Figure 11(b). The particle final states are all in or on the circle.

The average best function value and the deviation result from CPSO, CPSO (Krishna *et al.* 2006), CPSO (Tang *et al.* 2009), PSO (Kennedy *et al.* 2001) and GA (Goldenberg 1999) are $-3.8615 \pm 0.0013$, $-3.7621 \pm 0.1000$, $-3.8589 \pm 0.0021$, $-3.8572 \pm 0.0035$

Table 1. The parameters of the Hartmann function (when $n = 3$ and $q = 4$).

| $i$ | $a_{i1}$ | $a_{i2}$ | $a_{i3}$ | $c_i$ | $p_{i1}$ | $p_{i2}$ | $p_{i3}$ |
|---|---|---|---|---|---|---|---|
| 1 | 3 | 10 | 30 | 1 | 0.3689 | 0.1170 | 0.2673 |
| 2 | 0.1 | 10 | 35 | 1.2 | 0.4699 | 0.4387 | 0.7470 |
| 3 | 3 | 10 | 30 | 3 | 0.1091 | 0.8732 | 0.5547 |
| 4 | 0.1 | 10 | 35 | 3.2 | 0.03815 | 0.5473 | 0.8828 |

Table 2. The parameters of the Hartmann function (when $n = 6$ and $q = 4$).

| $i$ | $a_{i1}$ | $a_{i2}$ | $a_{i3}$ | $a_{i4}$ | $a_{i5}$ | $a_{i6}$ | $c_i$ | $p_{i1}$ | $p_{i2}$ | $p_{i3}$ | $p_{i4}$ | $p_{i5}$ | $p_{i6}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 3 | 17 | 3.5 | 1.7 | 8 | 1 | 0.1312 | 0.1619 | 0.5569 | 0.0124 | 0.8283 | 0.5886 |
| 2 | 0.05 | 10 | 17 | 0.1 | 8 | 14 | 1.2 | 0.2329 | 0.4135 | 0.8307 | 0.3736 | 0.1004 | 0.9991 |
| 3 | 3 | 3.5 | 1.7 | 10 | 17 | 8 | 3 | 0.2348 | 0.1451 | 0.3522 | 0.2883 | 0.3047 | 0.6550 |
| 4 | 17 | 8 | 0.05 | 10 | 0.1 | 14 | 3.2 | 0.4047 | 0.8828 | 0.8732 | 0.5743 | 0.1091 | 0.0381 |

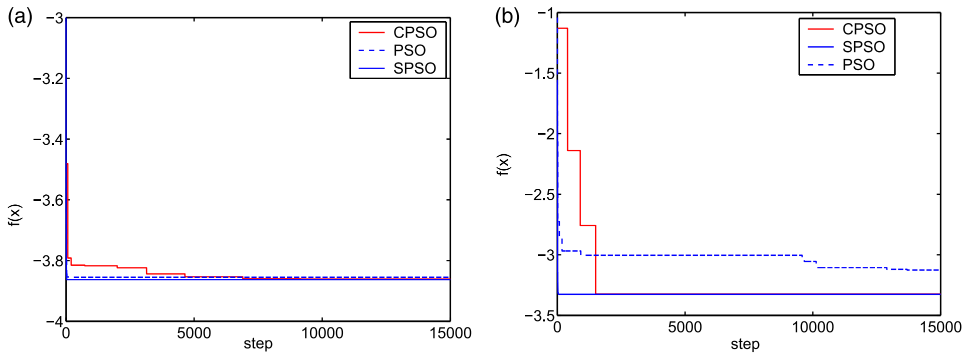

Figure 10. Time evolutions of the Hartmann function for the proposed CPSO, PSO (Kennedy *et al.* 1995), and SPSO (Clerc 2004): (a) when $n = 3$ and $q = 4$; and (b) when $n = 6$ and $q = 4$.

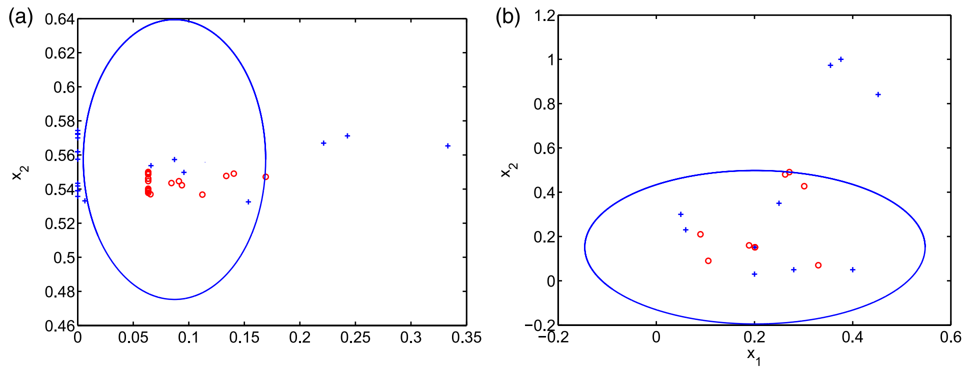

Figure 11. The particle final states and previous best experience using the proposed CPSO for the Hartmann function: (a) when $n = 3$ and $q = 4$; and (b) when $n = 6$ and $q = 4$.

and $-3.8571 \pm 0.0070$ when $n = 3$, $q = 4$, and $-3.1798 \pm 0.1459$, $-2.3375 \pm 0.8750$, $-2.8949 \pm 0.4301$, $-2.8943 \pm 0.3995$ and $-3.0212 \pm 0.4291$, respectively, with $n = 6$, $q = 4$. By comparing the results from the proposed CPSO, other CPSO, PSO and GA, it is seen that the best result from the proposed CPSO is more stable than that of the other algorithms.

## 7. An experiment on the design of a pressure vessel

There are some studies reported in the literature wherein improved PSOs were used for constrained optimization problems. Various constraint handling techniques were employed to facilitate the optimization process. The pressure vessel problem described in Dep (1997), Coello (2000) and Hu *et al.* (2003) is an example that has both linear and nonlinear constraints and has been solved by a variety of PSO techniques. The aim is to design a pressure vessel using the proposed approach. The objective of the problem is to minimize the total cost of the material, forming and welding of a cylindrical vessel. There are four design variables: $x_1$ ($T_s$, thickness of the shell), $x_2$ ($T_h$, thickness of the head), $x_3$ ($R$, inner radius), and $x_4$ ($L$, length of the cylindrical section of the vessel). $x_1$, $x_2$ are formed from plates available only on thickness increments of 0.0625 inch and $R$ and $L$ are continuous. The problem can be stated as follows.

Minimize

$$f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \tag{37}$$

subject to

$$\left.\begin{aligned} g_1(X) &= -x_1 + 0.0193x_3 \le 0, \\ g_2(X) &= -x_2 + 0.00954x_3 \le 0, \\ g_3(X) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3 + 1296000 \le 0, \\ g_4(X) &= x_4 - 240 \le 0. \end{aligned}\right\} \tag{38}$$

The following ranges of variable are used (Coello 2000):

$$0 < x_1 \le 99; \quad 0 < x_2 \le 99; \quad 10 \le x_3 \le 200; \quad 10 \le x_4 \le 200. \tag{39}$$

When CPSO is applied to deal with the constraints of this example, the constraint extended method is applied. Ho *et al.* (2005) found that there is a relationship between the global and local search which can improve the searchability of PSO. This result was applied in this article to choose the parameters $A$ and $B$ and it is proved that parameters chosen in this way can also guarantee global convergence in theory and simulation. In order to enhance the global searchability of CPSO, set $A = Rand(1)$, $B = 1 - A$ and $C = 0.5$ with a population size of 30. During the search process, 10 particles were chosen to search the extended area, given by

$$\begin{aligned} g_{11}(X) &= -x_1 + 0.0193x_3 - 3 \le 0, \\ g_{22}(X) &= -x_2 + 0.00954x_3 - 0.2 \le 0, \\ g_{33}(X) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3 + 1,296,000 - 500 \le 0. \end{aligned} \tag{40}$$

during the first 7500 iterations. The maximum iterations were 15,000. After the first 7500 iterations, these 10 particles were discarded and only the remaining 20 particles were used to search the constrained space given by Equation (38). Each time the best previous position ($p_g$) of the particle swarm changed, the refractory strength $z(t)$ was reset to 0.5 to guarantee that the system was chaotic. The time evolution of the pressure vessel problem for the proposed CPSO and SPSO (Clerc 2004) is shown in Figure 12. Although SPSO has fast convergence ability, it can not get a good result as shown in Figure 12. As in Coello (2000), eleven runs were executed to compare the results. The best solution found by CPSO is better than that previously reported in the literature

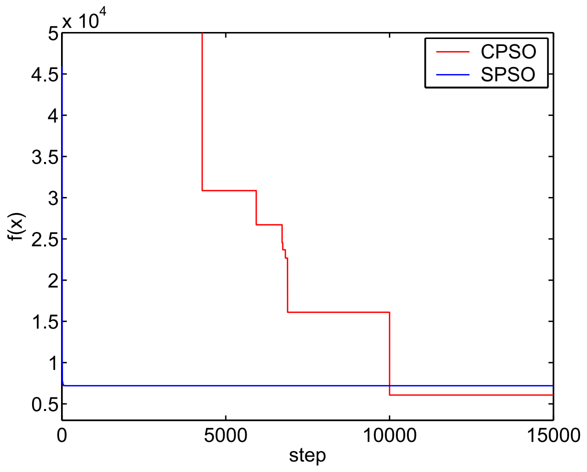

Figure 12. Time evolutions of pressure vessel function for the proposed CPSO and SPSO (Clerc 2004).

Table 3. Comparison of results for the pressure vessel design problem.

| Design variables | Best solutions found | | | |
|---|---|---|---|---|
| | This article | SPSO (Clerc 2004) | Coello (Coello 2000) | GeneAS (Deb 1997) |
| $x_1(T_s)$ | 0.8125 | 1.125 | 0.8125 | 0.9375 |
| $x_2(T_h)$ | 0.4375 | 0.625 | 0.4375 | 0.5000 |
| $x_3(R)$ | 42.09845 | 58.290155 | 40.3239 | 48.3290 |
| $x_4(L)$ | 176.6366 | 43.692656 | 200.0000 | 112.6790 |
| $g_1(X)$ | 0 | $-8.5 \times 10^{-9}$ | −0.03424 | −0.00475 |
| $g_2(X)$ | −0.03588 | −0.068912 | −0.052847 | −0.038941 |
| $g_3(X)$ | $-5.8208 \times 10^{-11}$ | −0.030298 | −27.105845 | −3652.8768 |
| $g_4(X)$ | −63.3634 | −196.3073 | −40 | −127.321 |
| $f(X)$ | **6059.7143** | **7197.7289** | **6288.7445** | **6410.3811** |

as shown in Table 3. The worst solution found by CPSO is 6480.7, which is also better than any solution previously reported. For this problem, the standard deviations are 80.418 and 327.188 for the CPSO and PSO (Kennedy *et al.* 1995), respectively. It is clear that the proposed CPSO is more stable than PSO (Kennedy *et al.* 1995).

*Remark 3* Although $A$ is a random number, the particle final states can also converge to a sphere with centre $p_g^j$ and radius max $\|p_g^j - p_i^j\|$.

*Proof* According to Theorem 5.2, the particle final states converge to a sphere with centre point $p_g^j$ and radius

$$r \leq \frac{BC}{AB + BC + AC}(\max \|p_g^j - p_i^j\|).$$

Set

$$\left.\begin{aligned} f(A) &= \frac{BC}{AB + BC + AC} \\ &= \frac{0.5(1 - A)}{(A - A^2 + 0.5)^2} \\ f'(A) &= \frac{-0.5[(A-1)^2 + 0.5]}{(A - A^2 + 0.5)^2} < 0. \end{aligned}\right\} \tag{41}$$

$f(A)$ therefore decreases monotonically. When $A = 0$, $f(A)_{\max} = 1$, the max radius is

$$\max \|p_g^j - p_i^j\|.$$

The above analysis verifies Remark 3. ■

The particle best experiences and final states all overlap at the optimal point (0.8125, 0.4375, 42.09845, 176.6366) in the simulation which means the particle final states converge.

## 8. Conclusion

This article proposed a chaotic particle model for PSO. The ergodic searching capability of chaos can improve the optimization performance. The decay factor introduced in the particle swarm can ensure that the searching evolves to convergence to the global optimum after chaotic searching.

The experimental results from three classic benchmarks and one engineering optimization problem showed that the proposed CPSO can guarantee the convergence of the particle swarm searching and can escape from local extrema. As the model was derived from an HNN, it more naturally describes the character of particles. As this is a general particle model, many techniques that have been proposed for the original PSO can be used together with the new model. This will be explored in future work.

## Acknowledgements

This work is supported by Research Funding from the TUT, grants from Incentive Funding of the National Research Foundation of South Africa (IFR2009090800049), and the National Scientific Foundation of China (10772135).

## References

Aihara, K., Takabe, T., and Toyoda, M., 1990. Chaotic neural networks. *Physics Letters A*, 144 (6–7), 333–340.

Chandramouli, K. and Izquierdo, E., 2006. Image classification using chaotic particle swarm optimization. *In*: *Proceedings of the international conference on image processing (ICIP'06)*, 8–11 October Atlanta, GA. New York: IEEE Press, 3001–3004.

Chen, L. and Aihara, K., 1995. Chaotic simulated annealing by a neural networks model with transient chaos. *Neural Networks*, 8 (6), 915–930.

Chen, L. and Aihara, K., 1999. Global searching ability of chaotic neural networks. *IEEE Transactions on Automatic Control*, 46 (8), 974–993.

Clerc, M., 2004. *Basic PSO program* [online]. Available from: http://clerc.maurice.free.fr/pso/ [Accessed 2 April 2010].

Clerc, M., 2006. *Particle swarm optimization*. London: ISTE Publishing.

Clerc, M. and Kennedy, J., 2002. The particle swarm: explosion, stability, and convergence in multidimension complex space. *IEEE Transactions on Evolutionary Computation*, 6 (1), 58–73.

Coello, C.A., 2000. Use ora self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41 (2), 113–127.

Deb, K., 1997. GeneAS: a robust optimal design technique for mechanical component design. *In*: D. Dasgupta and Z. Michalewicz, eds. *Evolurionoary algorithms in engineering applications*. Berlin: Springer-Verlag, 497–514.

del Valle, Y. and Venayagamoorthy, G.K., 2008. Particle swarm optimization: basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation*, 12 (2), 171–195.

Eberhart, R.C. and Shi, Y., 1998. Comparison between genetic algorithms and particle swarm optimization. *In*: *Proceedings of the 7th IEEE international conference on evolving computation*, 25–27 March San Diego, CA. New York: IEEE Press, 611–616.

Fan, C. and Jiang, G., 2008. A simple particle swarm optimization combined with chaotic search. *In*: *Proceedings of the 7th world congress on intelligent control and automation*, 25–27 June, Chongqing, China. New York: IEEE Press, 593–598.

Goldberg, D.E., 1989. *Genetic algorithms in search, optimization, and machine learning*. New York: Addison-Wesley.

Ho, S.L., *et al.*, 2005. A particle swarm optimization-based method for multiobjective design optimizations. *IEEE Transactions on Magnetics*, 41 (5), 1756–1759.

Hopfield, J.J., 2009. Hopfield network. *Scholarpedia*, 2 (5), 1977–1981.

Hopfield, J.J. and Tank, D.W., 1985. Neural network of decisions in optimization problems. *Biological Cybernetics*, 52 (3), 141–152.

Hu, X.H., Eberhart, R.C., and Shi, Y.H., 2003. Engineering optimization with particle swarm. *In*: *Proceedings of the 2003 IEEE swarm intelligence symposium*, 24–26 April, Indianapolis, IN. New York: IEEE Press, 53–57.

Hu, X.H., Shi, Y.H., and Eberhart, R., 2004. Recent advances in particle swarm. *In*: *Congress on evolutionary computation*, 19–23 June, Portland, OR. New York: IEEE Press, 90–97.

Huang, C.M., Huang, C.J., and Wang, M.L., 2005. A particle swarm optimization to identifying the ARMAX model for short-term load forecasting. *IEEE Transactions on Power Systems*, 20 (2), 1126–1133.

Kennedy, J. and Eberhart, R., 1995. Particle swarm optimization. *In*: *Proceedings of the IEEE international conference on neural networks*, 27 Nov–1 December, Perth, Australia. New York: IEEE Press, 942–948.

Kennedy, J., Eberhart, R.C., and Shi, Y.H., 2001. *Swarm intelligence*. San Francisco, CA: Morgan Kaufmann.

Kirkpatrick, S., Gelatt, C.D., Jr, and Recchi, M.P., 1983. Optimization by simulated annealing. *Science*, 220 (4598), 671–680.

Liu, B., *et al.*, 2005. Improved particle swarm optimization combined with chaos. *Chaos, Solitons and Fractals*, 25 (5), 1261–1271.

Mendes, R., Kennedy, J., and Neves, J., 2004. The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8 (3), 204–210.

Nedjah, N. and Mourelle, L.D.M., 2007. *Systems engineering using particle swarm optimization*. Hauppauge, NY: Nova Science Publishers.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature*, 323 (9), 533–536.

Shi, Y. and Eberhart, R., 1998. A modified particle swarm optimizer. *In*: *Proceedings of the IEEE international conference on evolutionary computation*, 4–9 May, Anchorage, AK. New York: IEEE Press, 69–73.

Song, Y., Chen, Z.Q., and Yuan, Z.Z., 2007. New chaotic PSO-based neural network predictive control for nonlinear process. *IEEE Transactions on Neural Networks*, 18 (2), 595–600.

Tang, X., Zhuang, L., and Jiang, 2009. Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization. *Expert Systems with Applications*, 36 (9), 11853–11857.

van den Bergh, F., 2001. An analysis of particle swarm optimizers. Thesis (PhD). University of Pretoria, South Africa.

Wang, L., 1997. On competitive learning. *IEEE Transactions on Neural Networks*, 8 (5), 445–447.

Wang, L. and Smith, K., 1998. On chaotic simulated annealing. *IEEE Transactions on Neural Networks*, 9 (4), 716–718.

Wilson, E.O., 1975. *Sociobiology: the new synthesis*. Cambridge, MA: Belknap Press.