Applied Soft Computing 17 (2014) 12–22

Contents lists available at ScienceDirect

Applied Soft Computing

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

# Hybrid parallel chaos optimization algorithm with harmony search algorithm[☆]

Xiaofang Yuan[*], Jingyi Zhao, Yimin Yang, Yaonan Wang

*College of Electrical and Information Engineering, Hunan University, 410082, China*

ARTICLE INFO

*Article history:*
Received 21 September 2012
Received in revised form 4 September 2013
Accepted 16 December 2013
Available online 2 January 2014

*Keywords:*
Chaos optimization algorithm (COA)
Parallel chaos optimization algorithm (PCOA)
Harmony search algorithm
Hybrid algorithm
Chaotic map

ABSTRACT

The application of chaotic sequences can be an interesting alternative to provide search diversity in an optimization procedure, named chaos optimization algorithm (COA). Since the chaotic motion is pseudo-randomness and chaotic sequences are sensitive to the initial conditions, the search ability of COA is usually effected by the starting values. Considering this weakness, parallel chaos optimization algorithm (PCOA) is studied in this paper. To obtain optimum solution accurately, harmony search algorithm (HSA) is integrated with PCOA to form a novel hybrid algorithm. Different chaotic maps are compared and the impacts of parallel parameter on the hybrid algorithm are discussed. Several simulation results are used to show the effective performance of the proposed hybrid algorithm.

## 1. Introduction

Chaos has been applied in many scientific fields since the first introduction of canonical chaotic attractor in 1963 by Lorenz [1]. Chaos is a bounded unstable dynamic behavior that exhibits sensitive dependence on its initial conditions [2]. An essential feature of the chaotic systems is that small changes in the parameters or the starting values lead to the vastly different future behaviors. Recently, chaotic sequences have been adopted instead of random sequences to provide the search diversity in an optimization procedure, named chaos optimization algorithm (COA) [2–7]. Due to the non-repetition of the chaos, the COA can carry out overall searches at higher speeds than stochastic ergodic searches that depend on the probabilities. The COA, which has the features of easy implementation, short execution time and robust mechanisms of escaping from the local optimum, is a promising optimization tool for the engineering applications [2–7]. Recently, in most of those literatures, researchers have considered integrated use of chaotic sequences in order to enhance the performance of the meta-heuristics algorithms, such as chaotic harmony search algorithm [8,9], chaotic ant swarm optimization [10–12], chaotic particle swarm optimization [13–21], chaotic genetic algorithms or chaotic evolutionary algorithm [22–26], chaotic differential evolution [27–29], chaotic firefly algorithm [30], chaotic simulated annealing [31,32], hybrid COA with artificial emotion [33].

Since the chaotic motion is pseudo-randomness and chaotic sequences are sensitive to the initial conditions, COA's search ability is usually effected by the starting values. So, a kind of parallel chaotic optimization algorithm (PCOA) is proposed in our former study [34,35], which shows its superiority over general COA. In the PCOA, multiple chaotic maps are simultaneously mapped onto one decision variable, so PCOA searches from diverse initial points and can detract the sensitivity of initial condition. Although the hybrid algorithm based on PCOA in [34,35] can reach optimum solutions, the search speed is slow since the employed local search technique – simplex search method has slow efficiency.

Harmony search algorithm (HSA) [36–38], first introduced to optimize various continuous nonlinear functions by Geem et al. [36], is one of the latest evolutionary computation techniques, mimicking the musical process of search for a perfect state of harmony. In comparison to other meta-heuristics, the HSA imposes fewer mathematical requirements and can be easily adapted, furthermore, numerical comparisons demonstrated that the evolution in HSA is faster than genetic algorithm [39]. Recently, several improvements have been presented to HSA, such as chaotic harmony search algorithm [8,9] and improved HSA [38–41].

In this paper, to obtain optimum solution accurately, HSA is integrated with PCOA to form a novel hybrid optimization algorithm.

[☆] This work was supported in part by the National Natural Science Foundation of China (No. 61104088), Planned Science and Technology Project of Hunan Province (No. 2011RS4029), and Young Teachers Promotion Program of Hunan University.

[*] Corresponding author. Tel.: +86 13873195923.
*E-mail addresses:* yuanxiaofang126@126.com, yuanxiaofang@hnu.edu.cn (X. Yuan).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.12.016

The proposed hybrid algorithm is a two stages search technique, the first stage is the PCOA with twice carrier wave parallel chaos search for global searching, and the second stage is the HSA for accurate local searching. In the proposed hybrid algorithm, PCOA is conducted until it has converged to a close neighborhood or it has reached its maximum iteration times, then the HSA is conducted for accurate solution. Different chaotic maps are compared and the impacts of parallel parameter on the hybrid algorithm are discussed.

The rest of this paper is organized as follows. Section 2 briefly describes chaotic maps. The PCOA approach is introduced in Section 3. Section 4 gives presentation of the proposed hybrid PCOA with HSA. Simulation results showing the effectiveness of the hybrid algorithm in Section 5. Conclusions are presented in Section 6.

## 2. Chaotic map

One dimensional noninvertible maps have capability to generate chaotic motion. In this study, eight well-known one-dimensional chaotic maps in [42] are considered here.

### 2.1. Logistic map

This map was introduced by Robert May in 1976, and he pointed out that the logistic map led to chaotic dynamics. Logistic map generates chaotic sequences in (0,1). This map is also frequently used in the COAs [2,14,22,34]. This map is formally defined by the following equation:

$$x_{n+1} = \varphi x_n(1 - x_n), \quad 0 < \varphi \leq 4, \quad x_n \in (0, 1) \tag{1}$$

### 2.2. Tent map

Tent chaotic map is very similar to the logistic map, which displays specific chaotic effects. Tent map generates chaotic sequences in (0,1). This map is formally defined by the following equation:

$$\begin{aligned} x_{n+1} &= \frac{x_n}{0.7}, \quad x_n < 0.7 \\ x_{n+1} &= \left(\frac{10}{3}\right) x_n(1 - x_n), \quad else \end{aligned} \tag{2}$$

### 2.3. Chebyshev map

Chebyshev chaotic map is a common symmetrical region map. It is generally used in neural networks, digital communication and security problems [16]. Chebyshev map generates chaotic sequences in (−1,1). This map is formally defined [43]:

$$x_{n+1} = \cos(\varphi\, cos^{-1} x_n), \quad 0 < \varphi, \quad x_n \in [-1, 1] \tag{3}$$

### 2.4. Circle map

Circle chaotic map was proposed by Kolmogorov. This map describes a simplified model of the phase locked loop in electronics [44]. This map is formally defined [45]:

$$x_{n+1} = x_n + \vartheta - (\frac{\tau}{2\pi})\sin(2\pi x_n)\, mod\,(1), \quad x_n \in (0, 1) \tag{4}$$

Circle map generates chaotic sequences in (0,1). In Eq. (4), $x_{n+1}$ is computed mod 1.

### 2.5. Cubic map

Cubic map is one of the most commonly used maps in generating chaotic sequences in various applications like cryptography. This map is formally defined by:

$$x_{n+1} = \rho x_n(1 - x_n^2), \quad x_n \in (0, 1) \tag{5}$$

Cubic map generates chaotic sequences in (0,1) with $\rho = 2.59$.

### 2.6. Gauss map

Gauss map is also one of the very well known and commonly employed map in generating chaotic sequences in various applications like testing and image encryption. This map is formally defined by the following equation:

$$\begin{aligned} x_{n+1} &= 0, \quad x_n = 0 \\ x_{n+1} &= \frac{1}{x_n}\, mod\,(1), \quad x_n \neq 0 \end{aligned} \tag{6}$$

Gauss map generates chaotic sequences in (0,1).

### 2.7. ICMIC map

Iterative chaotic map with infinite collapses (ICMIC) [43] is formally defined by the following equation:

$$x_{n+1} = \sin\left(\frac{\alpha}{x_n}\right), \quad \alpha \in (0, \infty), \quad x_n \in (-1, 1) \tag{7}$$

ICMIC map generates chaotic sequences in (−1,1).

### 2.8. Sinusodial map

Sinusodial map is formally defined by the following equation:

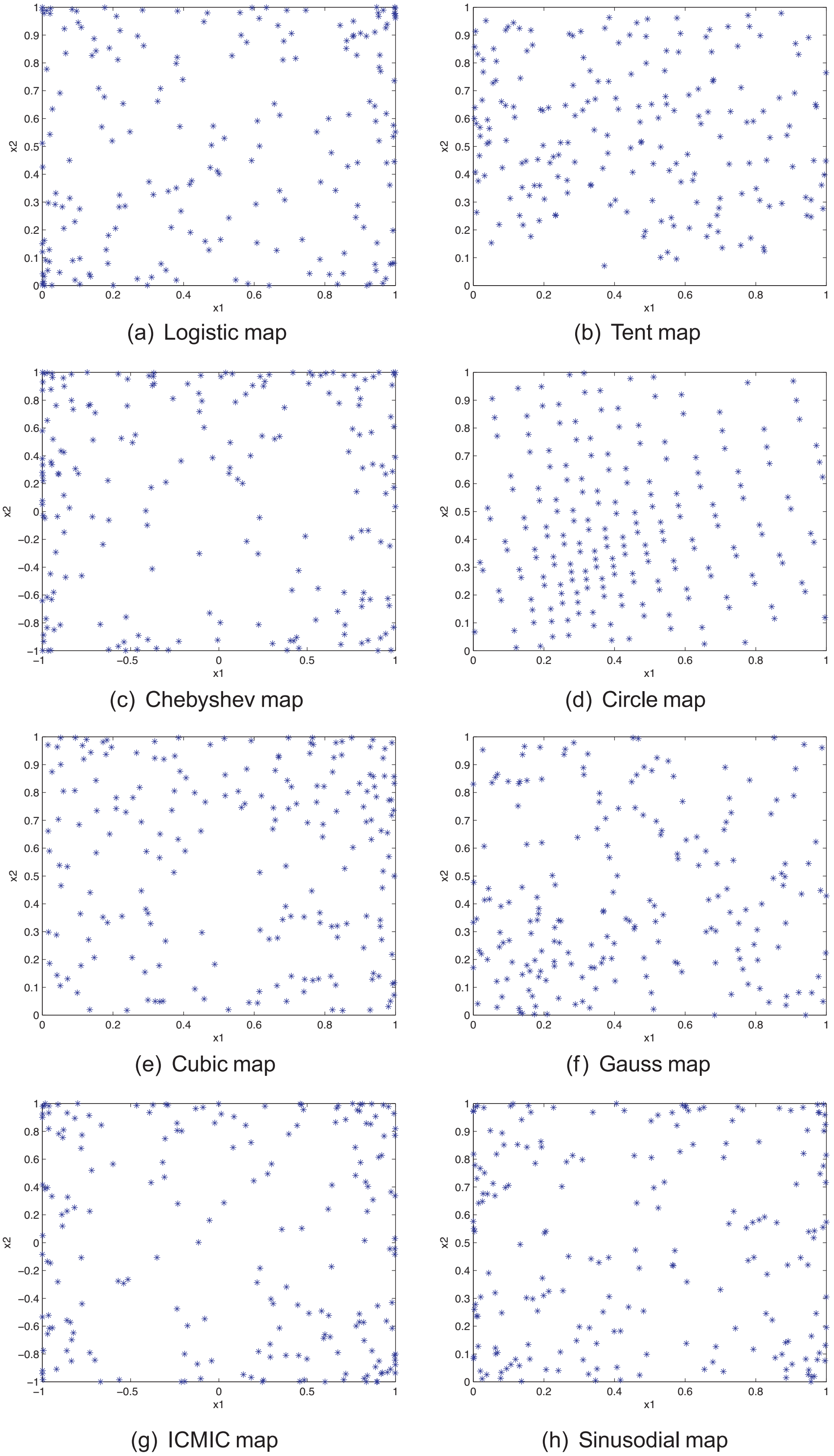
$$x_{n+1} = \sin(\pi x_n), \quad x_n \in (0, 1) \tag{8}$$

Sinusodial map generates chaotic sequences in (0,1).

Typical behaviors of two chaos variables ($x_1$, $x_2$) based on different chaotic maps with 200 iterations are shown in Fig. 1. Here the initial points of two chaos variables are: $x_1 = 0.152$, $x_2 = 0.843$. Fig. 1(a) is the result of logistic map ($\varphi = 4$), Fig. 1(b) is the result of tent map, Fig. 1(c) is the result of Chebyshev map ($\varphi = 5$), Fig. 1(d) is the result of circle map ($\vartheta_1 = \vartheta_2 = 0.5$, $\tau_1 = \tau_2 = 5$), Fig. 1(e) is the result of cubic map, Fig. 1(f) is the result of Gauss map, Fig. 1(g) is the result of ICMIC map ($\alpha = 70$), Fig. 1(h) is the result of sinusodial map, respectively. From Fig. 1, it can be observed that each chaos variables are randomly distributed in their certain bounds. Although 200 iterations are shown, chaos variables have good ergodic properties with the increase of iteration times. This characteristic is very useful for global search, and it is helpful for COA's global optimum. From Fig. 1, it can also be observed that the distribution or ergodic property of different chaotic maps are different. Therefore, the search patterns of different chaotic maps differs with each others in view of convergence rate, algorithm speed and accuracy [14,42,44].

## 3. PCOA approach

In the PCOA proposed in our former study [34,35], multiple chaotic maps are mapped onto one optimization variable simultaneously, and the search result is the best value of parallel multiple chaotic maps. In this way, the PCOA method searches from several different initial points and detracts the sensitivity of initial condition.



(a) Logistic map

(b) Tent map

(c) Chebyshev map

(d) Circle map

(e) Cubic map

(f) Gauss map

(g) ICMIC map

(h) Sinusodial map

**Fig. 1.** Chaotic maps in two-dimension space with 200 iterations.

Consider an optimization problem for nonlinear multi-modal function with boundary constraints as:

$$\min P(x) = P(x_1, x_2, \ldots, x_n), \quad x_i \in [a_i, b_i]. \tag{9}$$

In the following, $i = 1, 2, \ldots, n$, which represents each optimization variable; $j = 1, 2, \ldots, N$, which represents each optimization variable mapped by multiple N chaotic maps simultaneously.

The process of a kind of twice carrier wave PCOA approach is described as follows. The first is the raw searching in different chaotic trace, while the second is elaborate searching by continually reducing the searching space of variable optimized and enhancing the searching precision.

### *3.1. Parallel chaos search by using the first carrier wave*

*Step* 1: Initialize the maximum iteration times $S_1$ in the first carrier wave chaos search, random initial value of chaotic maps $0 < \gamma_{ij}^{(0)} < 1$.

*Step* 2: Set iteration times $l = 0$, parallel optimum $P_j^* = \infty$, and global optimum $P^* = \infty$.

*Step* 3: Map chaotic maps $\gamma_{ij}^{(l)}$ onto the variance range of the optimization variables by the following equation:

$$x_{ij(l)} = a_i + \gamma_{ij(l)}(b_i - a_i) \tag{10}$$

where

$$x^{(l)} = \begin{bmatrix} x_{11}^{(l)} & x_{12}^{(l)} & \ldots & x_{1N}^{(l)} \\ x_{21}^{(l)} & x_{22}^{(l)} & \ldots & x_{2N}^{(l)} \\ \ldots & \ldots & \ldots & \ldots \\ x_{n1}^{(l)} & x_{n2}^{(l)} & \ldots & x_{nN}^{(l)} \end{bmatrix}$$

*Step* 4: Compute the objective function value for each decision variable and update the searching results. If $P_j(x^{(l)}) \le P_j^*$, then $x_j^* = x_j^{(l)}$ and parallel optimum $P_j^* = P_j(x^{(l)})$. If $P_j^* \le P*$, then global optimum $P* = P_j^*$, and $x* = x_j^*$. This means that the search result is the best value of parallel multiple chaotic maps.

*Step* 5: Generate next values of chaotic maps by a chaotic map function ($M$) as in Eqs. (1)–(8):

$$\gamma_{ij(l+1)} = M(\gamma_{ij(l)}) \tag{11}$$

*Step* 6: If $l \ge S_1$ or $\|P_j^* - P*\| < P_H$, stop the first carrier wave search process; otherwise $l \longleftarrow l + 1$, go to Step 3.

### *3.2. Parallel chaos search by using the second carrier wave*

*Step* 1: Initialize the maximum iteration times $S_2$ in the second carrier wave chaos search, set iteration times $l' = 0$, random initial value of chaotic maps $0 < \gamma_{ij}^{(l')} < 1$.

*Step* 2: Compute the second carrier wave by the following equation:

$$x_{ij(l')} = x_j^* + \lambda_i(\gamma_{ij(l')} - 0.5) \tag{12}$$

*Step* 3: Compute the objective function value for each decision variable and update the searching results. If $P_j(x^{(l')}) \le P_j^*$, then and parallel optimum. If $P_j^* \le P*$, then global optimum $P* = P_j^*$, and $x* = x_j^*$.

*Step* 4: Generate next values of chaotic maps by a chaotic map function ($M$) as in Eqs. (1)–(8):

$$\gamma_{ij(l\prime+1)} = M(\gamma_{ij(l\prime)}) \tag{13}$$

*Step* 5: If $l' \ge S_2$ or $\|P_j^* - P*\| < P_L$, stop the second carrier wave search process; otherwise $\lambda_i \longleftarrow t\lambda_i$, go to Step 2.

The $\lambda$ is a very important parameter and adjusts small ergodic ranges around $x_j^*$, and $t > 1$. It is difficult and heuristic to determine the appropriate value of $\lambda_i$, initial value of this parameter is usually set to $0.01(b_i - a_i)$ [44].

Since the PCOA is an effective global search approach and it is not sensitive to its initial conditions, after many iteration times, parallel optimum $P_j^*$ will converge to a close neighborhood, and $P_j^*$ will be close to global optimum $P^*$. Here $\|P_j^* - P*\| < P_H$ or $\|P_j^* - P*\| < P_L$ means whether the parallel search results of PCOA have reached a close neighborhood, and $\|P_j^* - P*\| < P_H$ is computed as:

$$|P_1^* - P*| < P_H \text{ and } |P_2^* - P*| < P_H \ldots \text{and } |P_j^* - P*| < P_H \tag{14}$$

In the same manner, $\|P_j^* - P*\| < P_L$ is similar to Eq. (14), and $P_H$ and $P_L$ are pre-set values.

## 4. Hybrid PCOA with HSA

Due to the pseudo-randomness of chaotic motion, the motion step of chaotic maps between two successive iterations is always big, which resulted in the big jump of the decision variables in searching space. Thus, even if PCOA has reached the neighborhood of the global optimum, it needs to spend much computational effort to approach the optimum eventually by searching numerous points [4]. Then, although the above mentioned twice carrier wave PCOA has good capacity for global exploring, its accurate search ability is not enough. In order to improve the accurate local search of PCOA, a heuristic algorithm – harmony search algorithm (HSA) is employed here. The HSA can be conducted for accurate local search since numerical comparisons have demonstrated that the evolution in HSA is faster than genetic algorithm [39]. Consequently, by combining HSA with PCOA, a novel hybrid optimization algorithm can be developed with fine capability.

The flowchart of hybrid PCOA with HSA is illustrated in Fig. 2. The proposed hybrid algorithm is a two stages search technique, the first stage is the twice carrier wave PCOA for global searching, and the second stage is the HSA for accurate searching. In the proposed algorithm, the PCOA is conducted until it has converged to a close neighborhood or it has reached its maximum iteration times, then HSA will be conducted for accurate solutions. The condition of switch from PCOA to HSA is: the second carrier wave parallel chaos search is completed ($l' \ge S_2$) or PCOA has converged to a close neighborhood ($\|P_j^* - P*\| < P_L$).

As HSA is the successor of PCOA, so the parallel optimum $P_j^*$ is the starting points of HSA, that is, $x_j^*$ after PCOA search is the initial harmony memory (HM) for HSA. This section describes the HSA after PCOA search as follows:

*Step* 1: Initialize HSA parameters: maximum iteration times $S_3$, harmony memory considering rate (HMCR), pitch adjusting rate (PAR), bandwidth vector (BW).

*Step* 2: Set iteration times $l'' = 0$, initialize the harmony memory (HM) from the PCOA search results.

$$HM1 = x_1^*, \quad HM^2 = x_2^*, \ldots, HMj = x_j^* \tag{15}$$

*Step* 3: Improvise a new harmony from the HM. $x' = (x'_1, x'_2, \ldots, x'_n)$ is improvised based on the following three mechanisms [36,37]: random selection, memory consideration, and pitch adjustment. In the random selection, the value of each decision variable, in the new harmony vector is randomly chosen within the value range with a probability of (1 − HMCR). The HMCR, which varies between 0 and 1, is the rate of choosing one value from the

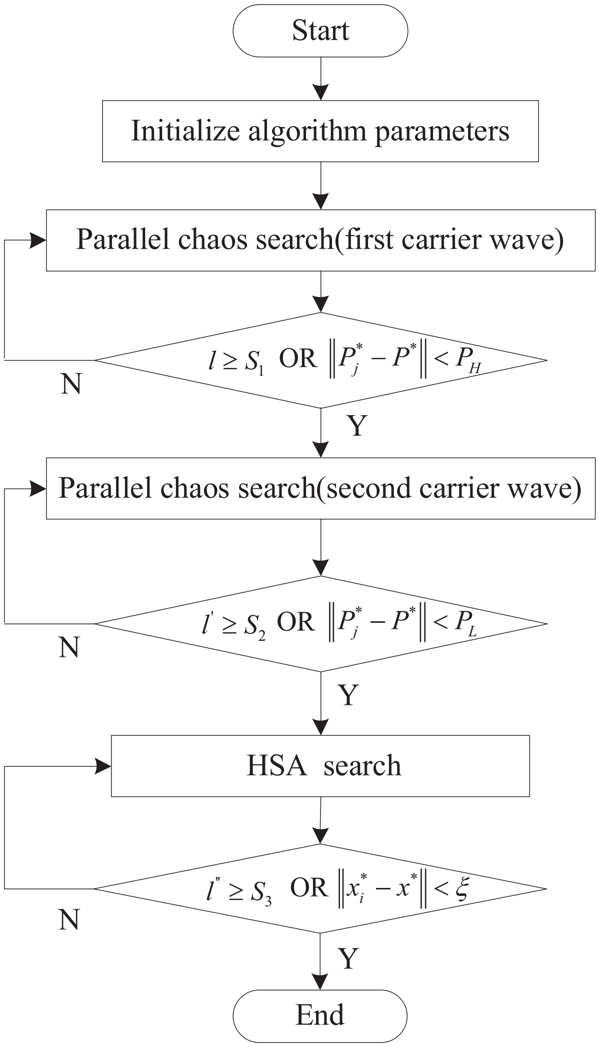

Start
Initialize algorithm parameters
Parallel chaos search(first carrier wave)
$l \geq S_1$ OR $\|P_j^* - P^*\| < P_H$
N
Y
Parallel chaos search(second carrier wave)
$l' \geq S_2$ OR $\|P_j^* - P^*\| < P_L$
N
Y
HSA search
$l'' \geq S_3$ OR $\|x_i^* - x^*\| < \xi$
N
Y
End

**Fig. 2.** Flowchart of the proposed hybrid PCOA with HSA.

historical values stored in the HM, and (1 – HMCR) is the rate of randomly selecting one value from the possible range of values [38].

$$\begin{aligned} x'_i &= x'_i \in \{x_i^1, x_i^2, \ldots, x_i^{HMS}\} \quad \text{with probability HMCR} \\ x'_i &= x'_i \in x_i \quad \text{with probability } (1 - \text{HMCR}) \end{aligned} \tag{16}$$

The value of each decision variable obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. If the pitch adjustment decision for $x'_i$ is made with a probability of PAR, $x'_i$ is replaced with $x'_i \pm u(-1, +1) \times BW$, where $BW$ is an arbitrary distance bandwidth for the continuous design variable, and $u(-1, 1)$ is a uniform distribution between –1 and 1. The value of (1 – PAR) sets the rate of performing nothing. Thus, pitch adjustment is applied to each variable as follows [37]:

$$\begin{aligned} x'_i &= x'_i \pm u(-1, +1) \times BW \quad \textit{with} \text{ probability HMCR} \times \text{PAR} \\ x'_i &= x'_i \quad \textit{with} \text{ probability HMCR} \times (1 - \text{PAR}) \end{aligned} \tag{17}$$

*Step* 4: Update the HM according to the objective function value. With the evaluation of the objective function value, if the new harmony vector is better than the worst harmony vector in the HM, the new harmony vector is included in the HM, and the existing worst harmony vector is excluded from the HM.

*Step* 5: If $\|x_i^* - x*\| < \xi$ or $l'' \geq S_3$, HSA search is terminated; otherwise $l'' \leftarrow l'' + 1$, Step 3 and Step 4 are repeated.

The *PAR* and *BW* in HSA search are important parameters, and they are potentially useful in adjusting convergence rate to optimal solutions [39]. The traditional HSA method uses fixed value for both *PAR* and *BW*, and variable *PAR* and *BW* in [39] is employed:

$$PAR(l'') = PARmin + \frac{(PARmax - PARmin)}{S_3} \cdot l'' \tag{18}$$

$$BW(l'') = BWmax \cdot \exp\left(\frac{l'' \cdot \ln(BWmin/BWmax)}{S_3}\right) \tag{19}$$

In this paper, large *PAR* value with small *BW* value are usually used for the improvement of best solutions in late generations, here $PAR_{min} = 0.75$, $PAR_{max} = 0.99$, $BW_{min} = 0.001$, $BW_{max} = 0.1$.

## 5. Simulation

### 5.1. Benchmarks tests

The efficiency and performance of the proposed hybrid algorithm with the following six nonlinear functions [2,4,40] is evaluated:

$$f_1(x, y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2, \quad -200 < x, y < 200 \tag{20}$$

$$f_2(x, y) = 0.5 - \frac{\sin^2\sqrt{x^2 + y^2} - 0.5}{(1 + 0.001(x^2 + y^2))^2}, \quad -200 < x, y < 200 \tag{21}$$

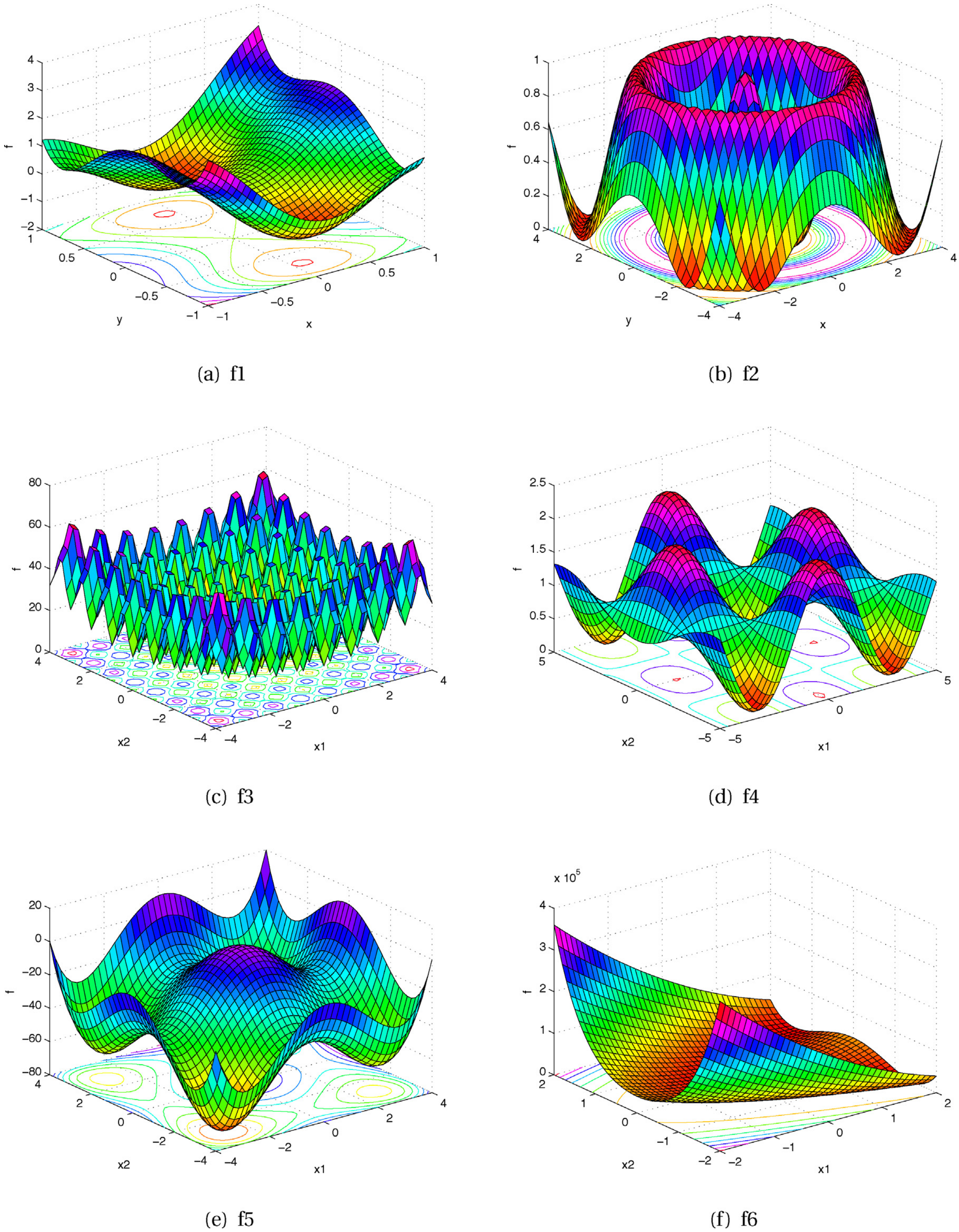
$$f_3(X) = \sum_{i=1}^{3}(x_i^2 - 10\cos(2\pi x_i) + 10), \quad -5 < x_i < 5, \quad i = 1, 2, 3 \tag{22}$$

$$f_4(X) = \frac{1}{4000}\sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad -5 < x_i < 5, \quad i = 1, 2, \ldots, 30 \tag{23}$$

$$f_5(X) = \frac{1}{30}\sum_{i=1}^{30}(x_i^4 - 16x_i^2 - 5x_i), \quad -10 < x_i < 10, \quad i = 1, 2, \ldots, 30 \tag{24}$$

$$f_6(X) = \sum_{i=1}^{30}((x_{i+1} - x_i^2)^2 + (x_i - 1)^2), \quad -10 < x_i < 10, \quad i = 1, 2, \ldots, 30 \tag{25}$$

Function $f_1$ is the Camel function as illustrated in Fig. 3(a), which has six local minima and two global minima $x^* = (-0.0898, 0.7126)$, $x^* = (0.0898, -0.7126)$, and optimal objective function value $f^* = -1.03162845$. Function $f_2$ is the Schaffer's function as illustrated in Fig. 3(b), which has infinite local maximum and one global maximum $x^* = (0, 0)$, and $f^* = 1.0$. Function $f_3$ is the Rastrigin's function as illustrated in Fig. 3(c), which has many local minima and one global minimum $x^* = (0, 0, 0)$, and $f^* = 0$. Function $f_4$ is the Griewank's function with 30 variables as illustrated in Fig. 3(d),



(a) f1

(b) f2

(c) f3

(d) f4

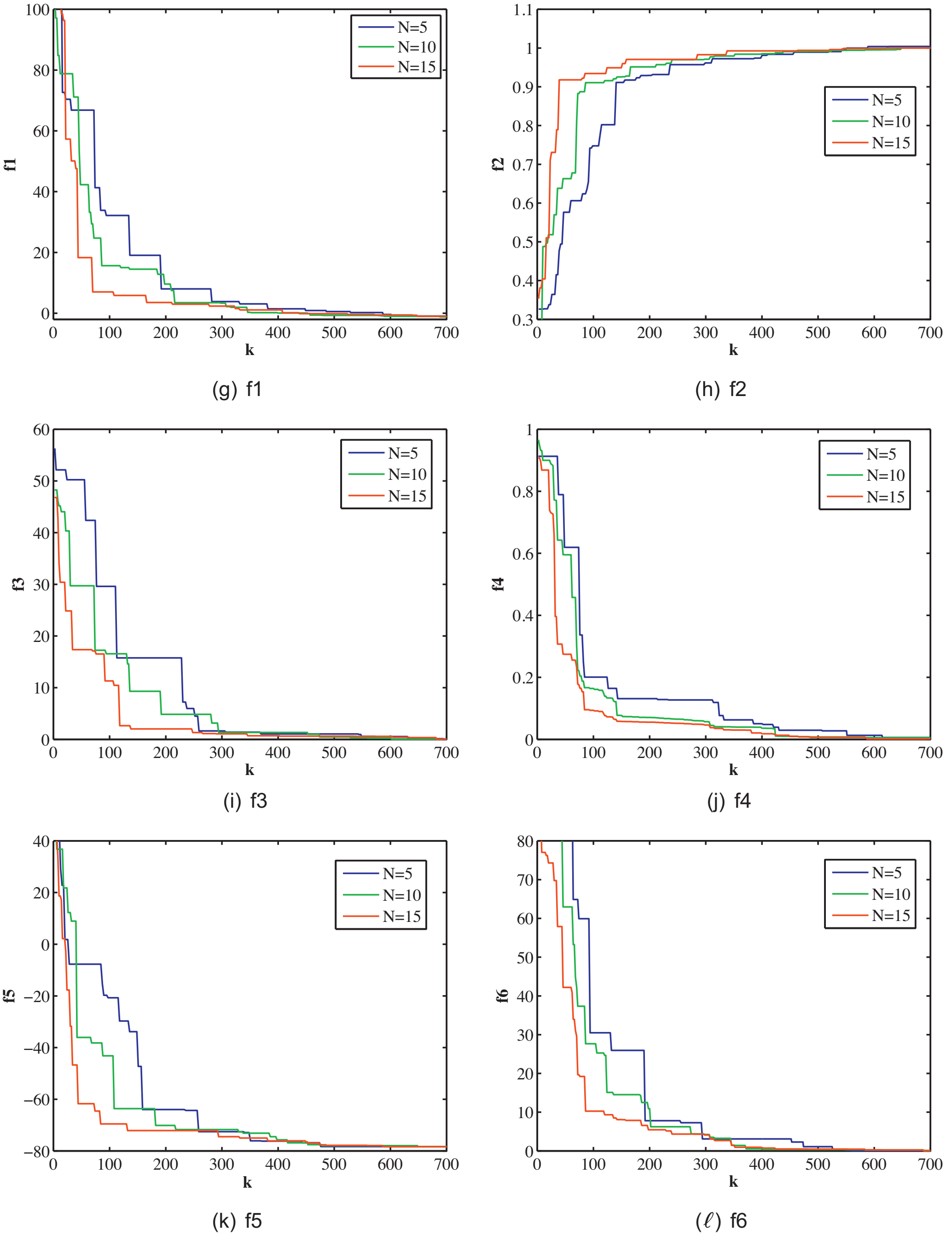
(e) f5

(f) f6

**Fig. 3.** $f1-f6$ function in two-dimension space.

which has several thousand local minima and one global minimum $x^*=(0, 0, \ldots, 0)$, and $f^*=0$. Function $f_5$ has 30 variables as illustrated in Fig. 3(e), which has many local minima and one global minimum $x^*=(2.9051, 2.9051, \ldots, 2.9051)$, and $f^*=-78.332314$. Function $f_6$ has 30 variables as illustrated in Fig. 3(f), which has several local minima and one global minimum $x^*=(1, 1, \ldots, 1)$, and $f^*=0$. These six nonlinear multi-modal functions are often used to test the convergence, efficiency and accuracy of optimization algorithm [2,4,40]. Among them the latter three functions have 30 variables, which is high dimensional problem.

The parameters is the hybrid algorithm are: $P_L=0.1$ for $f_2$, $f_4$ while $P_L=0.5$ for $f_1, f_3, f_5$ and $f_6$, $\xi=0.01$ for $f_1-f_3$ while $\xi=0.05$ for $f_4-f_6$, different multiple chaos variables $N$ is used, here Tent map in Eq. (2) is used as the chaotic mapping function $M$. Fig. 4 shows



(g) f1

(h) f2

(i) f3

(j) f4

(k) f5

($\ell$) f6

**Fig. 4.** Objective function values of $f1 - f6$ for different parallel number $N$.

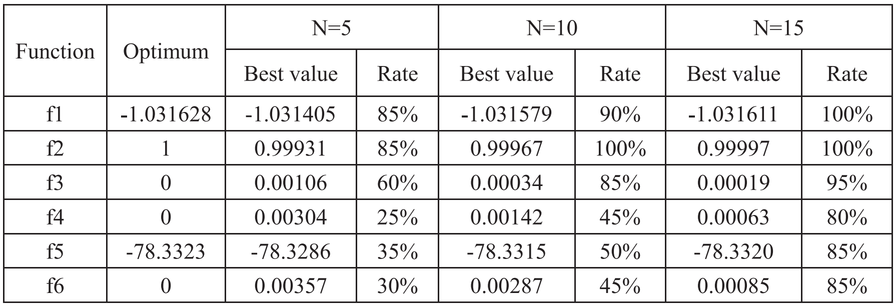
a typical objective function values in the search procedure with respect to different parallel number $N$.

The simulation results are shown in Fig. 5. The ‘best’ means the best objective function value in 20 times searching, the ‘rate’ means the success rate of reaching the optimum x* with the error less than 0.02. In can seen from Figs. 4 and 5 that, as the increase of parallel number $N$, the search speed a is faster and the convergence is better. When $N$ is 15, the proposed hybrid algorithm can reach satisfied simulation results for these benchmarks tests functions. So the parallel number $N$ of the proposed hybrid algorithm is usually no less than 15.

The performance of the proposed hybrid algorithm using different chaotic maps are shown in Fig. 6, where $N = 15$, while $P_L$ is as the same as the former. In can be seen form Fig. 6 that there is some little difference with respect to different chaotic maps, while the Tent map and Circle map are two of the best results.

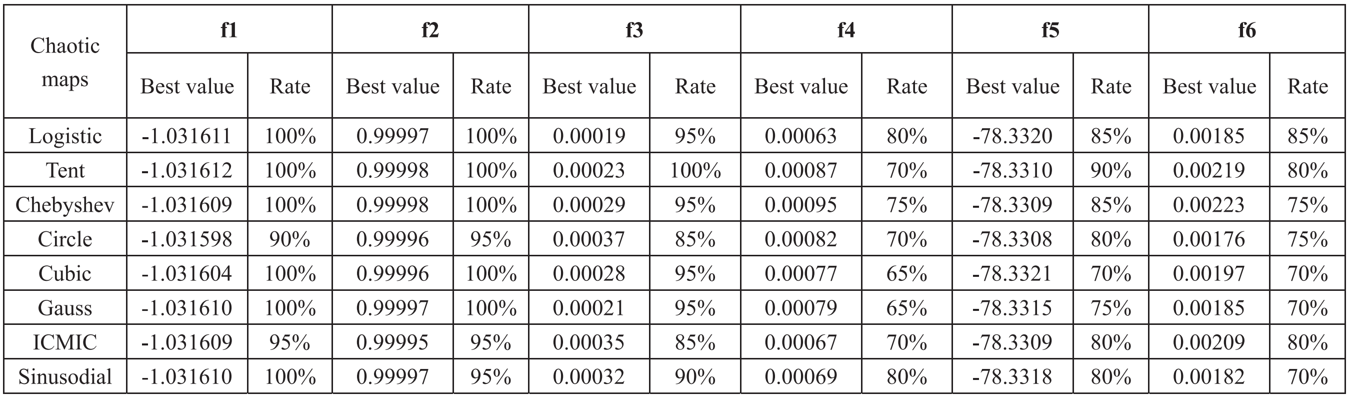
The proposed hybrid algorithm is also compared with other two optimization algorithm based on PCOA in [34,35]. With the similar conditions and parameters, the results of the optimum x* with the error less than 0.02 are shown in Fig. 7, where ‘PCOA+CCIC’ and ‘PCOA+SSM’ means the PCOA algorithms in [34,35], respectively, and ‘PCOA+HSA’ means the proposed hybrid algorithm in this paper. Fig. 7 indicates that the proposed hybrid algorithm have bigger success rate to search the accurate solutions. Fig. 8 shows a typical objective function values in the search procedure for different optimization algorithms with $N = 15$.

Here the proposed hybrid algorithm with N = 30 is also compared with the performances of four widely used evolutionary



| Function | Optimum | N=5 | | N=10 | | N=15 | |
|---|---|---|---|---|---|---|---|
| | | Best value | Rate | Best value | Rate | Best value | Rate |
| f1 | -1.031628 | -1.031405 | 85% | -1.031579 | 90% | -1.031611 | 100% |
| f2 | 1 | 0.99931 | 85% | 0.99967 | 100% | 0.99997 | 100% |
| f3 | 0 | 0.00106 | 60% | 0.00034 | 85% | 0.00019 | 95% |
| f4 | 0 | 0.00304 | 25% | 0.00142 | 45% | 0.00063 | 80% |
| f5 | -78.3323 | -78.3286 | 35% | -78.3315 | 50% | -78.3320 | 85% |
| f6 | 0 | 0.00357 | 30% | 0.00287 | 45% | 0.00085 | 85% |

**Fig. 5.** Simulation results of the proposed hybrid algorithm for different parallel number *N*.



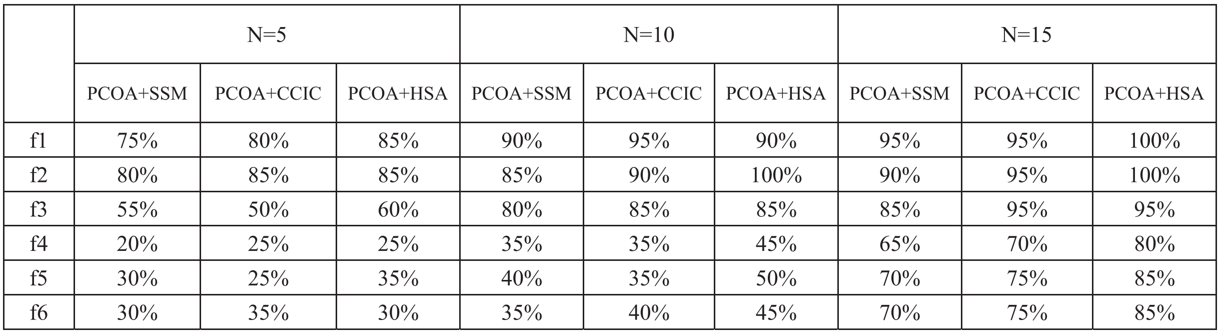
| Chaotic maps | **f1** | | **f2** | | **f3** | | **f4** | | **f5** | | **f6** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Best value | Rate | Best value | Rate | Best value | Rate | Best value | Rate | Best value | Rate | Best value | Rate |
| Logistic | -1.031611 | 100% | 0.99997 | 100% | 0.00019 | 95% | 0.00063 | 80% | -78.3320 | 85% | 0.00185 | 85% |
| Tent | -1.031612 | 100% | 0.99998 | 100% | 0.00023 | 100% | 0.00087 | 70% | -78.3310 | 90% | 0.00219 | 80% |
| Chebyshev | -1.031609 | 100% | 0.99998 | 100% | 0.00029 | 95% | 0.00095 | 75% | -78.3309 | 85% | 0.00223 | 75% |
| Circle | -1.031598 | 90% | 0.99996 | 95% | 0.00037 | 85% | 0.00082 | 70% | -78.3308 | 80% | 0.00176 | 75% |
| Cubic | -1.031604 | 100% | 0.99996 | 100% | 0.00028 | 95% | 0.00077 | 65% | -78.3321 | 70% | 0.00197 | 70% |
| Gauss | -1.031610 | 100% | 0.99997 | 100% | 0.00021 | 95% | 0.00079 | 65% | -78.3315 | 75% | 0.00185 | 70% |
| ICMIC | -1.031609 | 95% | 0.99995 | 95% | 0.00035 | 85% | 0.00067 | 70% | -78.3309 | 80% | 0.00209 | 80% |
| Sinusodial | -1.031610 | 100% | 0.99997 | 95% | 0.00032 | 90% | 0.00069 | 80% | -78.3318 | 80% | 0.00182 | 70% |

**Fig. 6.** Simulation results of the proposed hybrid algorithm using different chaotic maps.

algorithm: particle swarm optimization algorithm (PSO) [46], covariance matrix adaptation evolution strategy (CMAES) [47], adaptive differential evolution algorithm (JDE) [48], and self-adaptive differential evolution algorithm (SADE) [49]. Owing to stochastic nature, evolutionary algorithms may arrive at better or worse solutions than solutions they have previously reached by chance during their search for new solutions to a problem. Because of such cases, it is beneficial to use statistical tools to compare the problem-solving success of one algorithm with that of another. The simple statistical parameters that can be derived from the results of an algorithm solving a specific numerical problem K times under different initial conditions – i.e. the mean solution (mean), the standard deviation of the mean solution (std) and the best solution (best). The simulation results for $f1-f6$ with 30 runs using these techniques are shown in Fig. 9, which has also verified the good performance of the proposed hybrid algorithm.

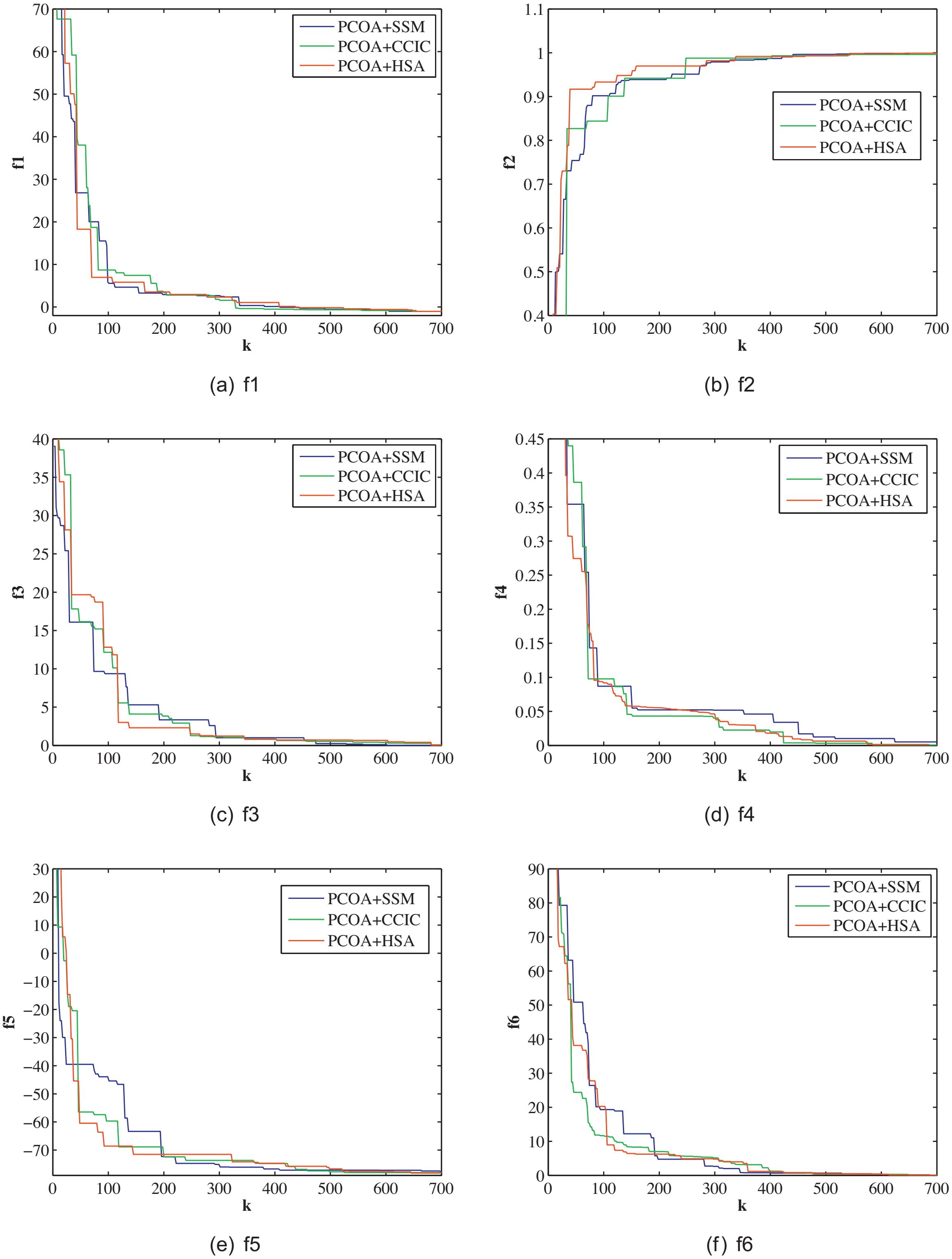
### 5.2. Parameter identification of synchronous generator

In this section, simulation is performed to evaluate the performance of the hybrid algorithm for parameter identification of synchronous generator as in [35]. The nominal values of synchronous generator are: rated power – 176.471 MVA, rated active power – 156.25 MW, rated voltage – 14.4 kV, power factor – 0.85, efficiency – 98.58%, rated speed – 3000 rpm, frequency – 50Hz. The cost function is: $\min C(\hat{p}) = \sum_{k=1}^{n}[(i_d-\hat{i}_d)^2+(i_q-\hat{i}_q)^2+(\delta-\hat{\delta})^2]$, where the current $i_d$, $i_q$ and the power angle $\delta$ are the measurable system outputs, the current $\hat{i}_d$, $\hat{i}_q$ and the power angle $\hat{\delta}$ are computed from the identified system model as in [35]. These variables are calculated using standard per unit values, and the learning data-set with size of $n = 400$ are the same as in [35].



| | N=5 | | | N=10 | | | N=15 | | |
|---|---|---|---|---|---|---|---|---|---|
| | PCOA+SSM | PCOA+CCIC | PCOA+HSA | PCOA+SSM | PCOA+CCIC | PCOA+HSA | PCOA+SSM | PCOA+CCIC | PCOA+HSA |
| f1 | 75% | 80% | 85% | 90% | 95% | 90% | 95% | 95% | 100% |
| f2 | 80% | 85% | 85% | 85% | 90% | 100% | 90% | 95% | 100% |
| f3 | 55% | 50% | 60% | 80% | 85% | 85% | 85% | 95% | 95% |
| f4 | 20% | 25% | 25% | 35% | 35% | 45% | 65% | 70% | 80% |
| f5 | 30% | 25% | 35% | 40% | 35% | 50% | 70% | 75% | 85% |
| f6 | 30% | 35% | 30% | 35% | 40% | 45% | 70% | 75% | 85% |

**Fig. 7.** The success rate by different optimization algorithms for different parallel number *N*.



(a) f1

(b) f2

(c) f3

(d) f4

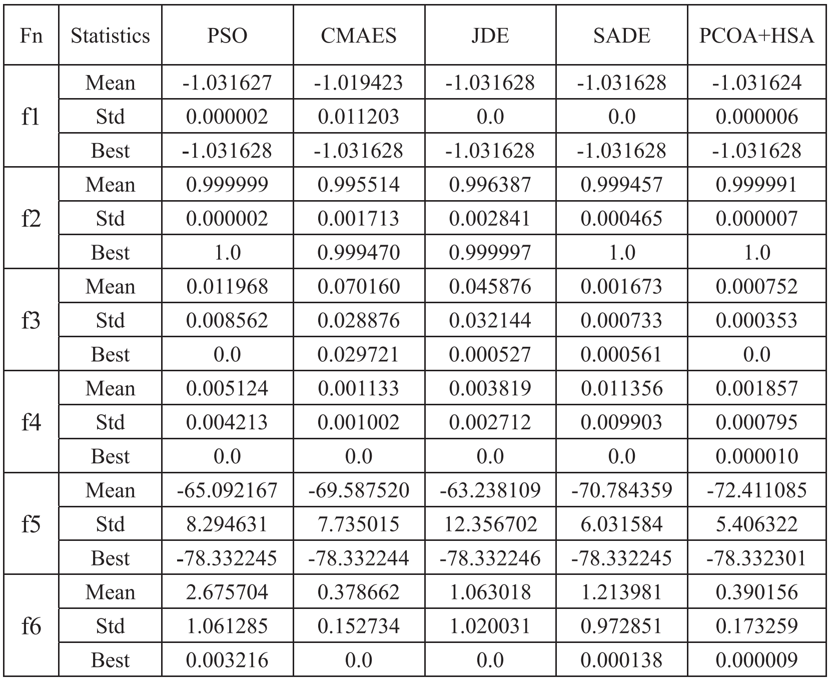
(e) f5

(f) f6

**Fig. 8.** Objective function values of $f1 - f6$ by different optimization algorithms.

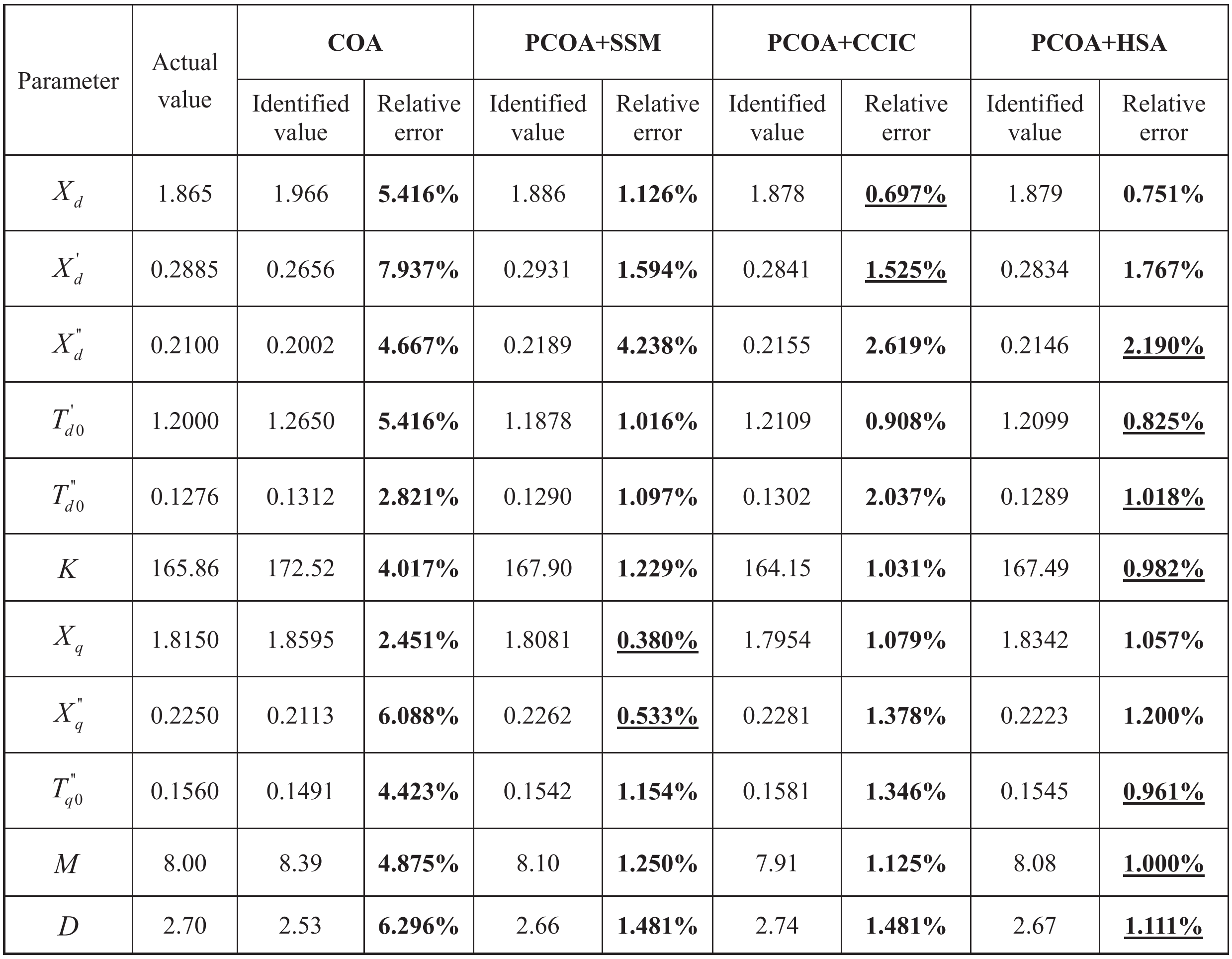
The parameters of the proposed hybrid algorithm are chosen as: $S_1 = 1000$, $S_2 = 800$, $P_L = 0.5$, $\xi = 0.1$, the number of decision variables $n = 11$, the number of multiple chaotic maps $N = 20$.

The results in Fig. 10 are the best results of each optimization algorithm repeated for 30 times, and the minimal cost function value of COA, PCOA+SSM, PCOA+CCIC, and PCOA+HSA are 5.917, 2.159, 2.087, 1.624, respectively. The results of the COA and the PCOA+SSM are mentioned in [35], and the PCOA+CCIC is just the algorithm in [34]. From the identification results in Fig. 10, it is seen that the relative error using the proposed PCOA+HSA algorithm is about 1%, while the relative error of both PCOA+SSM and PCOA+CCIC algorithm is bigger than 1%. In Fig. 10, the best identified value for each parameter is underlined, it shows that seven parameters among the 11 parameters derive the best value by the



| Fn | Statistics | PSO | CMAES | JDE | SADE | PCOA+HSA |
|---|---|---|---|---|---|---|
| f1 | Mean | -1.031627 | -1.019423 | -1.031628 | -1.031628 | -1.031624 |
| | Std | 0.000002 | 0.011203 | 0.0 | 0.0 | 0.000006 |
| | Best | -1.031628 | -1.031628 | -1.031628 | -1.031628 | -1.031628 |
| f2 | Mean | 0.999999 | 0.995514 | 0.996387 | 0.999457 | 0.999991 |
| | Std | 0.000002 | 0.001713 | 0.002841 | 0.000465 | 0.000007 |
| | Best | 1.0 | 0.999470 | 0.999997 | 1.0 | 1.0 |
| f3 | Mean | 0.011968 | 0.070160 | 0.045876 | 0.001673 | 0.000752 |
| | Std | 0.008562 | 0.028876 | 0.032144 | 0.000733 | 0.000353 |
| | Best | 0.0 | 0.029721 | 0.000527 | 0.000561 | 0.0 |
| f4 | Mean | 0.005124 | 0.001133 | 0.003819 | 0.011356 | 0.001857 |
| | Std | 0.004213 | 0.001002 | 0.002712 | 0.009903 | 0.000795 |
| | Best | 0.0 | 0.0 | 0.0 | 0.0 | 0.000010 |
| f5 | Mean | -65.092167 | -69.587520 | -63.238109 | -70.784359 | -72.411085 |
| | Std | 8.294631 | 7.735015 | 12.356702 | 6.031584 | 5.406322 |
| | Best | -78.332245 | -78.332244 | -78.332246 | -78.332245 | -78.332301 |
| f6 | Mean | 2.675704 | 0.378662 | 1.063018 | 1.213981 | 0.390156 |
| | Std | 1.061285 | 0.152734 | 1.020031 | 0.972851 | 0.173259 |
| | Best | 0.003216 | 0.0 | 0.0 | 0.000138 | 0.000009 |

**Fig. 9.** Simulation results of 30-solutions obtained by different optimization algorithms.



| Parameter | Actual value | COA | | PCOA+SSM | | PCOA+CCIC | | PCOA+HSA | |
|---|---|---|---|---|---|---|---|---|---|
| | | Identified value | Relative error | Identified value | Relative error | Identified value | Relative error | Identified value | Relative error |
| $X_d$ | 1.865 | 1.966 | **5.416%** | 1.886 | **1.126%** | 1.878 | **0.697%** | 1.879 | **0.751%** |
| $X'_d$ | 0.2885 | 0.2656 | **7.937%** | 0.2931 | **1.594%** | 0.2841 | **1.525%** | 0.2834 | **1.767%** |
| $X''_d$ | 0.2100 | 0.2002 | **4.667%** | 0.2189 | **4.238%** | 0.2155 | **2.619%** | 0.2146 | **2.190%** |
| $T'_{d0}$ | 1.2000 | 1.2650 | **5.416%** | 1.1878 | **1.016%** | 1.2109 | **0.908%** | 1.2099 | **0.825%** |
| $T''_{d0}$ | 0.1276 | 0.1312 | **2.821%** | 0.1290 | **1.097%** | 0.1302 | **2.037%** | 0.1289 | **1.018%** |
| $K$ | 165.86 | 172.52 | **4.017%** | 167.90 | **1.229%** | 164.15 | **1.031%** | 167.49 | **0.982%** |
| $X_q$ | 1.8150 | 1.8595 | **2.451%** | 1.8081 | **0.380%** | 1.7954 | **1.079%** | 1.8342 | **1.057%** |
| $X''_q$ | 0.2250 | 0.2113 | **6.088%** | 0.2262 | **0.533%** | 0.2281 | **1.378%** | 0.2223 | **1.200%** |
| $T''_{q0}$ | 0.1560 | 0.1491 | **4.423%** | 0.1542 | **1.154%** | 0.1581 | **1.346%** | 0.1545 | **0.961%** |
| $M$ | 8.00 | 8.39 | **4.875%** | 8.10 | **1.250%** | 7.91 | **1.125%** | 8.08 | **1.000%** |
| $D$ | 2.70 | 2.53 | **6.296%** | 2.66 | **1.481%** | 2.74 | **1.481%** | 2.67 | **1.111%** |

**Fig. 10.** Identification results of a synchronous generator by different optimization algorithms.

PCOA+HSA algorithm. From the minimal cost function value and the relative error in Fig. 10, we can know that the proposed hybrid optimization algorithm has superiority over other PCOA algorithms.

## 6. Conclusion

In the present paper, HSA is integrated with PCOA in order to analyze whether it is possible to achieve improved performance for solving optimization problems. Different chaotic maps are compared and the impacts of parallel parameter on the proposed hybrid algorithm are discussed. It is observed that considerable performance improvement is possible by the proposed hybrid algorithm. Benchmarks tests and synchronous generator parameter identification results have shown that the proposed hybrid algorithms has better performance than other PCOA algorithms. In further study, the condition of switch from PCOA to HSA may be discussed.

## References

[1] E.N. Lorenz, Deterministic nonperiodic flow, Journal of Atmospheric Sciences 20 (2) (1963) 130–148.

[2] B. Li, W.S. Jiang, Optimizing complex function by chaos search, Cybernetics and Systems 29 (4) (1998) 409–419.

[3] H. Lu, H. Zhang, L. Ma, A new optimization algorithm based on chaos, Journal of Zhejiang University SCIENCE A 7 (4) (2006) 539–542.

[4] D.X. Yang, G. Li, G.D. Cheng, On the efficiency of chaos optimization algorithms for global optimization, Chaos, Solitons and Fractals 34 (4) (2007) 1366–1375.

[5] X. Yuan, Y. Wang, Parameter selection of support vector machine for function approximation based on chaos optimization, Journal of Systems Engineering and Electronics 19 (1) (2008) 191–197.

[6] D. Yang, Z. Liu, J. Zhou, Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization, Communications in Nonlinear Science and Numerical Simulation 19 (4) (2014) 1229–1246.

[7] M. Alizadeh, M. Alizadeh, S. Ganjefar, Simultaneous coordinated design of PSS and SSSC using improved Lozi map based chaotic optimization algorithm (ILCOA), Neurocomputing 122 (2013) 181–192.

[8] B. Alatas, Chaotic harmony search algorithms, Applied Mathematics and Computation 216 (9) (2010) 2687–2699.

[9] Q. Pan, L. Wang, L. Gao, A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers, Applied Soft Computing 11 (8) (2011) 5270–5280.

[10] Y. Li, Q. Wen, L. Li, H. Peng, Hybrid chaotic ant swarm optimization, Chaos, Solitons and Fractals 42 (2) (2009) 880–889.

[11] M. Wan, C. Wang, L. Li, Y. Yang, Chaotic ant swarm approach for data clustering, Applied Soft Computing 12 (8) (2012) 2387–2393.

[12] Y. Li, Q. Wen, B. Zhang, Chaotic ant swarm optimization with passive congregation, Nonlinear Dynamics 68 (1-2) (2012) 129–136.

[13] L.-Y. Chuang, S.-W. Tsai, C.-H. Yang, Chaotic catfish particle swarm optimization for solving global numerical optimization problems, Applied Mathematics and Computation 217 (16) (2011) 6900–6916.

[14] B. Alatas, E. Akin, A.B. Ozer, Chaos embedded particle swarm optimization algorithms, Chaos, Solitons and Fractals 40 (4) (2009) 1715–1734.

[15] Q. Wu, Hybrid wavelet $\nu$-support vector machine and chaotic particle swarm optimization for regression estimation, Expert Systems with Applications 38 (12) (2011) 14624–14632.

[16] Y.Y. He, J.Z. Zhou, X.Q. Xiang, H. Chen, H. Qin, Comparison of different chaotic maps in particle swarm optimization algorithm for long-term cascaded hydroelectric system scheduling, Chaos, Solitons and Fractals 42 (5) (2009) 3169–3176.

[17] Q. Wu, A self-adaptive embedded chaotic particle swarm optimization for parameters selection of Wv-SVM, Expert Systems with Applications 38 (1) (2011) 184–192.

[18] V.C. Mariani, A.R.K. Duck, F.A. Guerra, L. dos Santos Coelho, R. Venkata Rao, A chaotic quantum-behaved particle swarm approach applied to optimization of heat exchangers, Applied Thermal Engineering 42 (9) (2012) 119–128.

[19] P. Acharjee, S.K. Goswami, Chaotic particle swarm optimization based robust load flow, International Journal of Electrical Power and Energy Systems 32 (2) (2010) 141–146.

[20] Y. He, S. Yang, Q. Xu, Short-term cascaded hydroelectric system scheduling based on chaotic particle swarm optimization using improved logistic map, Communications in Nonlinear Science and Numerical Simulation 18 (7) (2013) 1746–1756.

[21] K. Tatsumi, T. Ibuki, T. Tanino, A chaotic particle swarm optimization exploiting a virtual quartic objective function based on the personal and global best solutions, Applied Mathematics and Computation 219 (17) (2013) 8991–9011.

[22] R. Caponetto, L. Fortuna, S. Fazzino, M. Gabriella, Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Transactions on Evolutionary Computation 7 (3) (2003) 289–304.

[23] X. Yuan, Y. Zhang, Y. Yuan, Improved self-adaptive chaotic genetic algorithm for hydrogeneration scheduling, Journal of Water Resources Planning and Management 134 (4) (2008) 319–325.

[24] Z. Li, X. Wang, Chaotic differential evolution algorithm for solving constrained optimization problems, Information Technology Journal 10 (12) (2011) 2378–2384.

[25] Z. Ma, Chaotic populations in genetic algorithms, Applied Soft Computing 12 (8) (2012) 2409–2424.

[26] I. Pan, S. Das, Chaotic multi-objective optimization based design of fractional order $PI^{\lambda}D^{\mu}$ controller in AVR system, International Journal of Electrical Power and Energy Systems 43 (1) (2012) 393–407.

[27] L. dos Santos Coelho, M.W. Pessôa, A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map, Expert Systems with Applications 38 (11) (2011) 13694–13701.

[28] G.S. dos Santos, L.G. Justi Luvizotto, V.C. Mariani, L. dos Santos Coelho, Least squares support vector machines with tuning based on chaotic differential evolution approach applied to the identification of a thermal process, Expert Systems with Applications 39 (5) (2012) 4805–4812.

[29] M. Hemmati, N. Amjady, M. Ehsan, System modeling and optimization for islanded micro-grid using multi-cross learning-based chaotic differential evolution algorithm, International Journal of Electrical Power and Energy Systems 56 (2014) 349–360.

[30] L. dos Santos Coelho, V.C. Mariani, Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning, Computers and Mathematics with Applications 64 (8) (2012) 2371–2382.

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

[32] Somayeh Alizadeh, Mehdi Ghazanfari, Learning FCM by chaotic simulated annealing, Chaos, Solitons and Fractals 41 (3) (2009) 1182–1190.

[33] Y. Yang, Y. Wang, X. Yuan, F. Yin, Hybrid chaos optimization algorithm with artificial emotion, Applied Mathematics and Computation 218 (11) (2012) 6585–6611.

[34] X. Yuan, Y. Wang, L. Wu, Parallel chaotic optimization algorithm based on competitive-cooperative inter-communication, Control and Decision 22 (9) (2007) 1027–1031.

[35] Q. Zhu, X. Yuan, H. Wang, An improved chaos optimization algorithm-based parameter identification of synchronous generator, Electrical Engineering 94 (3) (2012) 147–153.

[36] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.

[37] O.M. Alia, R. Mandava, The variants of the harmony search algorithm: an overview, Artificial Intelligence Review 36 (1) (2011) 49–68.

[38] J. Majid, K. Esmaile, Two improved harmony search algorithms for solving engineering optimization problems, Communications in Nonlinear Science and Numerical Simulation 15 (11) (2010) 3316–3331.

[39] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Applied Mathematics and Computation 188 (2) (2007) 1567–1579.

[40] Q.-K. Pan, P.N. Suganthan, M. Fatih Tasgetiren, J.J. Liang, A self-adaptive global best harmony search algorithm for continuous optimization problems, Applied Mathematics and Computation 216 (3) (2010) 830–848.

[41] Q.-K. Pan, P.N. Suganthan, J.J. Liang, M. Fatih Tasgetiren, A local-best harmony search algorithm with dynamic sub-harmony memories for lot-streaming flow shop scheduling problem, Expert Systems with Applications 38 (4) (2011) 3252–3259.

[42] A. Baykasoglu, Design optimization with chaos embedded great deluge algorithm, Applied Soft Computing 12 (3) (2012) 1055–1067.

[43] D. He, C. He, L.G. Jiang, H.W. Zhu, G.R. Yu, Chaotic characteristics of a one-dimensional iterative map with infinite collapses, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 48 (7) (2001) 900–906.

[44] M.S. Tavazoei, M. Haeri, Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms, Applied Mathematics and Computation 187 (2) (2007) 1076–1085.

[45] W. Zheng, Kneading plane of the circle map, Chaos Solitons and Fractals 4 (7) (1994) 1221–1233.

[46] R. Thangaraj, M. Pant, A. Abraham, et al., Particle swarm optimization: hybridization perspectives and experimental illustrations, Applied Mathematics and Computation 217 (12) (2011) 5208–5226.

[47] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evolutionary Computation 15 (1) (2007) 1–28.

[48] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Transactions on Evolutionary Computation 10 (6) (2006) 646–657.

[49] A.K. Qin, P.N. Suganthan, Self-adaptive differential evolution algorithm for numerical optimization, 2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005, Proceedings 2 (2005) 1785–1791.