Applied Mathematics and Computation 219 (2012) 3590–3599

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

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

Improved parallel chaos optimization algorithm ☆

Xiaofang Yuan *, Yimin Yang, Hui Wang

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

ARTICLE INFO

Keywords:
Hybrid optimization algorithm
Chaos optimization algorithm (COA)
Harmony search algorithm
Simplex search method

ABSTRACT

Chaos optimization algorithm (COA), which has the features of easy implementation, short execution time and robust mechanisms of escaping from the local optimum, is a promising tool for the engineering applications. The design of approaches to improve the convergence of the COA is a challenging issue. Improved mutative-scale parallel chaotic optimization algorithm (MPCOA) are proposed in this paper, and three ways of improvements for MPCOA are investigated in detail: MPCOA combined with simplex search method, MPCOA based on competitive/cooperative inter-communication, MPCOA combined with harmony search algorithm. Several simulation results are used to show the effective performance of these chaos optimization algorithms.

1. Introduction

Chaos often exists in the nonlinear systems. It is a kind of highly unstable motion of the deterministic systems in finite phase space [1,2]. Chaos is a kind of characteristic which has a bounded unstable dynamic behavior and exhibits sensitive dependence on its initial conditions. An essential feature of the chaotic systems is that small changes in the parameters or the starting values for the data lead to the vastly different future behaviors, such as stable fixed points, periodic oscillations, bifurcations, and ergodicity [2]. This sensitive dependence on the initial conditions is generally exhibited by systems containing multiple elements with non-linear interactions, particularly when the system is forced and dissipative. Sensitive dependence on the initial conditions is not only observed in the complex systems, but even in the simplest logistic equation. The application of the chaotic sequences can be an interesting alternative to provide the search diversity in an optimization procedure, called chaos optimization algorithm (COA) [2–5] (some literatures also called chaotic optimization algorithm [5]). 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 tool for the engineering applications and has attracted much attention [3].

Due to the pseudo-randomness of chaotic motion, the motion step of chaotic variables between two successive iterations is always big, which resulted in the big jump of the design variables in design space [3]. Thus, even if the above COAs in [2–5] have reached the neighborhood of the optimum, it needs to spend much computational effort to approach the global optimum eventually by searching numerous points. Hence, the hybrid methods attract the attention of some researchers, in which the COA was employed for global exploring and other optimization algorithms were used for efficient accurate search. Recently, different kinds of hybrid optimization algorithms using the COA have been presented, such as: chaotic dynamic

☆ This work was supported in part by the National Natural Science Foundation of China (No. 61104088), China Postdoctoral Science Foundation Funded Project (Nos. 20100480937, 201104502), the Planned Science and Technology Project of Hunan Province (No. 2011RS4029), and Young Teachers Promotion Program of Hunan University.

* Corresponding author.
E-mail address: yuanxiaofang126@126.com (X. Yuan).

0096-3003/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.amc.2012.09.053

adaptive local search method [6], chaotic harmony search algorithm [7], hybrid chaotic ant swarm optimization [8], chaotic performance-dependent particle swarm optimization [9], multi-objective chaotic crazy particle swarm optimization algorithm [10], hybrid genetic algorithm with chaotic local search [11], chaotic differential evolution algorithm [12] et al.

The above hybrid optimization algorithms usually can obtain global optimum, but their effectiveness or efficiency are not always good enough due to the COA's sensitive to the initial conditions and the pseudo-randomness of chaotic motion in the COA. The design of approaches to improve the convergence of the COA is a challenging issue. As a typical COA is sensitive to the initial condition and its accurate search ability is not enough, this paper will presents hybrid optimization algorithms as the improvement of the COA.

In [13], we have proposed an improved chaos optimization algorithm, which is the effective combination of mutative-scale parallel chaotic optimization algorithm (MPCOA) and simplex search method (SSM)[14], preserves both the COA's global optimization capability and the SSM's accurate local search ability. In the MPCOA, multiple chaos variables are mapped onto one optimization variable, in this way, the algorithm will not be sensitive to its initial condition. The enable point of SSM is a preset iteration times, and this enable point should be different for different optimization problems. In this paper, two other kinds of hybrid optimization algorithms based on the MPCOA are proposed.

The first kind of hybrid optimization algorithm is the improved MPCOA based on competitive/cooperative inter-communication (CCIC). The main difference between the MPCOA-CCIC and the MPCOA-SSM in [13] is that the former has two ways of searching in addition to the MPCOA, that is, competitive or cooperative inter-communication at each iteration. The cooperative inter-communication is just the SSM local searching in [13] for exploiting, while the competitive inter-communication is a twice carrier chaos optimization search for global exploring. The CCIC can be an efficient supplement of MPCOA to form a hybrid optimization algorithm.

The second kind of hybrid optimization algorithm is the MPCOA combined with harmony search algorithm (HSA). HSA is a recently proposed meta-heuristic optimization algorithm, mimicking the musical process of search for a perfect state of harmony[15–17]. This hybrid optimization algorithm is a two stages search algorithm, the first stage is the MPCOA, and the second stage is the HSA. The MPCOA is conducted until it has converged to a close neighborhood, then the HSA will be conducted for accurate solution at a fast converge speed.

The rest of this paper is organized as follows. Section 2 briefly reviews mutative-scale parallel chaotic optimization algorithm (MPCOA). The MPCOA combined with SSM is introduced in Section 3. Section 4 and Section 5 give presentation of two different kinds of hybrid optimization algorithms, they are MPCOA based on competitive/cooperative inter-communication, MPCOA combined with harmony search algorithm, respectively. Simulation results showing the effectiveness of these hybrid optimization algorithms in Section 6. Conclusions are presented in Section 7.

2. MPCOA approach

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

$$min\, P(x) = P(x_1, x_2, \ldots, x_n), \quad x_i \in [a_i, b_i]. \tag{1}$$

In the COA approach, chaos variables are usually generated by the well-known Logistic map [3]. The Logistic map is defined by function $M(\cdot)$:

$$\gamma(l+1) = M(\gamma(l)), \quad M(\gamma(l)) = \beta\gamma(l)(1-\gamma(l)), \tag{2}$$

where $\beta = 4$.

In the MPCOA, multiple chaos variables are mapped onto one optimization variable, and the search result is the optimal value of parallel multiple chaos variables. In this way, the MPCOA algorithm will not be sensitive to its initial condition. 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 chaos variables.

The process of the MPCOA is illustrated in Fig. 1 and described as follows.

Step 1: Initialize the max iteration times S in the chaos search, random initial value of chaos variables $0 < \gamma_{ij}^{(0)} < 1$.
Step 2: Set iteration times $l = 0$, parallel optimal objective function value $P_j^* = \infty$, and global optimal objective function value $P^* = \infty$.
Step 3: Map chaos variables $\gamma_{ij}^{(l)}$ onto the variance range of the optimization variables by the following equation:

$$x_{ij}^{(l)} = a_i + \gamma_{ij}^{(l)}(b_i - a_i), \tag{3}$$

where

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

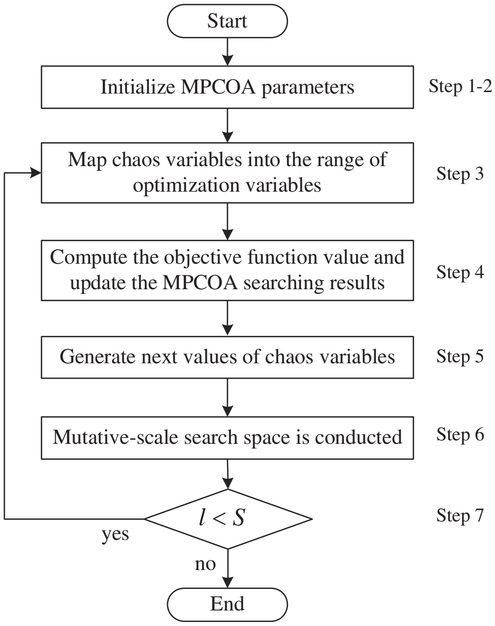

Fig. 1. The structure of MPCOA approach.

Step 4: Compute the objective function value for each optimization variable and update the searching results. If $P_j(x^{(l)}) \leqslant P_j^*$, then $x_j^* = x_j^{(l)}$ and parallel optimal objective function value $P_j^* = P_j(x^{(l)})$. If $P_j^* \leqslant P^*$, then global optimal objective function value $P^* = P_j^*$, and $x^* = x_j^*$. This means that the search result is the optimal value of parallel multiple chaos variables.

Step 5: Generate next values of chaos variables by a chaos mapping function (M):

$$\gamma_{ij}^{(l+1)} = M(\gamma_{ij}^{(l)}). \tag{4}$$

Step 6: Mutative-scale search space is conducted by the following equation:

$$a_i' = x_i^* - \Phi(b_i - a_i), b_i' = x_i^* + \Phi(b_i - a_i), \tag{5}$$

where Φ represents a mutative-scale factor which is a decreasing parameter given by:

$$\begin{aligned} &\Phi = 1 - \left(\frac{l - S_2}{l}\right)^2, \quad l \geqslant S_2, \\ &\Phi = 1, \quad l < S_2, \end{aligned} \tag{6}$$

where S_2 is a positive value set to $(0.05 * S \sim 0.2 * S)$. To avoid a_i', b_i' exceeding ranges $[a_i, b_i]$, the updated ranges are restricted to their bounds: if $a_i' < a_i$, then $a_i' = a_i$; if $b_i' > b_i$, then $b_i' = b_i$.

After these, the search space will be contracted for better accurate search, and the modified search space is used in the follow-up procedure as:

$$a_i = a_i', \quad b_i = b_i'. \tag{7}$$

Step 7: If $l < S$, $l \leftarrow l + 1$ and go to Step 3, otherwise stop the search process.

3. MPCOA combined with SSM

In order to improve the accurate search ability of MPCOA, a local optimization algorithm-simplex search method (SSM)[14] can be employed. In this approach, the MPCOA is used for global search and the conventional optimization algorithm is employed for local search near the global optimum. By integrating SSM within the MPCOA, a more efficient hybrid optimization algorithm is proposed in [13]. The structure of the MPCOA combined with SSM approach is shown in Fig. 2.

In this approach, SSM is employed as the local search for MPCOA, and the enable point of SSM is a preset iteration times S_1. When the iteration times l in MPCOA is bigger than S_1, the SSM is conducted one time, and the local search process using SSM is described as follows. (This local search is conducted just after Step 4 and before Step 5 in the MPCOA in Section 2.)

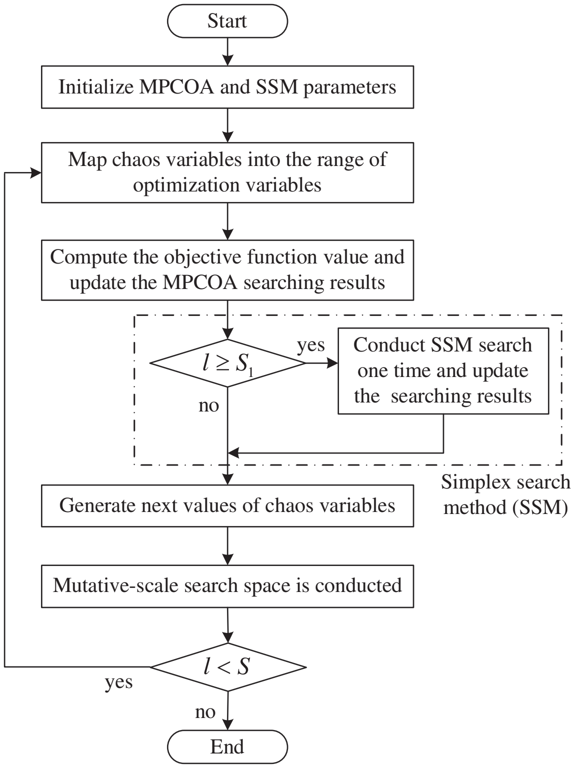

Fig. 2. The structure of the MPCOA with SSM approach.

Step 4.1. Initialization of SSM from the MPCOA search results. The initial vertex points are x_j^*, and the fitness at each vertex point of the simplex as P_j^*.

Step 4.2. Reflection. Determine x_{high} and x_{low}, vertices with the highest and the lowest function values, respectively. Let P_{high} , P_{low} represent the corresponding function values. Find x_{cent}, the center of the simplex without x_{high} in the minimization case. Generate a new vertex x_{refl} by reflecting the worst point as in (8):

$$x_{refl} = 2x_{cent} - x_{high}, \tag{8}$$

If $P_{refl} < P_{low}$ then goto Step 4.3; otherwise goto Step 4.4.

Step 4.3. Expansion. The simplex is expanded in order to extend the search space in the same direction and the expansion point is calculated by:

$$x_{exp} = 2x_{refl} - x_{cent}. \tag{9}$$

The expansion is accepted by replacing x_{high} with x_{exp}; otherwise, x_{refl} replaces x_{high}.

Step 4.4. Contraction. The contraction vertex is calculated as:

$$x_{cont} = 0.5x_{high} + 0.5x_{cent}. \tag{10}$$

The contraction is accepted by replacing x_{high} with x_{cont}; otherwise do shrinking in Step 4.5.

Step 4.5. Shrinkage. If the reflection and contraction have failed, shrinkage attempts to all points except x_{low} by the following equation:

$$x_i \leftarrow 0.5x_i + 0.5x_{low}. \tag{11}$$

Then the SSM search is completed and goto Step 5 of MPCOA in Section 2.

4. MPCOA based on CCIC

For a search process, the exploring searching is good at global optimization, while the exploiting searching is good at accurate optimum. Hence, a good hybrid optimization algorithm may consider both exploring and exploiting searching. Here a MPCOA based on competitive/cooperative inter-communication (CCIC) is proposed as illustrated in Fig. 3. In each iteration, competitive or cooperative inter-communication is used as the supplement for MPCOA searching. The cooperative

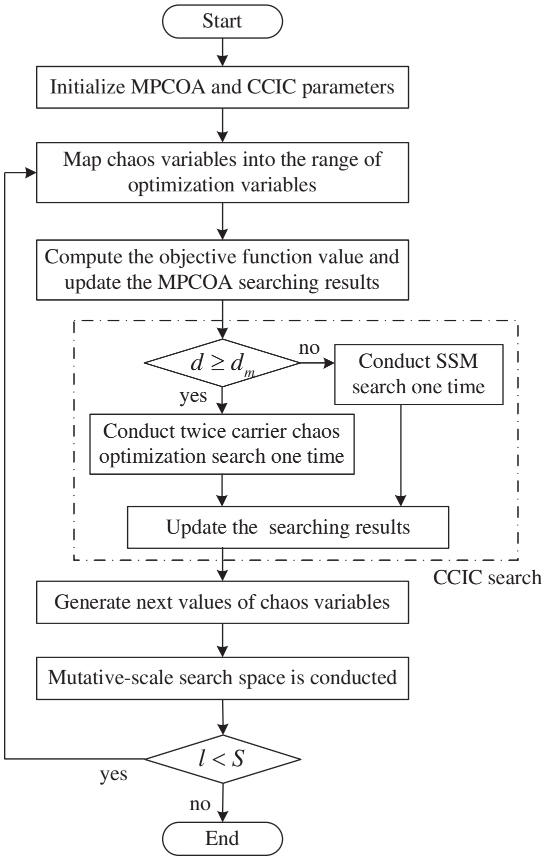

Fig. 3. The structure of the MPCOA based on CCIC.

inter-communication is just the SSM search in [13], which is a local search as a ways of exploiting. The competitive inter-communication is a twice carrier chaos optimization search which is used to explore when the parallel solution are dispersed. Hence both exploring and exploiting searching have been considered in the CCIC approach.

In this approach, the competitive/cooperative inter-communication is conducted according to the dispersion degree d, which is defined as:

$$d = \sqrt{\sum_{j=1}^{N}(x_j^* - x^*)^2}. \tag{12}$$

If $d \geqslant d_m$, the competitive inter-communication is conducted one time, else, the cooperative inter-communication is conducted one time as shown in Fig. 3. Here d_m is a pre-set positive parameter value. The cooperative inter-communication is computed as described in former Section 3, that is, computed from Step 4.1 to Step 4.5.

The competitive inter-communication, which is a process of twice carrier chaos optimization search for exploring based on the current optimal solution x^*, is described as follows:

Step 4.1. Initialization of twice carrier chaos optimization search from the MPCOA search results. Select the current optimal solution x^* as the twice carrier chaos optimization search starts.

Step 4.2. Determine the neighborhood for twice carrier chaos optimization search as:

$$D_i = q(b_i - a_i), \tag{13}$$

where $0 < q < 0.5$ is a coefficient of contraction.

Step 4.3. Map chaos variables $\gamma_{ij}^{(l)} = \beta\gamma_{ij}^{(l-1)}(1 - \gamma_{ij}^{(l-1)})$, where $\beta = 4$, onto the variance range of the optimization variables by the following equation:

$$x_{ij}^{(l)} = x_i^* + D_i * \gamma_{ij}^{(l)} \tag{14}$$

Step 4.4. Compute the objective function value and update the twice carrier chaos optimization search results. If $P_j(x^{(l)}) \leqslant P_j^*$, then $x_j^* = x_j^{(l)}$ and parallel optimal objective function value $P_j^* = P_j(x^{(l)})$. If $P_j^* \leqslant P^*$, then global optimal objective function value $P^* = P_j^*$, and $x^* = x_j^*$.

In this way, the twice carrier chaos optimization search is completed and goto Step 5 of MPCOA in Section 2.

5. MPCOA combined with HSA

Although the simplex search method in the above hybrid optimization algorithms can reach accurate solutions, the converge speed is slow since the simplex search method has slow convergence [14]. In order to improve the converge speed of MPCOA, a heuristic algorithm- harmony search algorithm (HSA)[15–17] is employed here. By combining HSA with the MPCOA, a more efficient hybrid optimization algorithm can be developed. The structure of the MPCOA combined with HSA is illustrated in Fig. 4.

Since MPCOA is an effective global search method and it is not sensitive to its initial conditions, after many iteration times, parallel optimal objective function value P_j^* will converge to a close neighborhood, and P_j^* will be close to global optimal objective function value P^*. Here $\|P_j^* - P^*\| < P_{NB}$ means whether the parallel search results of MPCOA have reach a close neighborhood, and $\|P_j^* - P^*\| < P_{NB}$ is computed as:

$$|P_1^* - P^*| < P_{NB} \; and \; |P_2^* - P^*| < P_{NB} \ldots \text{ and } |P_j^* - P^*| < P_{NB}, \tag{15}$$

where P_{NB} is a pre-set value. This condition of switch from the MPCOA to the HSA is feasible as the MPCOA has effective global search ability. Then HSA will be conducted for accurate search since it uses a stochastic random search based on the harmony memory considering rate and the pitch adjusting rate.

As the HSA is the successor of the MPCOA, so the parallel optimal objective function value P_j^* is the start point of HSA, that is, x_j^* in MPCOA search is the initial harmony memory (HM) for HSA. This section describes the HSA as the successor of MPCOA as follows.

Step 8. Initialize HSA parameters: harmony memory size (HMS), maximum number of improvisations (MaxImp), harmony memory considering rate (HMCR), pitch adjusting rate (PAR), bandwidth vector used in (BW).

Step 9. Initialize the harmony memory (HM) from the MPCOA search results.

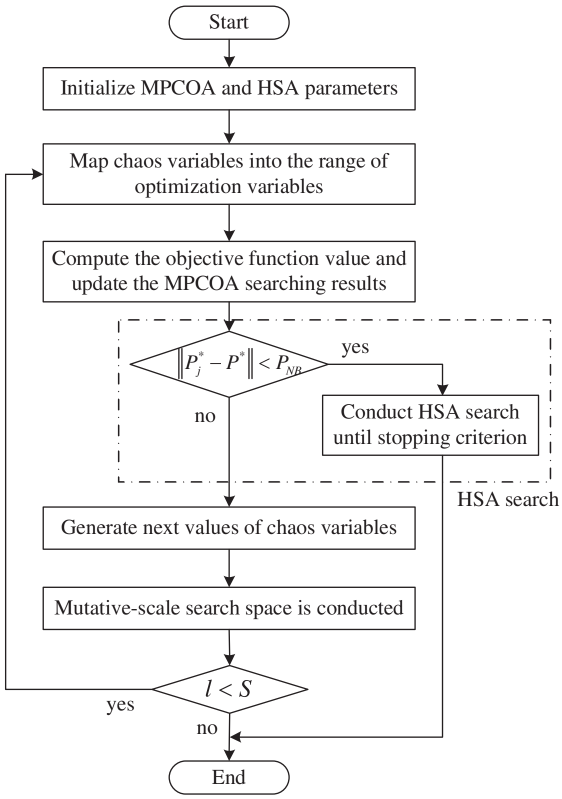

Fig. 4. The structure of the MPCOA combined with HSA.

$$HM^1 = x_1^*, \quad HM^2 = x_2^*, \ldots, HM^j = x_j^* \tag{16}$$

Step 10. Improvise a new harmony from the HM. $x' = (x'_1, x'_2, \ldots, x'_n)$ is improvised based on the following three mechanisms: 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 historical values stored in the HM, and $(1 - HMCR)$ is the rate of randomly selecting one value from the possible range of values.

$$\begin{aligned} &x'_i = x'_i \in \{x_i^1, x_i^2, \ldots, x_i^{HMS}\} \text{ with probability HMCR}, \\ &x'_i = x'_i \in x_i \text{ with probability } (1 - HMCR). \end{aligned} \tag{17}$$

The value of each decision variable obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. This operation uses the PAR parameter, which is the rate of pitch adjustment as it should be pitch-adjusted to neighboring pitches with a probability of $HMCR \times PAR$, while the original pitch obtained in the memory consideration is kept with a probability of $HMCR \times (1 - PAR)$. 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:

$$\begin{aligned} &x'_i = x'_i \pm u(-1,+1) \times BW \text{ with probability HMCR} \times \text{PAR}, \\ &x'_i = x'_i \text{ with probability HMCR} \times (1 - \text{PAR}). \end{aligned} \tag{18}$$

Step 11. Update the HM. If the new harmony vector is better than the worst harmony vector in the HM, based on the evaluation of the objective function value, the new harmony vector is included in the HM, and the existing worst harmony vector is excluded from the HM.

Step 12. If the stopping criterion (or maximum number of improvisations) is satisfied, the hybrid optimization algorithm is terminated. Otherwise, Step 10 and Step 11 are repeated.

6. Simulation

6.1. Performance of these hybrid optimization algorithm

The efficiency and performance of the above three hybrid optimization algorithms with the following five nonlinear functions [18,19] is evaluated:

$$f_1(x,y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2, \quad -200 < x,y < 200, \tag{19}$$

$$f_2(x,y) = 0.5 - \frac{sin^2\sqrt{x^2+y^2} - 0.5}{(1 + 0.001(x^2+y^2))^2}, \quad -200 < x,y < 200, \tag{20}$$

$$f_3(X) = \sum_{i=1}^{3}(x_i^2 - 10cos(2\pi x_i) + 10), \quad -5 < x_i < 5, \quad i = 1,2,3, \tag{21}$$

$$f_4(X) = \frac{1}{4000}\sum_{i=1}^{5}x_i^2 - \prod_{i=1}^{5}cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad -5 < x_i < 5, \quad i = 1,2,\ldots,5, \tag{22}$$

$$f_5(X) = \frac{1}{5}\sum_{i=1}^{5}(x_i^4 - 16x_i^2 - 5x_i), \quad -10 < x_i < 10, \quad i = 1,2,\ldots,5. \tag{23}$$

Function f_1 is the Camel function, which has six local minima and two global minima $x^* = (-0.0898, -0.7126)$, $x^* = (0.0898, -0.7126)$, and optimal objective function value $f^* = -1.031628$. Function f_2 is the Schaffer's function, which has infinite local maxima and one global maximum $x^* = (0,0)$, and $f^* = 1.0$. Function f_3 is the Rastrigin's function, which has many local minima and one global minimum $x^* = (0,0,0)$, and $f^* = 0$. Function f_4 is the Griewank's function, which has several thousand local minima and one global minimum $x^* = (0,0,\ldots,0)$, and $f^* = 0$. Function f_5 has five variables, which has 32 local minima and one global minimum $x^* = (2.9051, 2.9051, \ldots, 2.9051)$, and $f^* = -78.3323$. These five nonlinear multi-modal functions are often used to test the convergence, efficiency and accuracy of optimization algorithms [18,19]. Among them the former three functions have two or three design variables, which is 2- or 3-dimensional problem, and the latter two have five variables which is 5-dimensional problem. The parameters is these algorithms are: $S = 600$, $S_2 = 150$, the number of multiple chaos variables N are shown in Fig. 5.

Computational results for Function f1 (optimal objective function value -1.031628)

Algorithm	N=5		N=10		N=20	
	Best value	Probability	Best value	Probability	Best value	Probability
MPCOA+SSM	-1.031154	86.7%	-1.031599	93.3%	-1.031626	100%
MPCOA+CCIC	-1.031280	86.7%	-1.031603	93.3%	-1.031627	100%
MPCOA+HSA	-1.031374	90.0%	-1.031620	96.7%	-1.031628	100%

Computational results for Function f2(optimal objective function value 1.0)

Algorithm	N=5		N=10		N=20	
	Best value	Probability	Best value	Probability	Best value	Probability
MPCOA+SSM	0.99918	83.3%	0.99997	96.7%	1.0	100%
MPCOA+CCIC	0.99957	80.0%	0.99998	96.7%	1.0	100%
MPCOA+HSA	0.99939	86.7%	0.99999	100%	1.0	100%

Computational results for Function f3(optimal objective function value 0)

Algorithm	N=5		N=10		N=20	
	Best value	Probability	Best value	Probability	Best value	Probability
MPCOA+SSM	0.00095	50.0%	0.00011	83.3%	0.00001	93.3%
MPCOA+CCIC	0.00099	56.7%	0.00009	86.7%	0.00001	93.3%
MPCOA+HSA	0.00087	56.7%	0.00006	90.0%	0.00001	96.7%

Computational results for Function f4(optimal objective function value 0)

Algorithm	N=5		N=10		N=20	
	Best value	Probability	Best value	Probability	Best value	Probability
MPCOA+SSM	0.00167	43.3%	0.00121	73.3%	0.00042	93.3%
MPCOA+CCIC	0.00283	53.3%	0.00109	76.7%	0.00051	96.7%
MPCOA+HSA	0.00195	50.0%	0.00089	80.0%	0.00037	93.3%

Computational results for Function f5(optimal objective function value -78.3323)

Algorithm	N=5		N=10		N=20	
	Best value	Probability	Best value	Probability	Best value	Probability
MPCOA+SSM	-78.3310	53.3%	-78.3319	80.0%	-78.3322	96.7%
MPCOA+CCIC	-78.3312	60.0%	-78.3320	83.3%	-78.3322	96.7%
MPCOA+HSA	-78.3312	56.7%	-78.3321	80.0%	-78.3322	96.7%

Fig. 5. Computational results for function f1–f5.

The three hybrid optimization algorithms are represented as follows respectively. MPCOA + SSM, refers to the MPCOA combined with SSM in Section 3; MPCOA + CCIC, refers to the MPCOA based on CCIC in Section 4; MPCOA + HSA, refers to the MPCOA combined with HSA in Section 5. In this simulation, the hybrid optimization algorithms are repeated for 30 times, and the simulation results ate shown in Fig. 5. In Fig. 5, N is the parallel number, the 'best value' means the optimal objective function value f^* in 30 times searching for each algorithm, the 'probability' means the probability of reaching the optimum x^* with the error less than 0.01. In can seen from Fig. 5 that, as the increase of parallel number N, the search results is better. When N is 20, these hybrid optimization algorithms can reach satisfied results. So the parallel number N may be about $15 \sim 20$ for the proposed hybrid optimization algorithms as this can compromise the computation time and the optimal search results. It also can seen from Fig. 5 that the MPCOA + HSA has best search results compared with other optimization algorithms, especially for multi-dimensional function f_4 and f_5. Fig. 6 has shown the objective function values for f1–f5 for these hybrid optimization algorithms with $N = 15$.

Parameter	Actual value	COA		MPCOA+SSM		MPCOA+CCIC		MPCOA+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%**	1.884	**1.019%**
$X_d^{'}$	0.2885	0.2656	**7.937%**	0.2931	**1.594%**	0.2841	**1.525%**	0.2914	**1.005%**
$X_d^{''}$	0.2100	0.2002	**4.667%**	0.2189	**4.238%**	0.2155	**2.619%**	0.2176	**3.619%**
$T_{d0}^{'}$	1.2000	1.2650	**5.416%**	1.1878	**1.016%**	1.2109	**0.908%**	1.2126	**1.050%**
$T_{d0}^{''}$	0.1276	0.1312	**2.821%**	0.1290	**1.097%**	0.1302	**2.037%**	0.1293	**1.332%**
K	165.86	172.52	**4.017%**	167.90	**1.229%**	164.15	**1.031%**	163.99	**1.127%**
X_q	1.8150	1.8595	**2.451%**	1.8081	**0.380%**	1.7954	**1.079%**	1.8241	**0.501%**
$X_q^{''}$	0.2250	0.2113	**6.088%**	0.2262	**0.533%**	0.2281	**1.378%**	0.2239	**0.489%**
$T_{q0}^{''}$	0.1560	0.1491	**4.423%**	0.1542	**1.154%**	0.1581	**1.346%**	0.1577	**1.089%**
M	8.00	8.39	**4.875%**	8.10	**1.250%**	7.91	**1.125%**	8.10	**1.250%**
D	2.70	2.53	**6.296%**	2.66	**1.481%**	2.74	**1.481%**	2.75	**1.851%**

Fig. 6. Identification results of a synchronous generator.

Time	MPCOA+SSM	MPCOA+CCIC	MPCOA+HSA
Max	516	476	395
Min	377	343	326
Average	441	405	369

Fig. 7. Searching time for different optimization algorithms (in seconds).

6.2. Parameter identification of synchronous generator

In this section, simulations are performed to evaluate the performance of these hybrid optimization algorithms for parameter identification of synchronous generator as in [13]. 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 – 50 Hz. The fitness function and the learning data-set are the same as in [13].

The parameters of the hybrid optimization algorithms are chosen as: $S = 600,\ S_2 = 150$, the number of variables $n = 11$, the number of multiple chaos variables $N = 20$.

The results in Fig. 6 are the best results of each optimization algorithm repeated for 30 times. The results of the COA and the MPCOA + SSM are mentioned in [13]. From the identification results, it is seen that the relative error using these hybrid optimization algorithms can be greatly less than 5%, while the relative error using the COA is bigger than 5%. Compared with parameter identification approach in [20], these hybrid optimization algorithms have better performance. This identification results also show that the proposed MPCOA + CCIC and MPCOA + HSA algorithm have less relative errors compared with the MPCOA + SSM approach, while there is minor error difference between the MPCOA + CCIC and the MPCOA + HSA.

The searching time of these optimization algorithms repeated 30 times are shown in Fig. 7, which shows that the proposed proposed MPCOA + CCIC and MPCOA + HSA algorithm spend less time than that of MPCOA + SSM. This also can means that the MPCOA + HSA has less searching time than that of MPCOA + CCIC. So the simulation results show that the proposed MPCOA + HSA has good performance both in computation time and optimal search results.

7. Conclusion

Improved mutative-scale parallel chaotic optimization algorithm (MPCOA) are proposed in this paper, and three ways of improvements for PCOA are investigated in detail: MPCOA combined with simplex search method, MPCOA based on competitive/cooperative inter-communication, MPCOA combined with harmony search algorithm. Several simulation results are used to show the effective performance of these chaos optimization algorithms.

References

[1] Yimin Yang, Yaonan Wang, Xiaofang Yuan, Feng Yin, Hybrid chaos optimization algorithm with artificial emotion, Applied Mathematics and Computation 218 (11) (2012) 6585–6611.
[2] Huijuan Lu, Huoming Zhang, Longhua Ma, A new optimization algorithm based on chaos, Journal of Zhejiang University SCIENCE A 7 (4) (2006) 539–542.
[3] 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.
[4] Xiaofang Yuan, Yaonan 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.
[5] H. Shayeghi, H.A. Shayanfar, S. Jalilzadeh, A. Safari, Multi-machine power system stabilizers design using chaotic optimization algorithm, Energy Conversion and Management 51 (7) (2010) 1572–1580.
[6] Zhiqiang Zhang, Zheng Tang, Shangce Gao, Gang Yang, An algorithm of chaotic dynamic adaptive local search method for Elman neural network, International Journal of Innovative Computing, Information and Control 7 (2) (2011) 647–656.
[7] B. Alatas, Chaotic harmony search algorithms, Applied Mathematics and Computation 216 (9) (2010) 2687–2699.
[8] Yuying Li, Qiaoyan Wen, Lixiang Li, Haipeng Peng, Hybrid chaotic ant swarm optimization, Chaos, Solitons and Fractals 42 (2) (2009) 880–889.
[9] Zhihua Cui, Xingjuan Cai, Jianchao Zeng, Chaotic performance-dependent particle swarm optimization, International Journal of Innovative Computing, Information and Control 5 (4) (2009) 951–960.
[10] B.B. Firouzi, H.Z. Meymand, T. Niknam, H.D. Mojarrad, A novel multi-objective chaotic crazy PSO algorithm for optimal operation management of distribution network with regard to fuel cell power plants, International Journal of Innovative Computing, Information and Control 7 (11) (2011) 6395–6409.
[11] Yue Tan, Guanzheng Tan, Xuedong Wu, Hybrid real-coded genetic algorithm with chaotic local search for global optimization, Journal of Information and Computational Science 8 (14) (2011) 3171–3179.
[12] Zhiyong Li, Xiang Wang, Chaotic differential evolution algorithm for solving constrained optimization problems, Information Technology Journal 10 (12) (2011) 2378–2384.
[13] Qing Zhu, Xiaofang Yuan, Hui Wang, An improved chaos optimization algorithm-based parameter identification of synchronous generator, Electrical Engineering 94 (3) (2012) 147–153.
[14] K.I.M. McKinnon, Convergence of the Nelder–Mead simplex method to a nonstationary point, SIAM Journal on Optimization 9 (1) (1998) 148–158.
[15] Zong Woo Geem, Joong Hoon Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[16] O.M. Alia, R. Mandava, The variants of the harmony search algorithm: an overview, Artificial Intelligence Review 36 (1) (2011) 49–68.
[17] Jaberipour Majid, Khorram Esmaile, Two improved harmony search algorithms for solving engineering optimization problems, Communications in Nonlinear Science and Numerical Simulation 15 (11) (2010) 3316–3331.
[18] B. Li, W.S. Jiang, Optimizing complex function by chaos search, Cybernetics and Systems 29 (4) (1998) 409–419.
[19] T. Zhang, H.W. Wang, Z.C. Wang, Mutative scale chaos optimization algorithm and its application, Control and Decision 14 (3) (1999) 285–288.
[20] L.X. Sun, P. Qu, Q.X. Huang, P. Ju, Parameter identification of synchronous generator by using ant colony optimization algorithm, in: ICIEA 2007 – Second IEEE Conference on Industrial Electronics and Applications, vol. 1, 2007, pp. 2834–2838.