

Novel Hybrid Optimization Algorithm for Parameter Estimation of Chaotic System

Haotian Chang^{1, a}, Jing Feng^{1, b} and Lei Jiang^{1, c}

¹College of Meteorology and Oceanography, PLA Univ. of Sci. & Tech., Nanjing, 211101, China

^achanghaotian901120@126.com, ^bjfeng@seu.edu.cn, ^c18551622511@163.com

Keywords: chaotic system, parameter estimation, hybrid optimization algorithm

Abstract. This paper proposes a novel hybrid optimization algorithm of Adaptive Cuckoo Search and Particle Swarm Optimization algorithm for parameter estimation of chaotic system. In order to enhance the accuracy and efficiency of ACS, the strategy of exploitation velocity adjustment via acceleration by distance of PSO algorithm is adopted. Thus, the algorithms mentioned above are used for estimation of the parameters of Lorenz chaotic system. Estimation result from each algorithm generates a standard deviation with the true parameter data, which is regarded as the fitness. Compared with ACS and PSO algorithm, the hybrid optimization algorithm is more efficient and accurate for parameter estimation, thus benefitting the simulation and control of chaotic systems.

Introduction

Chaotic systems are generated by determined equations, but the outputs follow completely unpredictable stochastic patterns^[1]. Besides, chaos systems are sensitive to the initial state, just as the well-known butterfly reaction indicates. In many occasions of engineering and industry which are involved with chaotic systems, we are always trying to control and tune chaos, and the first necessity of control and synchronization of chaos is to acquire the parameters^[2]. Nevertheless, it is not realistic and practical to measure the system parameters directly. However, it is feasible and practical to estimate system parameters based on observation and analysis of the chaotic scalar time series.

Studies of parameter estimation methods and various optimization algorithms have been carried out by precursors^[3-5]. Optimization process is time-consuming and its computational cost is usually high. In certain models, the computation costs are mainly determined by the optimization algorithms. Therefore, optimization algorithms are the keys to solving the optimization problems with an aim to acquire its optimality^[6]. The need to find the optimality has led to the development of various optimization algorithms such as metaheuristics^[7-8]. Metaheuristic algorithms, such as genetic algorithms (GA), Taboo Search (TS), ant colony optimization (ACO) and Particle Swarm Optimization (PSO)^[9], have proved their potential and efficiency in solving various optimization problems. Among those algorithms, an outstanding and robust nature-inspired meta-heuristic algorithm named Cuckoo Search was proposed by Yang and Deb^[10]. Compared with GA and PSO, Cuckoo Search via Levy Flight acquires more efficient randomization which prevents the algorithm from being trapped in local optima^[11-12]. Additionally, Cuckoo Search requires less parameters to be set than GA and PSO do. However, Cuckoo Search has a poor capability of intensification due to its large degree of randomization and its convergence rate is low because of its fixed value of parameters. The adaptive cuckoo search (ACS) algorithm brings about improvement of the convergence rate by adjusting parameters in each time of iteration, but the intensification capability still needs enhancement. Hence, we adopt the strategy of exploitation velocity adjustment via acceleration by distance of PSO to enhance the intensification capability of ACS.

Hybrid Optimization Algorithm of ACS and PSO

Most of the nature-inspired meta-heuristic algorithms are powerful because they imitate the best feature in nature which can be regarded as two critical characteristics: intensification and diversification^[10]. Intensification refers to the capability of searching the vicinity of the current best local optima and finding the best solutions, while diversification ensures that the solution space is explored thoroughly. Some classic optimization algorithms, such as GA and PSO, both performs relatively poor in aspect of diversification, which leads to the algorithms being trapped in local optima and ending search immaturely. However, Cuckoo Search via Levy Flight acquires more efficient randomization which prevents the algorithm from being trapped in local optima. Therefore, the strategy of exploitation velocity adjustment via acceleration by distance of PSO is embedded into ACS so as to achieve improvement of both intensification and diversification capability of the hybrid algorithm.

Basic Cuckoo Search Algorithm

Parasitism breeding is the most well-known behavior of cuckoos. Cuckoos lay their eggs in other host birds' nests and remove the host birds' eggs to increase the hatching probabilities of their eggs. In response to this invasive breeding behavior, host birds develop the ability to discover fake eggs and either remove them or abandon the nest and build a new one somewhere else. For simplicity in modeling the basic CS algorithm, three basic principles^[10] are set: (1) each cuckoo lays one egg every time and dumps it in a host nest randomly, (2) better nests with higher quality of eggs will be preserved to next generation, (3) the number of available nests are certain and a fake egg is discovered by the host birds with a probability of $p_a \in [0,1]$. If a fake egg is discovered, the nest will be abandoned and the host bird will rebuild a nest somewhere else.

In consideration of diversification capability, CS algorithm uses Levy flight as an outstanding global random walk strategy. The global random walk can be described as:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \alpha * Levy(s, \lambda) \quad (1)$$

where $\mathbf{x}_i^t = (x_1, x_2, \dots, x_D)$ is the D-dimension location of nest i in t -th iteration, α is the step size scaling factor. And $Levy(s, \lambda)$ is the step lengths that are distributed by the probability distribution as shown in Equation (2):

$$Levy(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi s^{1+\lambda}} \quad (2)$$

where s represents the basic step length.

Levy flight distribution promises the global random walk and prevents the CS algorithm from being trapped in local optima effectively.

Meanwhile, CS algorithm also adopts a local random walk strategy when the nests with fake eggs are discovered and abandoned. Then, locations of the new nests will be generated as follows:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + r_u \cdot (\mathbf{x}_j^t - \mathbf{x}_k^t) \otimes H(p_a - r_u) \quad (3)$$

where $r_u \sim U(0,1)$, $r_u \sim U^D(0,1)$, \mathbf{x}_j^t and \mathbf{x}_k^t are two different solutions selected randomly and H refers to the Heaviside function, and $U^D(0,1)$ is the uniform distribution defined in the range of $[0,1]$.

Adaptive Cuckoo Search

It is found that the performance of basic CS algorithm in intensification capability is closely correlated to the parameter p_a and α . Since the two parameters are fixed in basic CS, adaptive parameter adjustment is integrated into the basic CS algorithm to enhance the exploitation performance^[2].

Relatively large p_a and small α contributes to increase in convergence rate, but also leads to lower quality of solution and even disability in searching the optima. Contrarily, small value of p_a and large α decreases the convergence speed. The adaptive parameter adjustment can be expressed as:

$$\alpha(n) = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \sqrt{1 - \frac{n}{N}} \quad (4)$$

$$p_a(n) = p_{a,\max} - \frac{(p_{a,\max} - p_{a,\min})n}{N} \quad (5)$$

where n and N refer to the current iteration number and max iteration number, respectively. From Equation (4) and (5) can be seen that values of p_a and α decreases gradually as the iteration times increases. Therefore, in early iterations the convergence speed is relatively high while the accuracy of solutions increases in late iterations, which results in the balance of overall convergence speed and the accuracy of solutions. However, as mentioned above, CS performs relatively poor in intensification capability, thus requiring further enhancement of searching in the vicinity of the current local optima. Strategy of velocity adjustment via acceleration by distance of PSO

PSO algorithm has been a classic optimization algorithm which was first proposed in 1995 by Kennedy and Eberhart. The algorithm is relatively weak in search the global optima while it performs well in exploiting local optima and has a high convergence speed. In other words, PSO is able to help develop the intensification capability of ACS. So we intend to integrate the velocity adjustment strategy^[9] into ACS to enhance the capability of exploiting in the vicinity of current local optima. The velocity of exploitation is renewed in each iteration time according to the best ‘remembered’ solution of each particle, p_{best} , and the best solution that all the particles have ever found, g_{best} . The exploiting velocity adjustment is carried out via acceleration by distance which can be expressed in Equation (6) and (7):

$$v_i(t+1) = v_i(t) + r_1(P_i - x_i(t)) + r_2(P_g - x_i(t)) \quad (6)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (7)$$

where r is a random factor drawn uniformly between $[0,1]$, $v_i = (v_1, v_2, \dots, v_D)$ represents the location increment of the i -th particle, x_i represent location of the i -th particle, t is the number of iteration, P_i is the best solution acquired by the i -th particle, and P_g is the best solution that all the particles have ever found.

Hybrid optimization algorithm of ACS and PSO

In the hybrid algorithm, all the locations of particles are given from the nests which are found by ACS and each location of nest is the ever best solution of its nest. So Equation (6) can be simplified as:

$$v_i(t+1) = v_i(t) + r(P_g - x_i(t)) \quad (8)$$

From Equation (8), it can be found that the velocity increment is mainly affected by the current optima and all the particles tend to move around the vicinity of the current optima, thus attempting to enhance the intensification capability of the algorithm.

The procedure of the hybrid algorithm of ACS and PSO is presented as follows.

Step 1: Initialize the location of each cuckoo’s nest with a D-dimension array $x_i^F = (x_1, x_2, \dots, x_D)$, ($t = 0$), and evaluate the initial fitness value of each cuckoo’s nest, F_i , using Equation 5. And initialize p_a and α .

Step 2: Generate new locations of nest by Levy flight and evaluate their fitness values, F_{i_new} . If $F_{i_new} < F_i$, replace nest i with the new solution.

Step 3: Rebuild nests randomly with the probability of p_a and evaluate their fitness values, F_{j_new} . If the new nests have better fitness, $F_{j_new} < F_j$, keep the new nest j and abandon the old solution.

Step 4: Update values of p_a and α . Record the current best nest and pass it to the parameter P_g .

Step 5: All nests are regarded as particles in this step and velocity increments are acquired using Equation 13. New location of each particle is acquired using Equation 12 and the fitness of each particle is calculated. New particles with better fitness values are preserved to next generation, otherwise they remain in old locations.

Step 6: End the procedure if the stop criterion is met. Otherwise, go to Step 2.

Parameter Estimation of Lorenz Chaotic System

Parameter estimation can be regarded as multi-dimensional optimization with proper fitness functions. A continuous nonlinear chaotic system can be described by Equation (13):

$$\dot{X} = F(X, X_0; \theta) \quad (9)$$

where $X = (x_1, x_2, \dots, x_n)^T \in R^n$ denotes the state vector of the system, \dot{X} is derived from X , X_0 represents the initial state of system, and $\theta = (\theta_1, \theta_2, \dots, \theta_d)$ is a set of parameters.

Under the hypothesis that the system function is known, the estimation of state vector is described as:

$$\dot{\tilde{X}} = F(\tilde{X}, X_0, \tilde{\theta}) \quad (10)$$

where \tilde{X} denotes the estimation of the state vector of system and $\tilde{\theta}$ is a set of estimation parameters. To value the fitness of the estimation, we use standard deviation function as :

$$F(\tilde{\theta}) = \sqrt{\frac{1}{M} \sum_{i=1}^M (X - \tilde{X})^2} \quad (11)$$

where M is the length of data used for estimation. The acquired global minimum fitness occurs when the fittest estimation of parameters is made.

In this paper, we choose Lorenz chaotic system as our research paradigm, so the optimization task can be more specifically expressed as follows,

$$\begin{cases} \dot{x} = \sigma_1(y - x) \\ \dot{y} = \sigma_2 x - xz - y \\ \dot{z} = xy - \sigma_3 z \end{cases} \quad (12)$$

where (x, y, z) are the state variables and $(\sigma_1, \sigma_2, \sigma_3)$ are the unknown parameters of the Lorenz chaotic system. The true values of the parameters are set as $\sigma_1 = 10$, $\sigma_2 = 28$, $\sigma_3 = 8/3$. When parameter estimation is acquired, the fitness of estimation is valued by the fitness function mentioned above with true values of parameters. And the fitness function can be expressed as:

$$F = \sqrt{\frac{1}{M} \sum_{i=1}^M ([\tilde{x}(n) - x(n)]^2 + [\tilde{y}(n) - y(n)]^2 + [\tilde{z}(n) - z(n)]^2)} \quad (13)$$

It can be seen that it is not easy to seek the global optima of the nonlinear function with multiple local optima accurately using traditional methods.

In order to acquire objective statistical results, we run each of PSO, ACS and hybrid algorithm of ACS and PSO 100 times and the max iteration number is 100. Average fitness, best fitness and average computing time are presented as the final results, as shown in Table 1.

Table 1. Parameter Estimation of Lorenz chaotic system

Algorithm	Average fitness	Best fitness	Computing time
PSO	1.2347	0.0184	1.2309s
ACS	5.4287e-4	2.7335e-5	2.8484s
ACS-PSO	4.2176e-4	3.2564e-6	3.9571s

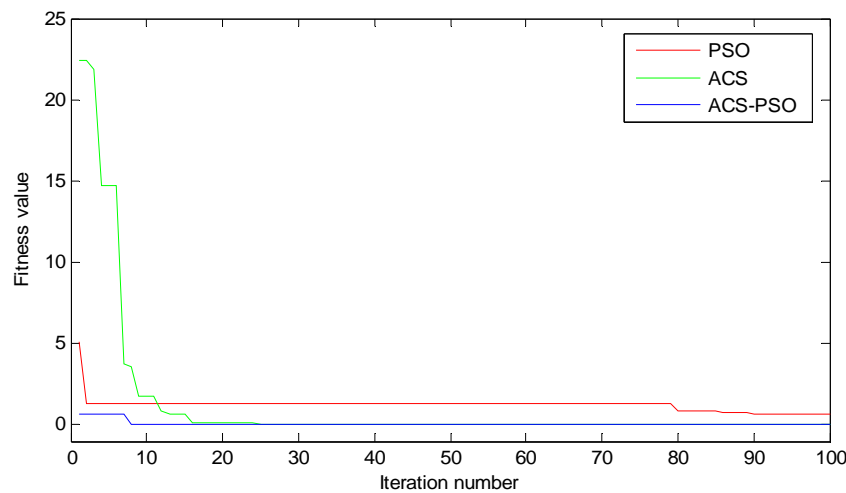


Fig.1 Convergence speed of the compared algorithms

It can be seen from Table 1 that the hybrid optimization algorithm obviously outperforms the comparison algorithms. More specifically, the average fitness and the best fitness value are both better than those of the other comparison algorithms. As the average fitness value almost 22% less than ACS, it can be concluded that the accuracy and intensification capability of the ACS algorithm is enhanced obviously by integration of PSO algorithm. The best solution of the hybrid optimization algorithm of ACS and PSO is **(9.9987, 28.0001, 2.6667)** while the true values of the system parameters are **(10, 28, 8/3)** as mentioned above.

The convergence speeds of the three algorithms are depicted in Figure 2. It is obvious that the hybrid algorithm outperforms the other compared algorithm. More significantly, the hybrid algorithm converges to a very low level of fitness value much earlier than ACS does. Therefore, our aim of enhancing the exploiting intensification capability of the ACS by integrating PSO is achieved.

Conclusions

In this paper, a novel hybrid optimization algorithm is proposed for parameter estimation of chaotic systems. The algorithm integrates the adaptive cuckoo search and particle swarm optimization algorithm. In order to verify the performance of the hybrid algorithm, we choose the application of parameter estimation of Lorenz chaotic system as our research object. The results indicate better accuracy of estimation and intensification capability of the proposed algorithm, compared with ACS and PSO. In general, the proposed algorithm is an efficient and accurate and shows an enhanced intensification capability compared with ACS when applied to parameter estimation of chaotic systems.

Acknowledgements

The work of this paper is supported by the National Natural Science Foundation of China (Grant No. 61371119).

References

- [1] Parlitz U 1996 Phys. Rev. Lett. 76(8) 1232
- [2] Wang J and Zhou B H 2015 Neural Compt & applic. DOI: 10.1007/s00521-015-1949-1.
- [3] Gandomi A H, Yang X S and Alavi A H 2011 Comput. Struct. 89(23-24) 2325
- [4] Chao T, Yu Z and Jack JJ 2007 Phys. Rev. E 76(1) 016209
- [5] Yao L and Sethares W A 1994 IEEE Trans Signal Process 42(4) 927
- [6] Yang X S and Deb S 2014 Neural Compt. & Applic. 24 169
- [7] Blum C and Roli A 2003 ACM Comput. Surveys (CSUR) 35(3) 268
- [8] Glover F and Kochenberger G A 2003 Handbook of metaheuristics (New York: Springer)
- [9] Kennedydy J and Eberart R 1995 Proceedings of the IEEE International Conference on Neural Networks (Piscataway, NJ, USA: IEEE) p.1942
- [10] Yang X and Deb S 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). (USA: IEEE Publications) p. 210
- [11] Gandomi A H, Yang X S, Talatahari S and Deb S 2012 Comput. Math Appl. 63(1) 191
- [12] Long W, Liang X and Huang Y F 2014 Neural Comput. Appl. 25(3-4) 911