

A Memory-Based Particle Swarm Optimization for Parameter Identification of Lorenz Chaotic System



Rizk M. Rizk-Allah , M. A. Farag, Mahmoud H. Barghout, and Aboul Ella Hassanien

Abstract A novel modified version of particle swarm optimization (PSO) is introduced in this paper to estimate the parameters of the chaotic Lorenz system. The parameters estimation of the Lorenz system is modeled as a multidimensional problem and solved by the proposed algorithm, a memory-based particle swarm optimization (MbPSO) algorithm. In MbPSO, two new terms are added to the standard PSO to vary the population direction and enhance search capability. Firstly, the impact of parameter configuration on MbPSO is studied. After that, the parameter estimation problem is solved. The performance of the proposed MbPSO is compared with other meta-heuristic algorithms in terms of parameter accuracy and convergence speed. According to the results, linking the memory of each particle to the memory of other particles has a very significant effect on the proposed algorithm compared to the original PSO. Briefly, the MbPSO algorithm is a successful and powerful optimization algorithm for parameter estimation of chaotic systems with accurate performance.

Keywords Chaotic system · Lorenz system · Parameter estimation · Particle swarm optimization

1 Introduction

Nonlinear dynamics, especially the study of chaotic systems (CSs), have been increasing interest in various fields, including science, engineering, communication, biomedical, finance, administration, and other areas [1]. The chaotic system,

R. M. Rizk-Allah (✉) · M. A. Farag · M. H. Barghout
Faculty of Engineering, Menoufia University, Shebin El-Kom, Egypt
e-mail: rizk_masoud@yahoo.com

A. E. Hassanien
Faculty of Computers and Information, Cairo University, Giza, Egypt
URL: <http://www.egyptscience.net>

Scientific Research Group in Egypt, Giza, Egypt

commonly known as the butterfly effect, means unpredictable, erratic motions, and a complex behavior governed by deterministic laws. The chaotic system that is extremely responsive to the initial conditions and parameters [2]. Chaotic systems with low-order deterministic models could enhance interpretation and description of phenomena (recorded data) of complicated real-world systems. The machinery condition, electrical circuits [3], and subjects of diagnosis of human health status [4] are some of the recent applications of chaotic systems. Many chaotic systems simulate real-world systems such as Chen, Lorenz, Newton–Leipnik, Volta, Rossler [5], and Lotka–Volterra [6].

In recent years, the control and synchronization of CSs have been intensively studied in multiple subjects [7]. Most of the recommended techniques only work according to the conditions of CSs that are defined in advance. Nevertheless, the parameters may be very difficult to identify in the actual world due to the CSs complexity. Parameter estimation of CSs has great value in nonlinear science. Control theory and signal processing have drawn great attention in different research subjects and can be investigated as a multidimensional optimization problem (multi-DOP) [2].

In the last decade, parameter identification of CSs has attracted a lot of attention in the literature [8]. Different types of traditional techniques have been improved to investigate these problems [9]. The most common approaches utilized to express the problem of parameters estimation of CSs as a multi-DOP are metaheuristic algorithms such as PSO [10], the genetic algorithm (GA) [11], and differential evolution algorithm (DE) [12]. They seem to be an efficient alternative to traditional methods because they do not require the gradient of the objective function, are not responsive to the starting point, and seldom get trapped in local optima [13].

The investigation of the parameter identification problems has a significant history, with a special focus on the Lorenz system. Lorenz introduced this system in 1963 while observing atmospheric convection [14], and it is considered the first chaotic attractor of a three-dimensional self-government system. The parameters are not easy to acquire due to the unstable dynamic behavior of CSs. So in recent years, metaheuristic algorithms have been popular and frequently used by the researchers for their efficient performance in a solution of optimization problems [15–17]; various metaheuristic algorithms employed for parameters identification of this system have been proposed by minimizing synchronization error such as GA [11], PSO [10], improved particle swarm optimization (IPSO) [18], a drifting particle swarm optimization (DPSO) [19], a quantum-behaved particle swarm optimization (QPSO) [20], PSO with dynamic inertia weight PSO (DIW-PSO) [21], and chaotic ant swarm (CAS) [5, 22, 23]. It is noted that PSO and its modifications are the most frequently used approaches for parameter estimation of CSs.

PSO was proposed in 1995 by Kennedy and Eberhart [24] as a population-based algorithm inspired by the social behavior of swarms in nature such as bird flock and fish school. The algorithm seeks out the best solution via sharing the historical and social knowledge between the candidates (particles) throughout the global solution space. PSO algorithm has been used as an important optimization technique in various applications due to its simple implementation, few parameters to adjust, and

fast convergence rate. On the other hand, PSO demonstrates some shortages. It is possible to get stuck in a local optimum in another sense, premature convergence, and the convergence rate reduces significantly in the later stages of evolution [25–27]. Therefore, many variants of the standard PSO approach have been proposed for different applications to make the most of the merits, improve the search capability and overcome the deficiencies such as immune PSO (IPSO) [28], continuous trait-based PSO (CTB-PSO) [29], and the hybrid GA-PSO [30, 31].

In this paper, parameters estimation of chaotic Lorenz system is modeled as a multi-DOP and solved based on a modified version of PSO by adding two terms to the velocity update formula. The modified algorithm is called memory-based particle swarm optimization (MbPSO). This modification of its role diversifies and enhances search capability. This is the first research of MbPSO for parameters estimation of the Lorenz system that the authors are aware of. The comparisons of the results acquired by other techniques show this algorithm's efficiency, effectiveness, and robustness.

The main contributions of this paper can be summarized as follows:

- The chaotic system is modeled, generally, and the Lorenz system is illustrated clearly.
- A modified version of PSO is proposed called MbPSO.
- The proposed algorithm (MbPSO) is applied for the Lorenz system to estimate its parameters.
- Comparisons are made between the proposed algorithm (MbPSO), modified PSO (PSO+), original PSO (programmed algorithms for this study), and other algorithms used in the literature.

The remainder of this paper is organized as follows: Section 2 introduces the problem formulation regarding the Lorenz system. Section 3 reviews the basic principles of PSO. Section 4 introduces the proposed MbPSO algorithm, and the parameter configuration for the proposed algorithm is illustrated in Sect. 5. The numerical simulation, and comparisons are presented in Sect. 6. Finally, this paper is concluded in Sect. 7.

2 Problem Formulation

Considering the n -dimensional chaotic system modeled by the ordinary nonlinear differential equation as below (ONLDE):

$$\dot{X} = f(X, X_o, \theta_o) \tag{1}$$

where $X = (x_1, x_2, \dots, x_n)^T \in R^n$ indicates the state vector, x_0 indicates the initial state and $\theta_0 = (\theta_{1o}, \theta_{2o}, \dots, \theta_{do})^T$ is a set of original parameters. During calculating the parameters, assume the chaotic system structure is known in advance, and hence, the approximated system can be defined as follows:

$$\dot{Y} = f(Y, X_o, \theta) \tag{2}$$

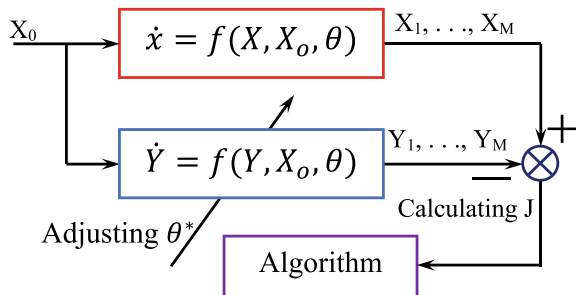
where $Y = (y_1, y_2, \dots, y_n)^T \in R^n$ indicates the state vector, and $\theta = (\theta_1, \theta_2, \dots, \theta_d)^T$ is a set of calculated parameters. Thus, the problem of the parameter estimation can be described as the following optimization problem:

$$\text{Min } J = \frac{1}{M} \sum_{k=1}^M X_k - Y_k^2 \text{ by searching suitable } \theta^* \tag{3}$$

where the length of data used for parameters estimation is defined by M . The state vectors of the original and the estimated systems at time k ($k = 1, 2, \dots, M$) are denoted by X_k and Y_k , respectively.

The parameters estimation for CSs is obviously expressed as a problem of multi-DOP, where the vector of decision is θ , and minimization of J is the objective function of the optimization problem. The parameters estimation problem of CSs based on optimization techniques can be demonstrated by Fig. 1. The parameters are not easy to acquire due to the unstable dynamic behavior of CS. Furthermore, multiple variables always exist in the problem, and multiple local optima solutions mislead the algorithm in the search zone of J , so conventional techniques are very easy to trap in local optima, and the global optimal parameters are difficult to attain. Thus, Lorenz system was selected as a CS model to test the performance of the proposed algorithm. Lorenz system [14] has an unpredictable complex dynamic behavior and exhibits infinite erratic periodic motions with extremely dependent initial conditions and parameters. The Lorenz model is employed as an equivalent model about the behavior of the atmosphere because it replicates some of the features of large-scale weather patterns, such as the simulation of system behavior, variations in the predictability of local events, and various time scales. The equations of the Lorenz system are system of three ONLDE resulting from a simple form of the fluid convection between two horizontal plates that is called Rayleigh Bénard problem. Lorenz system designated three time-dependent variables: convection motion intensity is represented by x , the temperature differential between increasing and decreasing currents is denoted by y , and z is the deviation from linearity in the vertical temperature profile [32]. The

Fig. 1 The overall scheme for optimization concept of parameters estimation in CSs



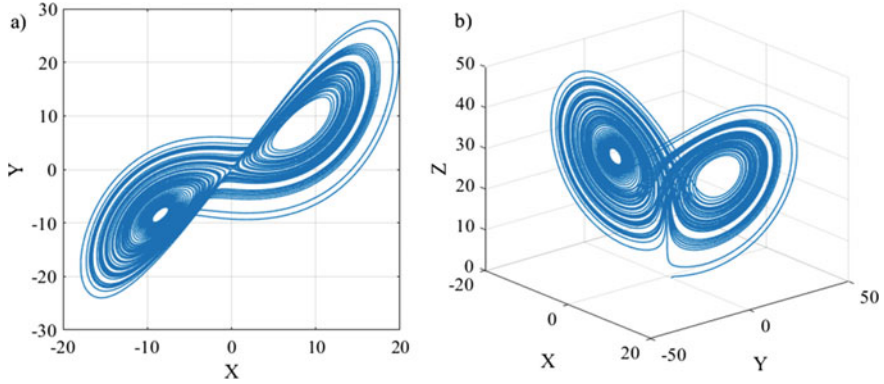


Fig. 2 The behavior of Lorenz system with $A = 10, B = 28, C = 8/3$ and initial conditions ($X_0 = 1, Y_0 = 0, Z_0 = 0$) for evolving time = 100 s. **a)** the projection of Lorenz attractor into x - y plane; **b)** the Lorenz attractor in three-dimensional space

following model is a mathematical representation of the Lorenz system:

$$\begin{cases} \dot{x} = A(y - x) \\ \dot{y} = Bx - xy - y \\ \dot{z} = xy - Cz \end{cases} \quad (4)$$

where $\dot{x} = dx/dt, \dot{y} = dy/dt, \dot{z} = dz/dt, A$ is defined as the Prandtl number, B ($B = R_a/R_c$) is defined as the Rayleigh number over the critical Rayleigh number, and C gives the size approximated by the system of the region [32]. All the parameters are positive numbers; for example, Ref. [33] demonstrates the behavior of a chaotic attractor with initial conditions ($X_0 = 1, Y_0 = 0, Z_0 = 0$) and $A = 10, B = 28,$ and $C = 8/3$. For a complete study of this system, see [33] (Fig. 2).

3 Particle Swarm Optimization (PSO)

PSO is considered as an evolutionary algorithm that is based on individual improvement in addition to collaboration and competition in the population. It depends on the simulation of simplified social models, such as the swarm theory: fish schooling, bird flocking [24]. PSO has a highly obvious, straightforward, and easy-to-implement theoretical structure.

PSO initializes randomly with a swarm of birds (particles) over the searching space. The particles search with a certain velocity and find the global optimum after several iterations. The main idea is to adjust the trajectory of each particle to its own best position and to the best particle of the swarm at each iteration. At each iteration, the velocity vector of each particle is affected by its inertia, its best position,

and the position of the best particle. Then, each particle moves to a new position. Assume that, the location and the velocity of particle i in the d -dimensional search space can be expressed as $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]^T$ and $V_i = [v_{i,1}, v_{i,2}, \dots, v_{i,d}]^T$, respectively. The particle fitness can be estimated according to the objective function of the optimization problem. The best position visited previously of particle i is denoted by the personal best position (pbest). It can be expressed as $P_i(t) = [p_{i,1}(t), p_{i,2}(t), \dots, p_{i,d}(t)]^T$. The position of the best particle of the swarm is defined by the global best position (gbest) $P_g(t) = [p_{g,1}(t), p_{g,2}(t), \dots, p_{g,d}(t)]^T$. At each step, the velocity and position of each particle is determined as follows:

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 r_1 (p_{i,j}(t) - x_{i,j}(t)) + c_2 r_2 (p_{g,j}(t) - x_{i,j}(t)), \quad j = 1, 2, \dots, d \quad (5)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1), \quad j = 1, 2, \dots, d \quad (6)$$

where ω , indicates to the inertia weight, controls the effect of previous particle velocity on its current one. C_1 and C_2 are the cognitive and collective acceleration coefficients, respectively, which are balance the effects of self-awareness and social awareness on particle movement toward the target and adjust the step size. r_1 and r_2 are two independent uniform distributed random constants within the range of $[0, 1]$.

In PSO, Eq. (5) implies that the particle's new velocity is updated based on its previous velocity and the distance of its present position from both the global best position of the entire swarm and its best historical position. A typical approach is to constrain the values of the elements of the vector v_i to a range between $[V_i, \min, V_i, \max]$ to control excessive wandering of particles outside of the search space $[X_i, \min, X_i, \max]$. The particle then moves toward a new position in accordance with Eq. (6). The procedure is repeated until a termination condition determined is met [34].

4 Memory-Based Particle Swarm Optimization (MbPSO)

This section presents a new modified version of PSO called memory-based particle swarm optimization (MbPSO). The suggested algorithm (MbPSO) is a proposed improvement for the particle velocity updating equation to determine the parameters of Lorenz system. Although the position updating equation in MbPSO is identical to the standard PSO, the equation of the velocity updating is improved by the addition of two new terms, as illustrated below:

$$v_{i,j}(t+1) = \Psi(\omega v_{i,j}(t) + c_1 r_1 (\text{pbest}_{i,j}(t) - x_{i,j}(t)) + c_2 r_2 (\text{gbest}_j(t) - x_{i,j}(t)) + c_3 r_3 (\text{pbest}_{i,j}(\text{rand}) - x_{i,j}(t)) + c_4 r_4 (v_{i,j}(\text{rand}))) \quad (7)$$

$$x_{i,j}(t + 1) = x_{i,j}(t) + v_{i,j}(t + 1),$$

$\forall i \in$ number of particles, $\forall j \in$ problem dimension, and $\forall t \in$ number of iterations

(8)

where ω is defined by the inertia weight. $r1, r2, r3,$ and $r4$ are random numbers within $[0, 1]$. $C1, C2, C3,$ and $C4$ denote the acceleration constants that control the effect of each term in the update process. $pbest$ denotes the particle’s best solution, while $gbest$ denotes the best solution reached by all particles. Knowing that the acceleration constants pull each particle to its best individual and global locations. As a result, small values cause short movements toward target regions, whereas high values cause abrupt movements. High values of the inertia weight, on the other hand, result in a broad search across the search space, whereas low values result in a more focused search [34].

In the original PSO, the first term, “inertial part” is represented as $\omega v_{i,j}(t)$ which is utilized as a search skill factor by using $\omega = \omega_{max} - t(\omega_{max} - \omega_{min})/t_{max}$, where ω_{min} and ω_{max} are the values of minimum and maximum inertia weight, respectively, and t_{max} refers to the maximum number of iterations [35]. The second term “cognitive part” is represented as $c_1 r_1 (pbest_{i,j}(t) - x_{i,j}(t))$ that is defined as the distance between the particle’s current position and the particle’s most well-known position, and it means that the particle’s decision will be influenced by its previous experiences. The third term “social part” is represented as $c_2 r_2 (gbest_j(t) - x_{i,j}(t))$ which is defined as the distance between the particle’s current position and its best neighborhood position, and it means that the particle’s decision is influenced by the rest of the particles [34]. From this idea, we added two new terms. The fourth term, “random self-cognition,” is represented as $c_3 r_3 (pbest_{i,j}(rand) - x_{i,j}(t))$ which relates the particle’s position to the best position of random particles, which in fact let each particle exploit the memory of other particles and allow the particles to randomly share their knowledge during the updating process; it results in a stronger interrelation between the whole swarm, higher chance in convergence toward good solutions, and higher diversity of the search process. The fifth term, which is represented as $c_4 r_4 v_{i,j}(rand)$, includes a random velocity that leads to increasing the potential of space exploration and preventing the MbPSO from being stuck in local optima. The new terms are inspired by [36].

The researchers utilize many additional terminologies with different meanings (e.g., [23, 37]). Still, this paper incorporates modifications to diversify the population search direction and enhance the swarm’s search capability. In original PSO, $gbest$ is utilized to enhance convergence characteristics, but the drawback of using $gbest$ is decreasing the diversity of the population that results in local minima. Thus, adding two new terms increases search capability, providing new knowledge and more information to the population. It can guide the particles to a better position, and the attraction of $gbest$ position to local optima in the search space is reduced. Figure 3 depicts a graphical view of the updating process for a particle’s position

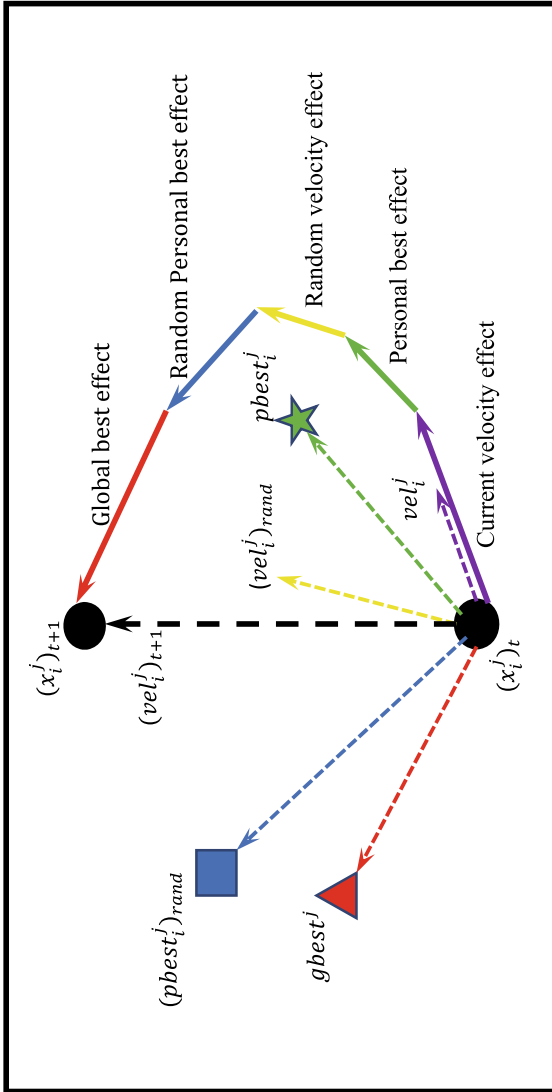


Fig. 3 Illustration of the particle's update equations in MbPSO

and velocity in the MbPSO algorithm. Also, the overall procedures of MbPSO are demonstrated in Fig. 4.

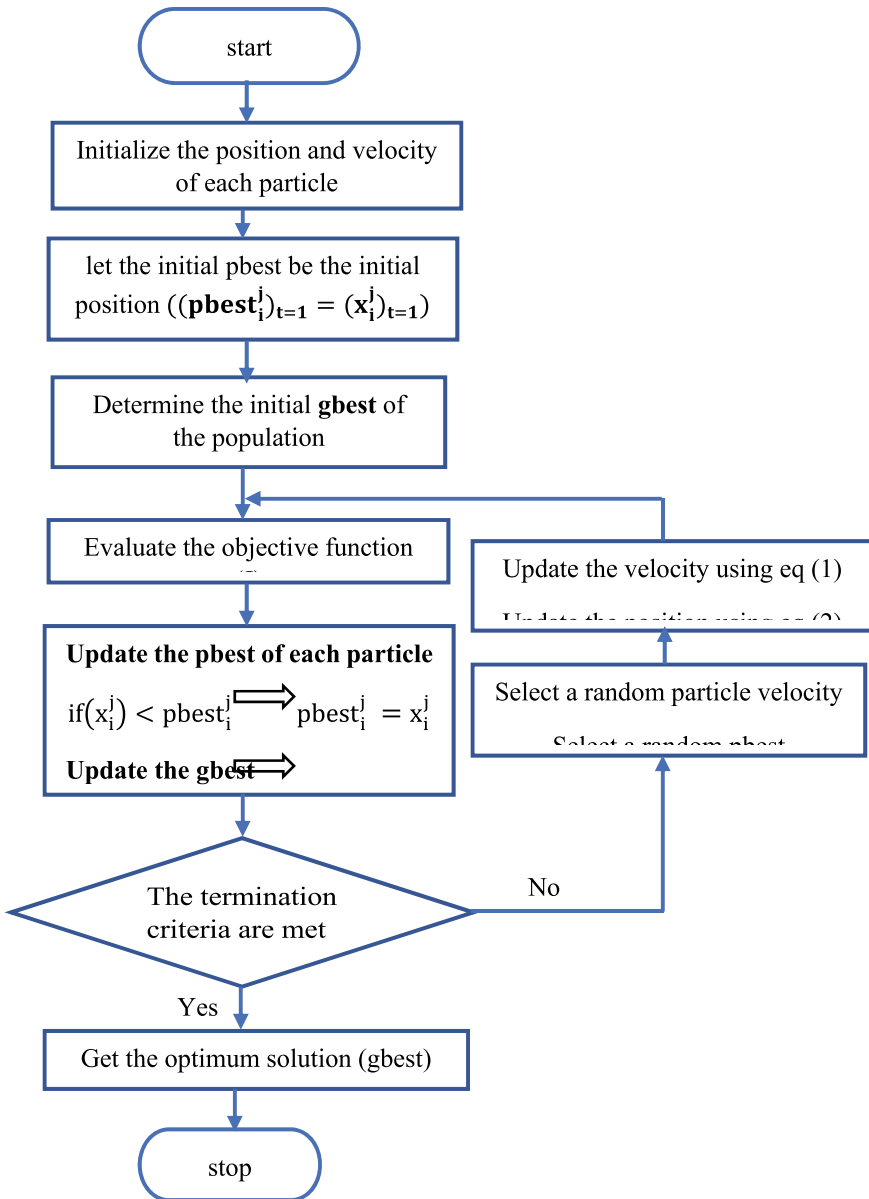


Fig. 4 Flowchart of the proposed algorithm (MbPSO)

5 Parameter Configuration for the Proposed Algorithm (MbPSO)

To make the best use of the proposed algorithm, the range of the parameter values are taken from the literature [36], and the trial and error method was used through the implementation on Lorenz system for configuring the parameters of the proposed algorithm during the operation of minimizing J . Thus, a population of 100 particles (p) was used; an inertia weight ω was established as linearly reducing from the peak value of 0.9 to a minimum value of 0.4. The acceleration constants were used as follows $C1 = 1.5$, $C2 = 2$, and $C3 = 3$ [38]. As different values of parameter $C4$ have a significant impact on the proposed algorithm, the various effects of $C4$ were studied for values under one; based on the literature, $C4$ was studied without including the constriction factor, and Fig. 5 shows the various effect of $C4$. For this study, the constant $C4 = 0.3$ introduces a better influence on the convergence than the others (the procedures for $C4$ are based on the coefficient mentioned above with other terms in [23]); in another aspect, the constriction factor has an extremely great effect on the proposed algorithm; following $C4$ is set to 0.3, the effect of constriction factor was analyzed from 0.2 to 0.8. Figure 6 shows the various effect of the constriction factor on the convergence and demonstrates that the constant $\psi = 0.4$ presents the best convergence. The maximum generation number was set to $t_{\max} = 100$ and $t_{\max} = 200$, and this is illustrated in the section of discussion and results. All simulations were implemented 20 times. Table 1 displays the parameters used in the compared algorithms (PSO, PSO+, and MbPSO).

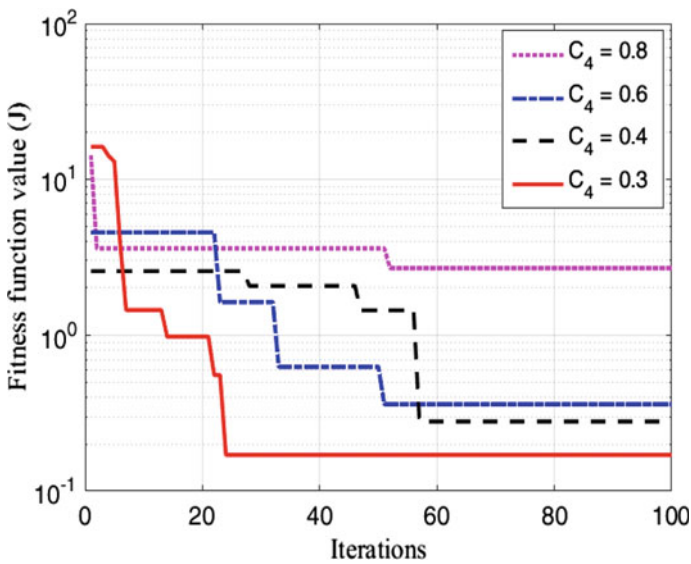


Fig. 5 Convergence graph. Specifying of the best value of constant C_4

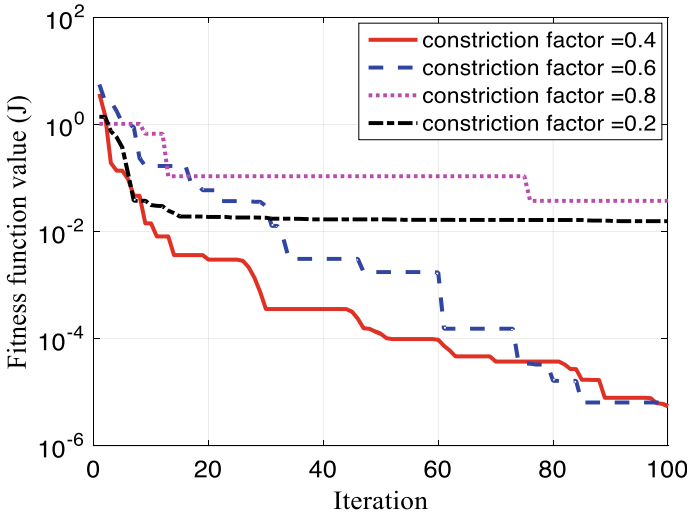


Fig. 6 Convergence graph. Specifying of the best value of constriction factor Ψ

Table 1 Parameter configuration for all algorithms used

Algorithm	Parameter configuration
PSO	$p = 100, \omega_{\max} = 0.9, \omega_{\min} = 0.4, c_1 = c_2 = 2$
PSO+	$p = 100, \omega_{\max} = 0.9, \omega_{\min} = 0.4, c_1 = c_2 = 1.5, c_3 = 0.4$
MbPSO	$p = 100, \Psi = 0.4, \omega_{\max} = 0.9, \omega_{\min} = 0.4, c_1 = 1.5, c_2 = 2, c_3 = 3, c_4 = 0.3$

6 Results and Discussion

The proposed algorithm MbPSO is implemented to estimate the parameters of the Lorenz chaotic system (A, B, and C). Figure 7 and Fig. 8 show the convergence characteristic of the fitness function J for MbPSO, PSO, and PSO + . These figures display that the value of J reduces to zero rapidly, which means MbPSO can converge to the global optimum very fast.

Furthermore, MbPSO is compared with a particle swarm optimization (PSO) [34] and a modified PSO (PSO+) [6] to test the performance of the proposed algorithm. The comparison was made for two cases: the first case, the maximum number of iterations was set to 100, the range of the estimated parameters is ($8 < A < 12, 20 < B < 30, 2 < C < 3$).

Table 2 compared the error and standard deviation (St.D.) values found by PSO, PSO+, and MbPSO. As shown in Table 2, MbPSO has fast convergence and accurate performance. Figure 7 illustrated that MbPSO has high performance and fast convergence, but PSO and PSO+ had almost the same performance. The second case, when the maximum generation number was set to 200 and the range of the estimated parameters increased ($0 < A < 20, 0 < B < 50, 0 < C < 5$), the performance of

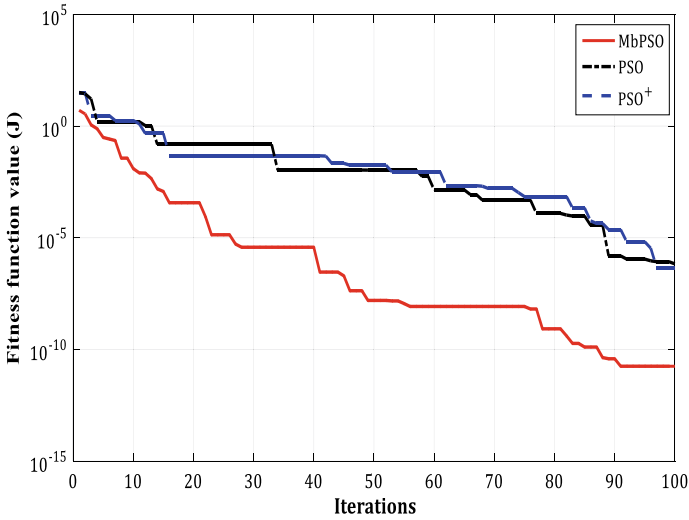


Fig. 7 Convergence characteristic of objective function (J) for three algorithms PSO, PSO+, and MbPSO (100 iterations)

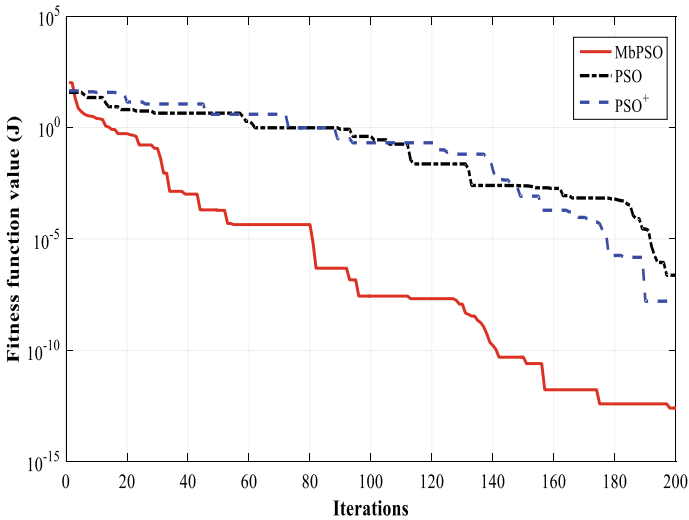


Fig. 8 Convergence characteristic of the objective function (J) for three algorithms PSO, PSO+, and MbPSO (200 iterations)

the modified algorithm (PSO+ and MbPSO) enhanced. Table 3 compared the error and St.D. values of PSO, PSO+ , and MbPSO. After the modifications of simulation conditions, MbPSO has the highest performance and the fastest convergence rate, and PSO+ has the second-best performance, as shown in Fig. 8.

Table 2 Comparison of convergence and statistical results of PSO, PSO+, and MbPSO (100 iterations)

Algorithm	MSE(J)	St.D		
	Best	Average	Worst	
PSO	1.2836E-06	1.2836E-06	3.7944E-04	1.5743E-04
PSO+	7.5670E-07	4.5711E-05	1.4774E-04	5.9247E-05
MbPSO	1.7666E-11	8.4102E-04	7.5000E-03	1.9000E-03

Table 3 Comparison of convergence and statistical results of PSO, PSO+, and MbPSO (200 iterations)

Algorithm	MSE(J)	St.D		
	Best	Average	Worst	
PSO	2.3475E-07	4.5743E-06	1.1863E-05	5.0764E-06
PSO+	1.6007E-08	1.5318E-05	5.3659E-05	2.3867E-04
MbPSO	2.5447E-13	1.4715E-04	5.5921E-04	2.5709E-05

Other comparisons can be made between the undefined parameter values A' , B' , and C' found by the MbPSO and the values found by with PSO and PSO+ from fitness function performance. The convergence rates of the parameters for the three-dimensional Lorenz system using MbPSO, PSO, and PSO+ are shown in Fig. 9. All estimated parameters found by MbPSO are very similar to the real values in all simulations, as shown in Fig. 9. This result demonstrates that the trajectories of the estimated parameters converge to their real values asymptotically. Therefore, MbPSO converges much faster than PSO and PSO+ . To get a full picture of our estimates, the statistical results found in estimating the parameters A' , B' , and C' using MbPSO, PSO, and PSO+ are summarized in Table 4. As shown in this table, the best results found by MbPSO are better than those obtained by PSO and the others. Additionally, the estimated parameter values found by MbPSO are still very similar to the real values of the original parameters.

On the other side, the abilities of MbPSO are compared to other metaheuristic techniques solved the same problem. Table 4 shows the comparison between the proposed algorithm and: (DE) [12], PSO [2], and a hybrid swarm intelligence algorithm (PSO-ACO) [23] in parameters estimation of Lorenz system. In general, MbPSO has accurate results better than the best results determined by these metaheuristic techniques (Table 5).

7 Conclusion

In this work, parameters identification for chaotic systems is modeled as a multi-dimensional optimization problem. The problem is solved using a modified PSO

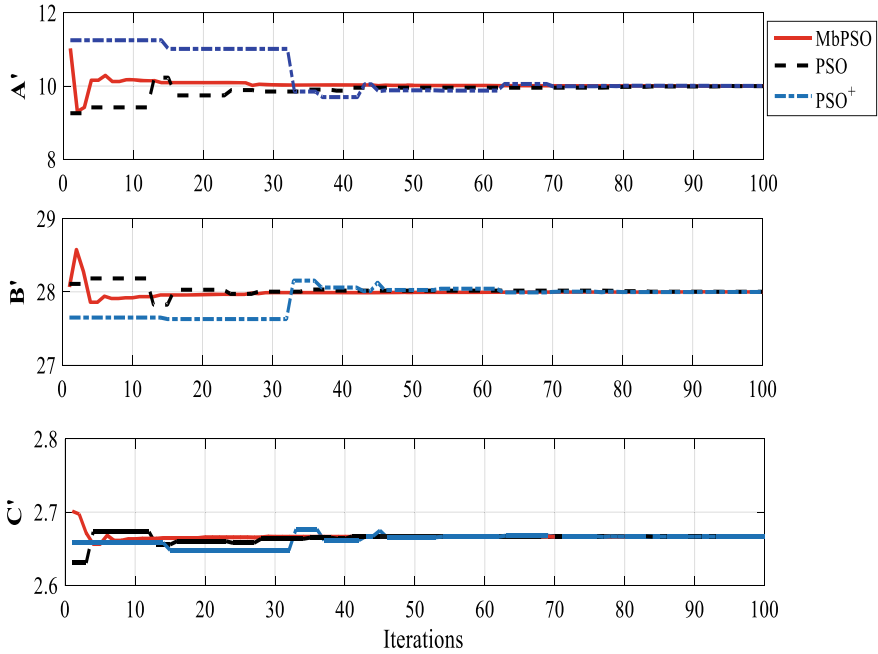


Fig. 9 Lorenz system searching process for A' , B' , and C' using PSO, PSO+, and MbPSO

Table 4 Statistical results found by PSO, PSO+, and MbPSO for the estimated parameters

Statistical results	Algorithms	Parameters		
		A'	B'	C'
Best	MbPSO	10.0000	28.0000	2.6667
	PSO+	10.0003	27.999	2.6667
	PSO	10.0009	27.9995	2.6666
Average	MbPSO	9.9981	28.0009	2.6667
	PSO+	10.0039	27.9985	2.6666
	PSO	9.9976	28.0011	2.6667
Worst	MbPSO	9.8567	28.0641	2.6689
	PSO+	10.0155	27.9938	2.6666
	PSO	9.9912	28.0001	2.6662

version called a memory-based particle swarm optimization (MbPSO). Lorenz system is selected to test the performance of MbPSO. In the proposed algorithm, MbPSO, two new terms are added to the standard PSO to diversify and enhance search capability. Comparisons are made between the proposed algorithm MbPSO, the original PSO, a modified PSO (PSO+), and other algorithms published in the literature. The comparisons and results demonstrate that the suggested algorithm

Table 5 Results obtained by many parameter estimation algorithms and the proposed MbPSO

Parameter	PSO [2]		DE [12]		PSO-ACO [23]		MbPSO (this work)		
	Average	Best	Average	Best	Average	Best	Average	Best	Worst
A'	10.0184	9.9953	10.0101	10.0001	10.0005	10.000	9.9981	10.00000	9.8567
B'	27.9934	28.0071	28.0000	28.0000	28.0011	28.0000	28.0009	28.00000	28.0641
C'	2.6663	2.6670	2.6667	2.6667	2.6673	2.6666	2.6667	2.66667	2.6689
MSE(J)	4.18E + 0	4.86E - 2	2.00E - 7	2.00E - 7	1.20E - 5	1.03E - 6	1.53E-6	1.76E-11	7.5E-3

is an effective and useful arithmetic method for parameter identification of chaotic systems, especially the Lorenz system, with high efficiency, fast convergence process, and accurate performance.

References

1. S.H. Strogatz, *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*, 1st pbk (2000)
2. Q. He, L. Wang, B. Liu, Parameter estimation for chaotic systems by particle swarm optimization. *Chaos Solitons Fractals* **34**, 654–661 (2007). <https://doi.org/10.1016/j.chaos.2006.03.079>
3. H.-G. Ma, X.-F. Zhu, J.-F. Xu, M.-S. Ai, Circuit state analysis using chaotic signal excitation. *J. Fr. Inst.* **345**, 75–86 (2008)
4. W. Klonowski, From conformors to human brains: an informal overview of nonlinear dynamics and its applications in biomedicine. *Nonlinear Biomed. Phys.* **1**, 5 (2007). <https://doi.org/10.1186/1753-4631-1-5>
5. Y. Mousavi, A. Alfi, Fractional calculus-based firefly algorithm applied to parameter estimation of chaotic systems. *Chaos Solitons Fractals* **114**, 202–215 (2018). <https://doi.org/10.1016/j.chaos.2018.07.004>
6. J.A. Lazzús, P. Vega-Jorquera, C.H. López-Caraballo, et al., Parameter estimation of a generalized Lotka–Volterra system using a modified PSO algorithm. *Appl. Soft Comput. J.* **96**, 106606 (2020). <https://doi.org/10.1016/j.asoc.2020.106606>
7. E. Elabbasy, H. Agiza, M. El-Dessoky, Global synchronization criterion and adaptive synchronization for new chaotic system. *Chaos. Chaos Solitons Fractals* **23**, 1299–1309 (2005)
8. Y. Liu, W.K.S. Tang, Modified dynamic minimization algorithm for parameter estimation of chaotic system from a time series. *Nonlinear Dyn* **66**, 213–229 (2011). <https://doi.org/10.1007/s11071-010-9922-0>
9. A. Alireza, *Ö. Acta Autom. Sin.* 37 (2011)
10. B. Samanta, C. Nataraj, Particle swarm optimization for chaotic system parameter estimation, in *2009 IEEE Swarm Intelligence Symposium SIS 2009*, pp. 74–80. <https://doi.org/10.1109/SIS.2009.4937847>
11. D. Dai, X. Ma, F. Li, Y. You, An approach of parameter estimation for a chaotic system based on genetic algorithm. *Acta Phys. Sin* **11**, 2459–2462 (2002)
12. B. Peng, B. Liu, F.-Y.Y. Zhang, L. Wang, Differential evolution algorithm-based parameter estimation for chaotic systems. *Chaos Solitons Fractals* **39**, 2110–2118 (2009). <https://doi.org/10.1016/j.chaos.2007.06.084>
13. A. ALFI, PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems. *Acta Autom. Sin.* 37:541–549 (2011). [https://doi.org/10.1016/s1874-1029\(11\)60205-x](https://doi.org/10.1016/s1874-1029(11)60205-x)
14. E.N. Lorenz, Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141 (1963)
15. R.M. Rizk-Allah, A.E. Hassanien, M. Elhoseny, M. Gunasekaran, A new binary salp swarm algorithm: development and application for optimization tasks. *Neural Comput. Appl.* **31**, 1641–1663 (2019)
16. R.M. Rizk-Allah, A.E. Hassanien, New binary bat algorithm for solving 0–1 knapsack problem. *Complex Intell. Syst.* **4**, 31–53 (2018)
17. A.E. Hassanien, R.M. Rizk-Allah, M. Elhoseny, A hybrid crow search algorithm based on rough searching scheme for solving engineering optimization problems. *J. Ambient Intell. Humaniz Comput.* First Online: 25 June 2018
18. H. Modares, A. Alfi, M.-M. Fateh, Parameter identification of chaotic dynamic systems through an improved particle swarm optimization. *Expert Syst. Appl.* **37**, 3714–3720 (2010)

19. J. Sun, J. Zhao, X. Wu et al., Parameter estimation for chaotic systems with a drift particle swarm optimization method. *Phys. Lett. A* **374**, 2816–2822 (2010)
20. K. Yang, K. Maginu, H. Nomura, Parameters identification of chaotic systems by quantum-behaved particle swarm optimization. *Int. J. Comput. Math.* **86**, 2225–2235 (2009)
21. A. Alfi, Particle swarm optimization algorithm with dynamic inertia weight for online parameter identification applied to Lorenz chaotic system. *Int. J. Innov. Comput. Inf. Control* **8**, 1191–1203 (2012)
22. L. Li, Y. Yang, H. Peng, X. Wang, An optimization method inspired by “chaotic” ant behavior. *Int. J. Bifurc Chaos* **16**, 2351–2364 (2006)
23. J.A. Lazzús, M. Rivera, C.H. López-Caraballo, Parameter estimation of Lorenz chaotic system using a hybrid swarm intelligence algorithm. *Phys. Lett. A* **380**, 1164–1171 (2016)
24. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN’95-International Conference on Neural Networks* (IEEE, 1995), pp. 1942–1948
25. R.M. Rizk-Allah, A. Slowik, A.E. Hassanien, Hybridization of grey wolf optimizer and crow search algorithm based on dynamic fuzzy learning strategy for large-scale optimization. *IEEE Access* **8**, 161593–161611 (2020). <https://doi.org/10.1109/ACCESS.2020.3021693>
26. R.M. Rizk-Allah, A.E. Hassanien, Locomotion-based hybrid salp swarm algorithm for parameter estimation of fuzzy representation-based photovoltaic modules. *J. Mod. Power Syst. Clean Energy* (2020)
27. R.M. Rizk-Allah, Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems. *J. Comput. Des. Eng.* **5**, 249–273 (2018)
28. C. Lin, Y. Liu, C. Lee, An efficient neural fuzzy network based on immune particle swarm optimization for prediction and control applications. *Int. J. Innov. Comput. Inf. Control* **4**, 1711–1722 (2008)
29. E. Keedwell, M. Morley, D. Croft, Continuous trait-based particle swarm optimisation (CTB-PSO), in *International Conference on Swarm Intelligence* (Springer, 2012), pp. 342–343
30. A. Gálvez, A. Iglesias, A new iterative mutually coupled hybrid GA–PSO approach for curve fitting in manufacturing. *Appl. Soft Comput.* **13**, 1491–1504 (2013)
31. I. Jain, V.K. Jain, R. Jain, Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. *Appl. Soft. Comput.* **62**, 203–215 (2018)
32. X.-J. Wu, S.-L. Shen, Chaos in the fractional-order Lorenz system. *Int. J. Comput. Math.* **86**, 1274–1282 (2009)
33. I. Pan, S. Das, Evolving chaos: Identifying new attractors of the generalised Lorenz family. *Appl. Math. Model* **57**, 391–405 (2018). <https://doi.org/10.1016/j.apm.2018.01.015>
34. J. Kennedy, R. Eberhart, Y. Shi, *Swarm Intelligence* (Morgan Kaufman, San Francisco, 2001)
35. R.C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 1 (2000), pp. 84–88
36. D. Sedighzadeh, E. Masehian, M. Sedighzadeh, H. Akbaripour, GEPSO: A new generalized particle swarm optimization algorithm. *Math. Comput. Simul.* **179**, 194–212 (2021)
37. J.A. Lazzús, Hybrid particle swarm-ant colony algorithm to describe the phase equilibrium of systems containing supercritical fluids with ionic liquids. *Commun. Comput. Phys.* **14**, 107–125 (2013)
38. A. Rezaee Jordehi, J. Jasni, Parameter selection in particle swarm optimisation: a survey. *J. Exp. Theor. Artif. Intell.* **25**, 527–542 (2013). <https://doi.org/10.1080/0952813X.2013.782348>