基于改进的布谷鸟搜索算法对分数阶生物系统的参数估计

卫佳敏[1], 于永光[1†], 张 硕[2]

(1. 北京交通大学 理学院, 北京 100044; 2. 西北工业大学 理学院, 陕西 西安 710072)

摘要: 近年来, 非线性分数阶系统的参数估计问题已经在许多科学和工程领域特别是计算生物学中, 引起了广泛的兴趣. 本文针对分数阶生物系统的参数估计问题, 将系统参数和分数阶导数同时作为独立的未知参数来进行估计, 并提出了一种改进的布谷鸟搜索(improved cuckoo search, ICS)算法来求解该问题. 在ICS算法中, 通过引入一个自适应参数控制机制, 同时结合反向学习方法, 从而达到提高算法收敛速度和估计值精度的目的. 最后, 以三种经典的分数阶生物动力系统模型为例进行了数值仿真, 其中还考虑了有测量误差和噪声数据的情形. 仿真结果表明ICS算法具有良好的适应性、较高的收敛可靠性及精度, 为求解非线性分数阶系统参数估计问题提供了一种有效工具.

关键词: 参数估计; 分数阶动力学模型; 布谷鸟搜索算法; 参数自适应控制; 能力诱导

引用格式: 卫佳敏, 于永光, 张硕. 基于改进的布谷鸟搜索算法对分数阶生物系统的参数估计. 控制理论与应用, 2019, 36(8): 1227 – 1238

DOI: 10.7641/CTA.2018.80158

Parameter estimation of fractional dynamical models arising from biological systems using an improved cuckoo search algorithm

WEI Jia-min, YU Yong-guang†, ZHANG Shuo

(1. Department of Mathematics, Beijing Jiaotong University, Beijing 100044, China;
2. School of Science, Northwestern Polytechnical University, Xi'an Shaanxi 710072, China)

Abstract: Recently, parameter estimation of nonlinear fractional-order systems has attracted great interest among many fields of science and engineering, especially computational biology. In this paper, we consider fractional dynamical models arising from biological systems, and parameter estimation of which is converted into a multi-dimensional optimization problem by treating both systematic parameters and fractional derivative orders as independent unknown parameters to be estimated. Moreover, an improved cuckoo search (ICS) algorithm is proposed as a novel technique to solve the problem of parameter estimation. In ICS, a simple adaptive parameter control mechanism is introduced, at the mean time, the opposition-based learning method is incorporated to the presented algorithm so that it can accelerate convergence speed and improve the accuracy of the estimated values. Numerical simulations are carried out on three typical fractional-order dynamical biological systems. We also investigate the condition with measurement error and noisy data. The simulation results demonstrate the effectiveness and efficiency of ICS, and show its significant superiority to the other methods. Thus, ICS may be deemed to be a promising tool for parameter estimation of nonlinear fractional-order systems.

Key words: parameter estimation; fractional dynamical models; cuckoo search algorithm; adaptive parameter control; competence induction

Citation: WEI Jiamin, YU Yongguang, ZHANG Shuo. Parameter estimation of fractional dynamical models arising from biological systems using an improved cuckoo search algorithm. *Control Theory & Applications*, 2019, 36(8): 1227 – 1238

1 Introduction

In the past decades, much attention has been drawn for the study of fractional calculus[1], which is a generalization of the traditional integer order integration and differentiation. In comparison with integer calculus, the major merit of fractional calculus lies in that it can offer more accurate modelling of dynamical systems possessing memory and hereditary properties. Recently, the applications of fractional calculus in modelling and control of systems in fields of science and engineering[2–3] has aroused spectacular interest, where dynamical systems with anomalous diffusion effects in constrained environments can be described by fractional differential equations, for example, computational biology, physics, chemistry and biochemistry, and hydrological applications[4–7].

Diffusive processes are crucial to biological interactions. But due to molecular crowding, the environ-

Received 3 July 2018; accepted 24 October 2018.
†Corresponding author. E-mail: ygyu@bjtu.edu.cn; Tel.: +86 10-51682692.
Recommended by Associate Editor: YANG Chun-hua.
Supported by the Fundamental Research Funds for the Central Universities (2017YJS200), the China Scholarship Council (201807090092) and the National Nature Science Foundation of China (61772063).

ments where these processes occur have high densities and viscosities. Biological media exhibits a large degree of complexity and heterogeneity, and often exhibits substantial compartmentalization[8]. Certain types of cellular differentiation are probabilistic and transient[9–11]. In this paper, we consider the behaviour of a genetic regulatory feedback model in B. *Subtilis* bacteria. In B. *Subtilis* bacteria, competence is a transiently differentiated state associated with the capability for DNA uptake from the nutrient limited environment[12]. In [12], Süel et al. considered interactions among the associated proteins Com K, Com S and the MecA complex, and accordingly put forward a dynamical model of competence induction that can be described by a highly nonlinear coupled ordinary differential equation. However, the translocations of Com K and Com S take place in the individual cells in fact. Hence, in order to recapitulate the biological phenomenon accurately, a nonlinear fractional dynamical system derived from the original dynamical model of competence induction in B. *Subtilis* bacteria is proposed by Liu et. al[13]. The fractional biological model characterized by long range memory transients describes dynamic processes that occur at different length and time scales.

In recent years, parameter estimation has become a subject of great significance in mathematics and applications, especially computational biology[14–17]. In computational biology, there are various kinds of parameter estimation problems which need to go backwards from the measurement data sets to estimated parameter values[13], such as the rates of some chemical reactions or to determine approximate values of the coefficients in the differential equations governing a particular phenomenon[18–20]. As for fractional dynamical systems, diffusion modelling through fractional order techniques has obtained accurate estimates of frequency-domain behavior from time-domain equations[21–22]. Gabano and Poinot developed a new identification algorithm to estimate the thermal conductivity and diffusivity via fractional order modelling[21]. What's more, in [13], novel techniques based on hybrid simplex search and particle swarm optimization was employed to parameter estimation for fractional dynamical models coming from biological systems.

Over the past two decades, nature-inspired algorithms, as a kind of global stochastic search methods, have attracted much attention among researchers for handling complex and tough optimization problems. In virtue of evolutionary processes existed in nature, numbers of nature-inspired intelligent algorithms have been accordingly proposed in succession, including genetic algorithm (GA)[23], differential evolution (DE)[24], particle swarm optimization (PSO)[25], cuckoo search (CS) algorithm[26], artificial bee colony (ABC)[27], state transition algorithm (STA)[28–31] and so on. Due to the outstanding characteristics, different nature-inspired algorithms have achieved a lot of promising results on a variety of optimization problems so far[32–34]. Parameter estimation essentially can be regarded as a complex optimization problem, which is enabled to be solved by the nature-inspired algorithms. In this case, the unknown parameters are considered as a series of independent variables, and parameter estimation is converted to a multi-dimensional optimization problem via system inversion mechanism[35]. However, as far as we know, little research has been done using cuckoo search algorithm to estimate unknown parameters of fractional-order biological systems, including systematic parameters and fractional derivative orders.

Cuckoo search (CS) is a population-based heuristic evolutionary technique proposed by Yang and Deb[26], the basic idea of which comes out from the parasitic brood swarm intelligence technique in cuckoo species together with the Lévy flight behavior of some birds and fruit flies. CS evolves a population of candidate solutions by conducting Lévy flights random walk (L-FRW) followed by using biased/selective random walk (BSRW), to keep a balance between global exploration and local exploitation in the search space. Moreover, at each iteration process, a greedy (selection) strategy is used to select a better solution from the current and new generated solutions according to the corresponding fitness values after each random walk. The salient features of CS lie in its simple concept, limited parameters and easy combination with other intelligent algorithms. CS has received much attention in recent years and has proven to be an efficient method theoretically and practically. On the one hand, CS satisfies the global convergence requirements, and thus is able to guarantee global convergence properties[36]. On the other hand, CS has been applied in many fields of optimization and computational intelligence with promising results. For instance, CS exhibits better performance than other algorithms when applied to a range of continuous optimization problems in the engineering design applications[37–38]. Hence, it's necessary to utilize CS to tackle the problem of parameter estimation of fractional dynamical biological systems.

Various CS extensions have been put forward so far. However, there is no specific algorithm to achieve the best solution for all optimization problems[39]. Meanwhile, it is hard to achieve an appropriate trade-off of CS between exploration and exploitation. At each iteration process, exploration walk around more extensive search space results in increased required computation time, whilst highly favored exploitation usually means a rapid loss of diversity and leads to the so-called premature convergence or stagnation problem. Therefore, it is still necessary to put forward new techniques to improve the comprehensive performance of CS. In the basic C-

S algorithm, BSRW searches for new solutions using a uniformly distributed random mutation factor which plays an important role in controlling the population diversity and the explorative power of the algorithm. Generally speaking, a large mutation factor contributes to population diversification, while a small mutation factor helps to accelerate the convergence rate. However, the mutation factor in BSRW lacks the ability to balance well these two aspects. What's more, given that a constant fraction ($P_{\rm a}$) of worse nests are abandoned in BSRW, only using the adaptive parameter control might help little at the later stages of evolution.

Motivated by these observations, an improved cuckoo search (ICS) algorithm is proposed in this paper to accelerate convergence speed and improve the accuracy of the estimated values for the parameter estimation problems. To sum up, the contributions of this research are presented as follows:

1) A simple adaptive parameter control mechanism is introduced to BSRW as similar done in MDE_pBX algorithm[40]. The adaptive parameter control mechanism dynamically updates the control parameters based on a Cauchy distribution and the Lehmer mean at each iteration process. It is worth noting that there is no need to obtain a user's prior knowledge of the relationship between the parameter settings and the features of a specific optimization problem. The former is beneficial for diversifying the mutation factors so that premature convergence can be avoided, and the latter one is capable of propagating larger mutation factors, which in turn helps to enhance the progress rate of CS.

2) The opposition-based learning (OBL) method is incorporated to the proposed algorithm by considering an estimate and its corresponding opposite estimate simultaneously. OBL provides a higher chance of finding solutions which are closer to the global optima[41], and thus can help to obtain higher precision.

3) ICS is applied to parameter estimation of fractional dynamical models arising from biological systems. Numerical simulations are performed on the nonlinear fractional dynamical model of competence induction in B. *Subtilis* bacteria. In addition, to further increase the credibility for the optimization performance of the proposed ICS algorithm, another two fractional-order dynamical biological systems are also chosen as examples for numerical simulations, namely the fractional-order cellular neural network (CNN) and the fractional-order Lotka-Volterra system.

Statistical comparisons with the basic CS, three improved CS variants and three other state-of-the-art algorithms are provided as well. The simulation results demonstrate the effectiveness and efficiency of ICS, and show significant its superiority to the other methods. Thus, ICS may be deemed to be a promising tool for parameter estimation of nonlinear fractional-order systems.

The rest of the paper is organized as follows. In Section 2, preliminaries including the basic concepts of fractional derivative and problem formulation are presented. In Section 3, the ICS algorithm is proposed in sufficient details after a brief introduction of the basic CS. In Section 4, parameter estimation of three typical fractional-order dynamical biological systems is conducted through numerical simulations. Finally, conclusions are drawn in Section 5.

2 Preliminaries

2.1 Definitions of fractional derivative

There are several definitions of fractional derivatives, such as Grunwald-Letnikov (G–L), Riemann-Liouville (R–L) and Captuo definitions. In particular, the Caputo definition for fractional derivative is used in this paper since the initial conditions of Caputo derivatives differential equations own the identical form with integer-order ones.

Definition 1[42] Let $z(t) \in C_{\mu}$, $\mu \geqslant 1$, the Riemann-Liouville integral operator of $\beta > 0$ is defined as

$$J^{\beta}z(t) = \frac{1}{\Gamma(\beta)}\int_{t_0}^{t}(t-\tau)^{\beta-1}z(\tau)\mathrm{d}\tau, \tag{1}$$

where $\Gamma(\cdot)$ is the gamma function.

Definition 2[43] The Captuo fractional derivative of function $f(t)$, and $f(t) \in C_{-1}^{m}$, $m \in \mathbb{N} \cup \{0\}$ with $\alpha > 0$ is defined as

$$ {}_{t_0}D_{\rm t}^{\alpha}f(t) = J^{m-\alpha}f^{(m)}(t) = \begin{cases} \dfrac{1}{\Gamma(m-\alpha)}\displaystyle\int_{t_0}^{t}(t-\tau)^{m-\alpha-1}f^{(m)}(\tau)\mathrm{d}\tau, \\ \qquad\qquad \alpha \in [m-1, m], \\ \dfrac{\mathrm{d}^m}{\mathrm{d}t^m}f(t),\ \alpha = m, \end{cases} \tag{2}$$

where $y^{(m)}$ is the ordinary mth derivative of y, J is the Riemann-Liouville integral operator.

2.2 Problem formulation

Consider the following n-dimensional fractional-order dynamic system:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = {}_{\rm a}D_t^{1-\gamma}f(X(t), X_0, \theta), \tag{3}$$

where $X(t) = (x_1(t), x_2(t), \cdots, x_n(t))^{\rm T} \in \mathbb{R}^n$ denotes the state vector, $X_0 = (x_{10}, x_{20}, \cdots, x_{n0})^{\rm T}$ denotes the initial state, $\theta = (\theta_1, \theta_2, \cdots, \theta_n)^{\rm T}$ is a set of original systematic parameters, and $\gamma = (\gamma_1, \gamma_2, \cdots, \gamma_n)^{\rm T}$ is a set of fractional derivative orders.

It is noted that the initial-value problem (3) is equivalent to the following Volterra integral equation:

$$X(t) = X_0 + \frac{1}{\Gamma(\gamma)}\int_0^t (t-\tau)^{\gamma-1}f(\tau, X(\tau))\mathrm{d}\tau, \tag{4}$$

which can be solved by using the well-known predictor-

corrector method with Caputo derivatives proposed by Diethelm et al. in Ref. [44].

Suppose the structure of system (3) is known, and then the estimated system can be written as

$$\frac{\mathrm{d}\hat{X}(t)}{\mathrm{d}t} =_{\mathrm{a}} D_t^{1-\hat{\gamma}} f(\hat{X}(t), X_0, \hat{\theta}), \tag{5}$$

where $\hat{X}(t) = (\hat{x}_1(t), \hat{x}_2(t), \cdots, \hat{x}_n(t))^{\mathrm{T}} \in \mathbb{R}^n$ denotes the state vector of the estimated system, $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_n)^{\mathrm{T}}$ is a set of estimated systematic parameters, and $\hat{\gamma} = (\hat{\gamma}_1, \hat{\gamma}_2, \cdots, \hat{\gamma}_n)^{\mathrm{T}}$ is a set of estimated fractional derivative orders.

To estimate the parameters of system (3), the following objective function is defined as

$$F = \sum_{k=1}^{N} \|X_k - \hat{X}_k\|^2, \tag{6}$$

where $k = 1, 2, \cdots, N$ is the sampling time point, and N denotes the length of data used for parameter estimation. X_k and $\hat{X}_k$ denote the state vector of the original system (3) and the estimated system (5) at time k, respectively. $\|\cdot\|$ denotes Euclid norm.

The parameter estimation of system (3) can be achieved by searching suitable γ^* and θ^* such that the objective function (6) is minimized, i.e,

$$(\gamma^*, \theta^*) = \arg \min_{(\gamma,\theta)\in\Omega} F, \tag{7}$$

where Ω is the searching space admitted for systematic parameters and fractional derivative orders. In other words, parameter estimation of fractional-order dynamic systems is converted into a multi-dimensional optimization problem, thus, the main task is to find the best combination of the independent variables γ^* and θ^* via the objective function.

3 Improved cuckoo search (ICS) algorithm

3.1 Cuckoo search algorithm

CS algorithm is a simple yet very promising stochastic population-based method. For simplicity in describing the basic CS algorithm, three idealized rules[26] are used: 1) Each cuckoo bird lays one egg at a time and dumps it at a random chosen host nest; 2) The best nests with high-quality eggs will be carried over to the next generations; 3) The number of available host nests is fixed, and the host bird may discover the alien egg laid by a cuckoo with a probability $P_{\mathrm{a}} \in [0, 1]$.

In CS, each egg in a nest represents a solution, and each cuckoo is assumed to lay only one egg (thus representing one solution). The aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests[45]. At each iteration process, CS employs a balanced combination of a local random walk and a global explorative random walk, which are controlled by a switching parameter P_{a}. After each random walk, a greedy strategy is used to select better solutions from the current and new generated solutions according to their fitness values.

At generation t, based on the rules and description above, the global random walk is carried out by using Lévy flights expressed as

$$X_i^t = X_i^t + \alpha \oplus \mathrm{L\acute{e}vy}(\lambda), \tag{8}$$

where $\alpha > 0$ is the step size related to the scales of the problem of interest, and the product $\oplus$ denotes the entry-wise multiplication. Essentially, Lévy flights provide a random walk, the random steps of which are drawn from a Lévy distribution for large steps, and can be calculated as follows[36]:

$$\mathrm{L\acute{e}vy}(\lambda) \sim \frac{\phi \times \mu}{|\mathrm{v}|^{\frac{1}{\lambda}}}, \tag{9}$$

$$\phi = \left(\frac{\Gamma(1+\lambda) \times \sin(\frac{\pi \times \lambda}{2})}{\Gamma(\frac{1+\lambda}{2}) \times \beta \times 2^{\frac{\lambda-1}{2}}}\right)^{\frac{1}{\lambda}}, \tag{10}$$

where λ is a constant number suggested as 1.5 in [37], μ and v are random numbers drawn from a normal distribution and $\Gamma(\cdot)$ denotes the gamma function. Hence, Eq. (8) in LFRW can be reformulated as

$$X_i^t = X_i^t + \alpha_0 \cdot \frac{\phi \times \mu}{|v|^{\frac{1}{\lambda}}} \cdot (X_i^t - X_{\mathrm{best}}), \tag{11}$$

where α_0 is a scaling factor (generally, $\alpha_0 = 0.01$), X_{best} is the best solution obtained so far.

Algorithm 1 Pseudo code of the basic CS algorithm.

Generate an initial population of NP host nests X_i, $(i = 1, 2, \cdots, \mathrm{NP})$;
Evaluate the fitness value of each nest X_i;
Determine the best nest with the best fitness value;
while termination_condition_does_not_meet **do**
  for $i = 1, 2, \cdots, \mathrm{NP}$, **do**
    Generate a cuckoo X_i randomly using L-FRW according to Eq. (11);
    Evaluate the fitness value $F_i = f(X_i)$;
    Choose a random nest X_j;
    if $(F_j < F_i)$ **then**
      Replace nest X_i with X_j;
    end if
  end for
  for $i = 1, 2, \cdots, \mathrm{NP}$, **do**
    Search for a new solution using BSRW according to Eq. (13);
  end for
  Keep the best nest with quality solution;
  Rank the nests and find the current best one;
  Pass the current best nest to the next generation;
end while

After LFRW, CS continues to search for new solutions by performing the local random walk, which can be represented as biased/selective random walk (BSR-

W). A substantial fraction of the new solutions are generated by far field randomization and their positions are supposed to be far enough from the current best solution[26]. Accordingly, a mutant vector V_i^t is generated as follows:

$$V_i^t = X_i^t + r \cdot (X_{\mathrm{r1}}^t - X_{\mathrm{r2}}^t), \tag{12}$$

where r_1 and r_2 are two random indexes, r denotes the mutation factor which is a uniformly distributed random number in the interval $[0, 1]$. In case of being trapped into a local optimum, BSRW employs a crossover operator with the probability of cuckoos' being discovered:

$$U_i^t = \begin{cases} V_i^t, & \text{if } (\mathrm{rand}[0,1] > P_{\mathrm{a}}), \\ X_i^t, & \text{otherwise.} \end{cases} \tag{13}$$

By using the greedy strategy, the next generation solution X_i^{t+1} is selected from X_i^t and U_i^t through evaluation of their corresponding fitness values.

The procedure of the basic CS algorithm is described as the pseudo code shown in Algorithm 1.

3.2 Adaptive parameter control

In the basic CS algorithm, BSRW searches for new solutions through three simple evolutionary operations which are similar to the procedures of DE, namely mutation, crossover and selection. For mutation, BSRW adopts a uniformly distributed random number in the interval $[0, 1]$ as its mutation factor. According to earlier theoretical studies on DE[46–47], it can be drawn that the mutation factor plays an important role in controlling the population diversity and the explorative power of the algorithm. Generally speaking, a large mutation factor contributes to population diversification, while a small mutation factor helps to accelerate the convergence rate. However, the mutation factor in BSRW lacks the ability to balance well these two aspects. Inspired by the parameter adaptation schemes used in MDE_pBX algorithm[40], we combine the adaptive method with CS in order to improve the original mutation factor.

At each generation t, the mutation factor F_i of each individual target vector X_i^t is generated independently as

$$F_i = \mathrm{Cauchy}(F_{\mathrm{m}}, 0.1), \tag{14}$$

where $\mathrm{Cauchy}(F_{\mathrm{m}}, 0.1)$ is a random number sampled from a Cauchy distribution with the location parameter F_{m} and mutation factor 0.1.

The value of F_i is to be truncated when $F_i \geqslant 1$, and regenerated when $F_i \leqslant 0$. All the successful mutation factors will be stored in the set S_{F}, that is to say, the current generation generating better trial vectors are able to be carried over to the next generation. The location parameter F_{m} of the Cauchy distribution is initialized to be 0.5, and then updated at the end of each generation according to the following formula:

$$F_{\mathrm{m}}^{t+1} = w_{\mathrm{F}} \cdot F_{\mathrm{m}}^t + (1 - w_{\mathrm{F}}) \cdot \mathrm{mean}_{\mathrm{L}}(S_{\mathrm{F}}), \tag{15}$$

where w_{F} is a random weight factor between 0.8 and 1 given by

$$w_{\mathrm{F}} = 0.8 + 0.2 * \mathrm{rand}(0, 1), \tag{16}$$

and $\mathrm{mean}_{\mathrm{L}}(\cdot)$ is the Lehmer mean[48] formulated by

$$\mathrm{mean}_{\mathrm{L}}(S_{\mathrm{F}}) = \frac{\sum\limits_{F_i \in S_{\mathrm{F}}} F_i^2}{\sum\limits_{F_i \in S_{\mathrm{F}}} F_i}. \tag{17}$$

In this case, Eq. (12) can be rewritten as follows:

$$V_i^t = X_i^t + F_i \cdot (X_{\mathrm{r1}}^t - X_{\mathrm{r2}}^t), \tag{18}$$

where F_i presents an adaptive mutation factor drawn from Cauchy distribution. Therefore, new mutant solutions are generated through the modified Eq. (18), and further evolve by applying a crossover operator under the discovering probability and a selection operator as well in BSRW.

It is noteworthy that F_i is updated based on a truncated Cauchy distribution. Compared with a normal distribution, the Cauchy distribution has a far wider tail which is beneficial for diversifying the mutation factor, and thus brings sufficient perturbation so as to avoid premature convergence. Moreover, the set S_{F} is used to memorize the successful mutation factors in the current generation. By applying the Lehmer mean of S_{F} in Eq. (17), the adaptation of F_{m} can place more weight on larger successful scaling factors which leads to larger perturbation to the target vectors, and thus helps to circumvent premature convergence at local optima.

3.3 Opposition-based learning

Opposition-based learning (OBL), introduced by Ti-zhoosh[49], is a new concept in computational intelligence. The main idea behind OBL is to consider both of a solution and its corresponding opposite solution in order to get a better approximation of the current candidate solutions. It has been proven to be an effective method to enhance various optimization approaches[41,50]. Hence, the OBL idea is incorporated into our proposed algorithm, to further increase diversity and speed up the convergence.

Suppose $X = (x_1, x_2, \cdots, x_n)$ is a solution in an n–dimensional space, where $x_i \in [Lx_i, Ux_i]$, $(i = 1, 2, \ldots, n)$. Then, the opposite solution $X' = (x_1', x_2', \cdots, x_n')$ is given by

$$x_i' = Lx_i + Ux_i - x_i. \tag{19}$$

Let $f(\cdot)$ be a fitness function via which the fitness value can be evaluated. According to the above given definitions of X and X', if $f(X') \leqslant f(X)$, then X is replaced with X', otherwise X is kept. Thereby, the solution and its opposite solution are evaluated simultaneously in order to obtain the fitter one. OBL is implemented to initialize population and produce new solutions during evolution process similar to [41].

3.4 The proposed ICS algorithm

In this section, the proposed ICS algorithm is presented on the basis of the aforementioned methods in order to improve the performance of the basic CS. Firstly, population initialization is carried out by OBL which favors diversification so that a superior set of starting candidate solutions are generated. Then, ICS respectively uses LFRW and BSRW to search for new solutions at the following each iteration process, where BSRW evolves under the adaptive parameter control mechanism that dynamically updates the control parameters based on a Cauchy distribution and the Lehmer mean at each iteration process. Afterwards, new populations are calculated in terms of opposition-based generation jumping to accelerate the convergence speed at the later stages of evolution.

Thereinto, at the stage of population initialization, a double-size population is generated through uniform random distribution and the corresponding quasi-opposite positions, which are defined as

$$x_{i,j} = a_j + (b_j - a_j) \cdot \epsilon, \tag{20}$$

$$x^{\mathrm{O}}_{i,j} = a_j + b_j - x_{i,j}, \tag{21}$$

where a_j and b_j are the lower and upper bounds of jth variable, respectively; ϵ is a uniformly random number from $[0, 1]$. The NP best solutions are selected as the final initial population from X and X^{O}.

Generation jumping calculates the opposite of each variable based upon the minimum and maximum values of that in the current population:

$$x^{\mathrm{O}}_{i,j} = x^{\mathrm{P}}_{\min}(j) + x^{\mathrm{P}}_{\max}(j) - x_{i,j}, \tag{22}$$

where $x^{\mathrm{P}}_{\min}(j)$ and $x^{\mathrm{P}}_{\max}(j)$ denote the minimum and maximum values of the jth variable in the current population P, respectively.

In addition, since that the scaling factor in LFRW is sensitive to the length-scale of candidate optimization problems, we use a varied scaling factor as similar done in [51] instead of a fixed constant value α_0 in the basic CS algorithm. Hence, Eq. (11) is reformulated as

$$X^t_i = X^t_i + \delta \cdot \frac{\phi \times \mu}{|v|^{\frac{1}{\lambda}}} \cdot (X^t_i - X_{\mathrm{best}}), \tag{23}$$

where δ is a varied scaling factor drawn from a uniform distribution within the range $[0, 1]$. Besides, a simple boundary-handling method[52] is utilized in the evolution process of LFRW and BSRW. If the jth element of U^t_i is out of the boundary $[x_{\min,j}, x_{\max,j}]$, then it is reset as below:

$$u^t_{i,j} = \begin{cases} \min\{x_{\max,j}, 2x_{\min,j} - u^t_{i,j}\}, & \text{if } u^t_{i,j} < x_{\min,j}, \\ \max\{x_{\min,j}, 2x_{\max,j} - u^t_{i,j}\}, & \text{if } u^t_{i,j} > x_{\max,j}. \end{cases} \tag{24}$$

In this paper, to make a fair comparison, the same yet simple boundary-handling method is applied for all mentioned algorithms. The pseudo-code of the proposed ICS algorithm is shown in Algorithm 2.

Algorithm 2 Pseudo code of the proposed ICS algorithm.

Generate an initial population of NP host nests X using OBL by Eqs. (20)–(21);

Determine the best nest with the best fitness value;

while termination_condition_does_not_meet **do**

  for $i = 1, 2, \cdots, \mathrm{NP}$, **do**

    Search for a new solution X_i randomly using LFRW by Eq. (23);

    Perform the boundary-handling method using Eq. (24);

    Greedily select a better solution using the greedy strategy according to their fitness values;

  end for

  for $i = 1, 2, \cdots, \mathrm{NP}$, **do**

    Generate the mutation factor F_i using Eq.(14);

    Search for a mutant solution V_i using Eq. (18);

    Perform the boundary-handling method using Eq. (24);

    Obtain a new solution U_i via crossover operation using Eq. (13);

    Greedily select a better solution from U_i and X_i according to their fitness values;

  end for

  Generate new solutions using opposition-based generation jumping by Eq. (22);

  Keep the best nest with quality solution;

  Rank the nests and find the current best one;

  Pass the current best nest to the next generation;

  Update F_{m} using Eqs. (15)–(17);

end while

3.5 Computational complexity of ICS

Compared with the basic CS algorithm, ICS needs to perform additional computations on the OBL process. The newly proposed mutation factor under adaptive parameter control in ICS has the same order of complexity level as that of the uniformly distributed random number in CS. During one generation, calculations of opposition population and population sorting are carried out after BSRW. Suppose n and N denote dimension and population size, the computation complexity of this procedure takes $\mathrm{O}(N \cdot n)$. Since the complexity of the original CS algorithm is $\mathrm{O}(G\max \cdot N \cdot n)$ where $G\max$ denotes the maximal number of generation, the total computational complexity of ICS is $\mathrm{O}(G\max \cdot N \cdot n + N \cdot n)$, which is simplified to $\mathrm{O}(G\max \cdot N \cdot n)$. Hence, the proposed ICS does not significantly increase the overall complexity compared with the original CS.

4 Parameter estimation of fractional dynamical models arising from biological systems using ICS

In this section, we investigate the parameter estimation problem of fractional dynamical models arising from biological systems. The proposed ICS algorithm is applied to estimate the unknown parameters for three typical fractional-order dynamical biological systems, i.e., the nonlinear fractional dynamical model of competence induction in B. *Subtilis* bacteria, the fractional-order cellular neural network (CNN) and the fractional-order Lotka-Volterra system. Also, comparisons with the basic CS, three improved CS variants and three other state-of-the-art algorithms are conducted through various numerical simulations.

4.1 Nonlinear fractional-order dynamical biological systems

To start with, a brief introduction of the competence induction in B. *Subtilis* bacteria[12] is given before carrying out parameter estimation. When encountering the case of nutrient limitation, a minority of B. *Subtilis* cells become competent for DNA uptake while most of those commit irreversibly to sporulation. Extensive researches have demonstrated that the process of competence induction is dependent on and driven by the concentration levels of two key proteins, namely Com K and Com S. On the one hand, Com K activates its own expression together with expression of a series of genes necessary for competence required for competence induction. On the other hand, the Com S peptide competitively inhibits Com K degradation via the MecA complex as mentioned in [53–54]. In particular, Süel et al.[12] considered interactions among Com K, Com S and the MecA complex, which is collectively referred to as the 'MeKS' module, and accordingly proposed its mathematical model. The MeKS model can be reduced to a dynamical system of two ordinary differential equations incorporating both the direct positive and the Com S mediated negative feedback loops of Com K. According to [13], Liu et al. introduced a nonlinear fractional dynamical model of competence induction in B. *Subtilis* bacteria which can be written as a system of two nonlinear fractional ordinary differential equations in dimensionless form. The system is selected as the first example in this paper, expressed as

$$\begin{cases} \dfrac{\mathrm{d}K}{\mathrm{d}t} = {}_0D_{\mathrm{t}}^{1-\gamma_1}\left(a_k + \dfrac{b_k K^n}{K_0^n + K^n} - \dfrac{K}{1+K+S}\right), \\ \dfrac{\mathrm{d}S}{\mathrm{d}t} = {}_0D_{\mathrm{t}}^{1-\gamma_2}\left(\dfrac{b_{\mathrm{s}}}{1+(\dfrac{K}{k_1})^p} - \dfrac{S}{1+K+S}\right), \end{cases} \tag{25}$$

where K and S denote the concentration levels of Com K and Com S protein, respectively. a_k and b_k represent the minimal and fully activated rate of Com K production, respectively. k_0 indicates the concentration of Com K required for 50% activation. Moreover, the cooperativity of Com K auto-activation and Com S repression are parameterized by the Hill coefficients n and p, respectively. Expression of Com S has the maximum rate b_s and is half-maximal at $K = k_1$.

The dynamic behavior of fractional-order differential equations in system (25) varies upon different set of fractional orders and systematic parameters. Fig. 1 shows numerical simulations of Com K and Com S activities as a function of time under $\gamma_1, \gamma_2 = 0.99, 0.90, 0.85$ when systematic parameters are set as $(a_k, b_k, b_s, n, p, k_0, k_1) = (0.004, 0.07, 0.82, 2, 5, 0.2, 0.222)$ with the initial point $(K_0, S_0) = (0.05, 6.0)$.

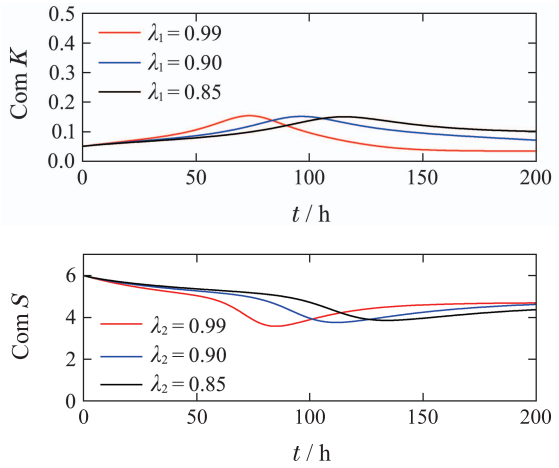

Fig. 1 Simulated Com K and Com S activities with $\gamma_1, \gamma_2 = 0.99, 0.90, 0.85$ during competence for system (25)

In addition, to increase the credibility for the optimization performance of the proposed ICS algorithm, another two fractional-order dynamical biological systems are also chosen as the examples for numerical simulations. The second example, fractional-order cellular neural network (CNN) [7], is described as

$$\begin{cases} {}_0D_{\mathrm{t}}^{q_1} x_1 = -x_1 + p_1 f(x_1) - s f(x_2) - s f(x_3), \\ {}_0D_{\mathrm{t}}^{q_2} x_2 = -x_2 - s f(x_1) + p_2 f(x_2) - r f(x_3), \\ {}_0D_{\mathrm{t}}^{q_3} x_3 = -x_3 - s f(x_1) + r f(x_2) + p_3 f(x_3), \end{cases} \tag{26}$$

where the activation function is defined as

$$f(x_j) = \frac{1}{2}(|x_j + 1| - |x_j - 1|),\ j = 1, 2, 3,$$

and the system (26) is chaotic when parameters $(p_1, p_2, p_3, s, r) = (1.24, 1.1, 1, 3.21, 4.4)$, and system orders $(q_1, q_2, q_3) \equiv (q = 0.99)$ with initial conditions $(x_1(0), x_2(0), x_3(0)) = (0.1, 0.1, 0.1)$.

The third example, fractional-order Lotka-Volterra system[7], is described as

$$\begin{cases} {}_0D_{\mathrm{t}}^{q_1} x = ax - bxy + ex^2 - szx^2, \\ {}_0D_{\mathrm{t}}^{q_2} y = -cy + dxy, \\ {}_0D_{\mathrm{t}}^{q_3} z = -pz + szx^2, \end{cases} \tag{27}$$

where the system (27) is chaotic when parameters

$(a, b, c, d, e, p, s) = (1, 1, 1, 1, 2, 3, 2.7)$, and system orders $(q_1, q_2, q_3) \equiv (q = 0.95)$ with initial conditions $(x(0), y(0), z(0)) = (1, 1.4, 1)$.

4.2 Numerical simulations

To demonstrate the performance of the proposed ICS algorithm, ICS is compared with the basic CS algorithm, three promising CS variants and three other state-of-the-art algorithms, namely ACS[55], ICS 2011[56], NNCS–F[57], PSO[58], DE[24], STA[28–31]. The parameter settings of all the comparison methods are given in Table 1. In order to eliminate the difference of each experiment, each algorithm is executed 15 times and all runs are terminated after the predefined maximum iteration number is reached. To make a fair comparison, the same parameter settings are employed to the aforementioned three examples in numerical simulations.

Table 1 Parameter settings of different methods

Method	Parameters
CS	$\mathrm{NP} = 40$, $P_{\mathrm{a}} = 0.25$
ACS	$\mathrm{NP} = 40$, $P_{\mathrm{a}} = 0.25$
ICS 2011	$\mathrm{NP} = 40$, $P_{\mathrm{a,min}} = 0.05$, $P_{\mathrm{a,max}} = 0.5$, $\alpha_{\min} = 0.01$, $\alpha_{\max} = 0.5$
NNCS–F	$\mathrm{NP} = 40$, $P_{\mathrm{a}} = 0.25$, $p = 0.25$
PSO	$\mathrm{NP} = 40$, $\omega_{\min} = 0.4$, $\omega_{\max} = 0.9$, $c_1 = c_2 = 2$
DE	$\mathrm{NP} = 40$, $F = 0.5$, $CR = 0.7$
STA	$\mathrm{SE} = 40$, $\alpha_{\min} = 1\mathrm{E}-4$, $\alpha_{\max} = 1$, $\beta = 1$, $\gamma = 1$, $\delta = 1$, $fc = 2$
ICS	$\mathrm{NP} = 40$, $P_{\mathrm{a}} = 0.25$

For the first example, the proposed ICS algorithm is applied to study the inverse problem and determine unknown parameters of system (25), i.e., for all given values of $K(t)$ and $S(t)$ at $t = t_k \in [0, \mathscr{T}]$, $(k = 1, \cdots, N)$, $N = 100$, $\mathscr{T} = 20$ and given initial conditions (K_0, S_0). To validate the effectiveness and efficiency of ICS and for ease of illustration, γ_1, γ_2 and b_s are taken as unknown parameters needed to be estimated, the searching spaces of which are set to $(\gamma_1, \gamma_2, b_{\mathrm{s}}) \in [0.8, 1.2] \times [0.8, 1.2] \times [0.6, 1.0]$. The true values of γ_1, γ_2 and b_s are set as 0.90, 0.90 and 0.82 in advance. The maximum iteration number is set to 200.

The statistical results of the average estimated parameters with corresponding relative error values and the objective function values for system (25) over 15 independent runs are presented in Table 2. According to Table 2, we can see that all the algorithms have a certain capability of estimating parameters, among which ICS is the best one among the seven algorithms in terms of solutions accuracy. From Table 2, it can be easily found that the estimated values generated by ICS are closer to the true parameter values, which shows that ICS is more accurate than CS, ACS, ICS 2011, NNCS-F, PSO, DE and STA. Besides, from the experimental results marked in bold, ICS obtains the smallest relative error values for the corresponding estimated parameters of system (25) only except for γ_2, where its relative error value is slightly worse than that of DE. However, the mean and the standard deviation of objective function values obtained by ICS are significantly better than those calculated by the comparison algorithms namely CS, ACS, ICS 2011, NNCS–F, PSO, DE and STA. This indicates that ICS can bring estimated values with much higher accuracy to the parameter estimation problem for the nonlinear fractional dynamical model of competence induction in B. *Subtilis* bacteria (25).

Table 2 Statistical results of system (25) using different methods

Method	CS	ACS	ICS 2011	NNCS–F	PSO	DE	STA	ICS
γ_1	0.899984	0.900000	0.900001	0.900001	0.873333	0.900000	0.899585	0.900000
$\frac{\lvert\gamma_1 - 0.90\rvert}{0.90}$	1.81E−05	1.95E−09	1.06E−06	1.33E−06	2.96E−02	9.05E−14	4.61E−04	**1.28E−14**
γ_2	0.899995	0.900000	0.900000	0.907061	0.900001	0.900000	0.899977	0.900000
$\frac{\lvert\gamma_2 - 0.90\rvert}{0.90}$	5.04E−06	3.25E−10	4.95E−07	1.17E−06	7.85E−03	**2.84E−15**	2.60E−05	3.45E−15
b_{s}	0.820000	0.820000	0.820000	0.820000	0.820469	0.820000	0.819999	0.820000
$\frac{\lvert b_{\mathrm{s}} - 0.82\rvert}{0.82}$	2.57E−07	1.65E−11	3.38E−08	7.93E−08	5.72E−04	1.90E−15	1.10E−06	**2.71E−16**
F_{Avg}	5.42E−06	5.96E−10	1.09E−06	2.11E−06	5.60E−03	2.13E−13	3.16E−04	**7.58E−15**
F_{Std}	3.02E−06	5.71E−10	5.43E−07	1.27E−06	9.29E−03	1.66E−13	3.48E−04	**4.63E−16**

Figure 2 shows the evolution process of the average objective function values via ICS, CS, ACS, ICS 2011, NNCS–F, PSO, DE and STA. From Fig. 2, it can be found that the objective function value obtained by ICS decreases to zero much faster than the other algorithms. From the foregoing discussion, it can be concluded that ICS demonstrates better effectiveness and efficiency in parameter estimation of the fractional nonlinear dynamical model of the competence induction.

Moreover, since that researches have shown that in the cell stochastic effects in gene expression would generate prominent variability[12]. Hence, a further test is carried out to find out whether the proposed pa-

rameter estimation method is tolerant to the measurement error. We assume that there are some perturbations in measurement, i.e., $K(t) = \bar{K}(t) + \xi_{\mathrm{K}}(t)$ and $S(t) = \bar{S}(t) + \xi_{\mathrm{S}}(t)$, where $\bar{K}(t)$ and $\bar{S}(t)$ are solutions of system (25), and the perturbations are given in the following forms similar as[13]

$$\xi_{\mathrm{K}}(t) = C_{\mathrm{K}}(-0.5 + \mathrm{Rand_K}), \tag{28}$$

$$\xi_{\mathrm{S}}(t) = C_{\mathrm{S}}(-0.5 + \mathrm{Rand_S}), \tag{29}$$

where C_{K} and C_{S} are scale constants, $\mathrm{Rand_K}$ and $\mathrm{Rand_S}$ are random numbers in the range $[0, 1]$. Comparison of simulated Com K and Com S activities with and without noise perturbation during competence are depicted in Fig. 3 with $(\gamma_1, \gamma_2) = (0.9, 0.9)$.

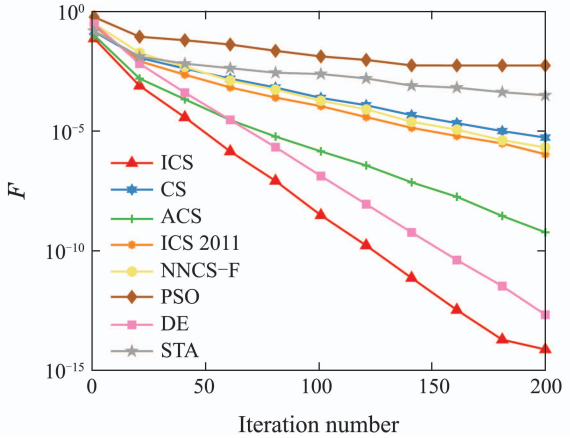

Fig. 2 Evolution process of the average objective function values for system (25) based on different methods

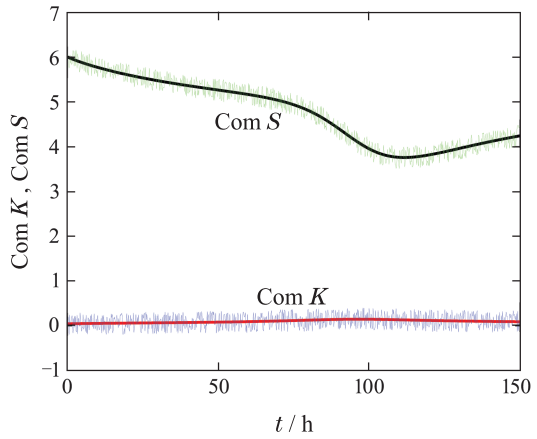

Fig. 3 Comparison of simulated Com K (red continuous line) & Com S (block continuous line) activities with given data Com K (blue noise line) & Com S (green line) under noise perturbation with $C_K = C_S = 0.3$ during competence with $(\gamma_1, \gamma_2) = (0.9, 0.9)$

In this test, the scale constants C_{K} and C_{S} are set to 0.01, and the maximum iteration number is set to 1000. Unknown parameters γ_1, γ_2 and b_{s} are estimated through ICS via a single run. Under noise perturbation, we obtain the estimated values:

$$\gamma_1 = 0.900, \gamma_2 = 0.904, b_s = 0.820,$$

which illustrates that the robustness of ICS and can bring solutions with high quality even placed in perturbation environment.

For the second example, the true values of systematic parameters p_1, r and fractional order q in system (26) are randomly considered as unknown parameters which need to be estimated. The corresponding searching spaces are set to $(p_1, r, q) \in [0.1, 3] \times [2, 8] \times [0.8, 1.2]$. Besides, the length of the sampled data N is set to 200, and the step size is set to 0.005. The maximum iteration number is set to 200.

The statistical results including the mean estimated values, the relative error values, and the objective function values via different methods over 15 independent runs are summarized in Table 3. In addition, the evolution process of the average results of the objective function values for system (26) is shown in Fig. 4. According to Table 3, it can be noted that ICS obtains more accurate estimated values than CS, ACS, ICS 2011, NNCS–F, PSO, DE and STA. Besides, it can also be clearly seen that the relative error values obtained by ICS are all smaller than those of CS, ACS, ICS 2011, NNCS–F, PSO, DE and STA, which can further prove that ICS is able to achieve a higher calculation accuracy. What's more, the mean and the standard deviation of objective function values obtained by ICS are also superior to the comparison algorithms. Fig. 4 depicts the convergence process of the average results of the objective function values. From Fig. 4, it is obvious that ICS can converge more rapidly than the other algorithms. Therefore, the ICS algorithm demonstrates the good performance in aspects of robustness and convergence accuracy, which is highly competitive with those of CS, ACS, ICS 2011, NNCS–F, PSO, DE and STA for fractional-order cellular neural network (CNN) (26).

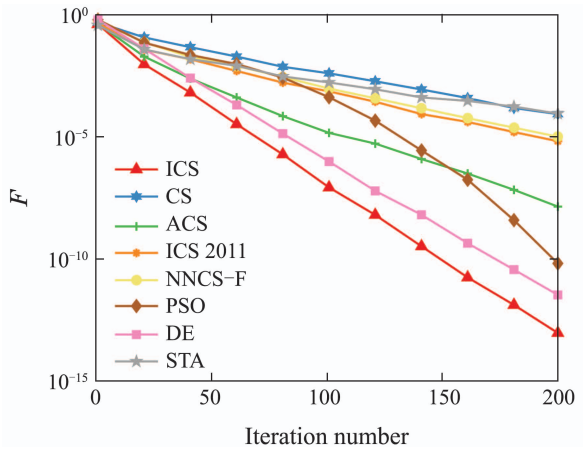

Fig. 4 Evolution process of the average objective function values for system (26) based on different methods

Table 3 Statistical results of system (26) using different methods

Method	CS	ACS	ICS 2011	NNCS–F	PSO	DE	STA	ICS
p_1	1.240031	1.240000	1.240001	1.240000	1.239995	1.240000	1.239977	1.240000
$\frac{\lvert p_1-1.24\rvert}{1.24}$	2.48E−05	8.66E−09	6.69E−07	4.35E−11	3.81E−06	3.35E−13	1.86E−05	**2.67E−14**
r	4.399972	4.400000	4.399999	4.400000	4.400004	4.400000	4.400012	4.400000
$\frac{\lvert r-4.40\rvert}{4.40}$	6.47E−06	1.92E−09	2.25E−07	9.99E−12	8.19E−07	5.17E−14	2.65E−06	**6.06E−15**
q	0.989996	0.990000	0.990000	0.990000	0.990001	0.990000	0.990002	0.990000
$\frac{\lvert q-0.99\rvert}{0.99}$	3.76E−06	1.05E−09	2.85E−07	2.63E−12	5.36E−07	6.45E−14	2.23E−06	**4.60E−15**
$F_{\rm Avg}$	8.64E−05	1.41E−08	6.71E−06	6.66E−11	1.03E−05	3.39E−12	9.09E−05	**9.38E−14**
$F_{\rm Std}$	6.62E−05	2.03E−08	4.49E−06	3.97E−11	6.06E−06	2.45E−12	1.04E−04	**1.20E−13**

For the third example, the true values of systematic parameters e, p and fractional order q in system (27) are randomly assumed as unknown parameters needed to be estimated, the searching spaces of which are set to $(e, p, q) \in [0.1, 10] \times [0.1, 10] \times [0.8, 1.2]$. The length of the sampled data N is set to 200, the step size is set to 0.005, and the maximum iteration number is set to 100.

Table 4 records the comparison results of the mean estimated values with corresponding relative error values and the objective function values by different methods over 15 independent runs. The evolution process of the average results of the objective function values for system (27) are plotted in Fig. 5. Based on Table 4 and Fig. 5, it can be observed that ICS is much better than the other six algorithms, and supplies more precise and robust results with faster convergence speed. In particular, the relative error values produced by ICS marked in bold are all smaller than those by CS, ACS, ICS 2011, NNCS–F, PSO, DE and STA. The relative error values are also an important indicator to assess the accuracy of estimated values. Accordingly, the estimated values generated by ICS are closer to the true parameter values than those by the other algorithms. Besides, the objective function values of ICS decline significantly faster than the other comparison algorithms. In general, the comprehensive performance of ICS is remarkably superior to all the listed comparison algorithms in terms of convergence precision and searching efficiency.

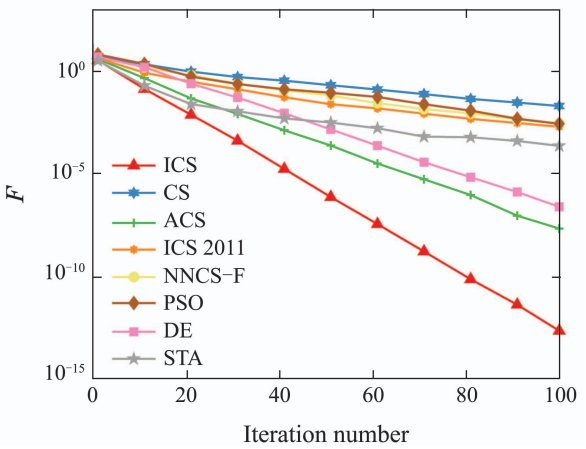

Fig. 5 Evolution process of the average objective function values for system (27) based on different methods

Table 4 Statistical results of system (27) using different methods

Method	CS	ACS	ICS 2011	NNCS–F	PSO	DE	STA	ICS
e	2.000530	2.000000	2.000034	1.999927	2.000122	2.000000	2.000001	2.000000
$\frac{\lvert e-2.00\rvert}{2.00}$	2.65E−04	3.72E−10	1.72E−05	3.64E−05	6.08E−05	6.97E−09	4.76E−07	**4.00E−15**
p	3.001392	3.000000	3.000127	2.999911	2.999612	3.000000	3.000005	3.000000
$\frac{\lvert p-3.00\rvert}{3.00}$	4.64E−04	1.21E−10	2.98E−05	4.22E−05	1.29E−04	1.37E−08	1.65E−06	**7.11E−15**
q	0.950082	0.950000	0.950001	0.950000	0.949993	0.950000	0.950000	0.950000
$\frac{\lvert q-0.95\rvert}{0.95}$	8.58E−05	1.08E−11	1.55E−06	4.69E−07	7.26E−06	8.06E−10	1.46E−07	**3.51E−16**
$F_{\rm Avg}$	2.05E−02	2.26E−08	1.99E−03	2.79E−03	2.88E−03	2.56E−07	2.35E−04	**2.41E−13**
$F_{\rm Std}$	1.06E−02	1.76E−08	9.46E−04	2.25E−03	2.55E−03	1.74E−07	1.48E−04	**1.11E−13**

5 Conclusions

In this paper, an improved cuckoo search (ICS) algorithm is proposed for tackling inverse problems that correspond to biological systems of fractional nonlinear dynamical models. Parameter estimation is converted into a multi-dimensional optimization problem by treating both systematic parameters and fractional derivative orders as independent unknown parameters to be estimated. Particularly, in ICS, a simple adaptive parameter control mechanism is introduced, at the mean time, the opposition-based learning (OBL) method is incorporated to the presented algorithm so that it can accelerate convergence speed and improve the accuracy of the estimated values. Thereinto, the adaptive parameter control mechanism dynamically updates the control parameters based on a Cauchy distribution and the Lehmer mean at each iteration process. Moreover, OBL favors increasing the chance of finding solutions which are closer to the global optima. The techniques help to enhance the comprehensive performance of optimization. Finally, numerical simulations are conducted on three typical fractional-order dynamical biological systems, where the condition with measurement error and noisy data is also investigated. From the simulation results, it can be seen that parameters are estimated successfully through ICS, and are much closer to the true matching values than the comparison algorithms, which demonstrate the effectiveness, efficiency and superiority of ICS. These methods and techniques can also be extended to other kinds of parameter estimation problems of nonlinear fractional-order systems in various fields.

References:

[1] PODLUBNY I. *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications*. San Diego: Academic Press, 1998.

[2] MAINARDI F. *Fractional Calculus and Waves in Linear Viscoelasticity: An Introduction to Mathematical Models*. London: Imperical College Press, 2010.

[3] DUMITRU B, KAI D, ENRICO S. *Fractional Calculus: Models And Numerical Methods*. Boston: World Scientific, 2012.

[4] LIU F, YANG C, BURRAGE K. Numerical method and analytical technique of the modified anomalous subdiffusion equation with a nonlinear source term. *Journal of Computational and Applied Mathematics*, 2009, 231(1): 160 – 176.

[5] LAZAREVIĆ M P. Elements of mathematical phenomenology of self-organization nonlinear dynamical systems: Synergetics and fractional calculus approach. *International Journal of Non-Linear Mechanics*, 2015, 73(1): 31 – 42.

[6] IONESCU C, LOPES A, COPOT D, et al. The role of fractional calculus in modeling biological phenomena: A review. *Communications in Nonlinear Science and Numerical Simulation*, 2017, 51: 141 – 159.

[7] PETRÁŠ I. Fractional-order chaotic systems. *Fractional-Order Nonlinear Systems*. Berlin: Springer, 2011: 103 – 184.

[8] BERRY H. Monte Carlo simulations of enzyme reactions in two dimensions: fractal kinetics and spatial segregation. *Biophysical Journal*, 2002, 83(4): 1891 – 1901.

[9] DUPIN E, REAL C, GLAVIEUX-PARDANAUD C, et al. Reversal of developmental restrictions in neural crest lineages: transition from Schwann cells to glial-melanocytic precursors in vitro. *Proceedings of the National Academy of Sciences*, 2003, 100(9): 5229 – 5233.

[10] LEE E J, RUSSELL T, HURLEY L, et al. Pituitary transcription factor-1 induces transient differentiation of adult hepatic stem cells into prolactin-producing cells in vivo. *Molecular Endocrinology*, 2005, 19(4): 964 – 971.

[11] MEEKS J C, CAMPBELL E L, SUMMERS M L, et al. Cellular differentiation in the cyanobacterium nostoc punctiforme. *Archives of Microbiology*, 2002, 178(6): 395 – 403.

[12] SÜEL G M, GARCIA-OJALVO J, LIBERMAN L M, et al. An excitable gene regulatory circuit induces transient cellular differentiation. *Nature*, 2006, 440(7083): 545 – 551.

[13] LIU F, BURRAGE K. Novel techniques in parameter estimation for fractional dynamical models arising from biological systems. *Computers & Mathematics with Applications*, 2011, 62(3): 822 – 833.

[14] BECK J V, ARNOLD K J. *Parameter Estimation in Engineering and Science*. New York: John Wiley & Sons, 1977.

[15] YAO L, SETHARES W A. Nonlinear parameter estimation via the genetic algorithm. *IEEE Transactions on Signal Processing*, 1994, 42(4): 927 – 935.

[16] RODRIGUEZ-FERNANDEZ M, EGEA J A, BANGA J R. Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC Bioinformatics*, 2006, 7(1): 483 – 500.

[17] MOLES C G, MENDES P, BANGA J R. Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, 2003, 13(11): 2467 – 2474.

[18] BARNSLEY M F, ERVIN V, HARDIN D, et al. Solution of an inverse problem for fractals and other sets. *Proceedings of the National Academy of Sciences of the United States of America*, 1986, 83(7): 1975.

[19] KUNZE H E, HICKEN J E, VRSCAY E R. Inverse problems for ODEs using contraction maps and suboptimality of the 'collage method'. *Inverse Problems*, 2004, 20(3): 977.

[20] DENG X, WANG B, LONG G. The Picard contraction mapping method for the parameter inversion of reaction-diffusion systems. *Computers & Mathematics with Applications*, 2008, 56(9): 2347 – 2355.

[21] DJOUAMBI A, CHAREF A, BESANÇON A. Optimal approximation, simulation and analog realization of the fundamental fractional order transfer function. *International Journal of Applied Mathematics and Computer Science*, 2007, 17(4): 455 – 462.

[22] BENCHELLAL A, POINOT T, TRIGEASSOU J C. Approximation and identification of diffusive interfaces by fractional models. *Signal Processing*, 2006, 86(10): 2712 – 2727.

[23] GOLDBERG D E, HOLLAND J H. Genetic algorithms and machine learning. *Machine Learning*, 1988, 3(2): 95 – 99.

[24] STORN R, PRICE K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341 – 359.

[25] KENNEDY J. Particle swarm optimization. *Encyclopedia of Machine Learning*. Boston: Springer, 2011: 760 – 766.

[26] YANG X S, DEB S. Cuckoo search via Lévy flights. *Proceeings of World Congress on Nature & Biologically Inspired Computing*. Coimbatore, India: IEEE, 2009: 210 – 214.

[27] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 2007, 39(3): 459 – 471.

[28] ZHOU X, ZHOU C, GUI W. State transition algorithm. *Journal of Industrial & Management Optimization*, 2013, 8(4): 1039 – 1056.

[29] ZHOU X, ZHOU C, GUI W. A matlab toolbox for continuous state transition algorithm. *Proceedings of the 35th Chinese Control Conference*. Chengdu: IEEE, 2016: 9172 – 9177.

[30] ZHOU X, ZHOU C, GUI W. Nonlinear system identification and control using state transition algorithm. *Applied Mathematics and Computation*, 2014, 226(1): 169 – 179.

[31] ZHANG F, YANG C, ZHOU X, et al. Fractional-order PID controller tuning using continuous state transition algorithm. *Neural Computing and Applications*, 2018, 29(10): 795 – 804.

[32] SUN X, GONG D. Surrogate model-assisted interactive genetic algorithms with individual's fuzzy and stochastic fitness. *Journal of Control Theory and Applications*. 2010, 8(2): 189 – 199.

[33] GUI Haixia, ZHANG Guofu, SU Zhaopin, et al. An algorithm for overlapping coalition structure generation based on differential evolution and encoding repairs. *Control Theory & Applications*, 2018, 35(2): 215 – 223.
(桂海霞, 张国富, 苏兆品, 等. 带有非线性感染率的SIRS模型的建立与稳定性分析. 控制理论与应用, 2018, 35(2): 215 – 223.)

[34] LI Wenfeng, CAO Yulian, ZHANG Han. Hybrid particle swarm optimization algorithm with simple and efficient coupling strategy. *Control Theory & Applications*, 2018, 35(1): 13 – 23.
(李文锋, 曹玉莲, 张汉. 简单高效耦合策略的粒子群混合算法. 控制理论与应用, 2018, 35(1): 13 – 23.)

[35] GAO F, FEI F, LEE X, et al. Inversion mechanism with functional extrema model for identification incommensurate and hyper fractional chaos via differential evolution. *Expert Systems with Applications*, 2014, 41(4): 1915 – 1927.

[36] WANG F, HE X S, WANG Y, et al. Markov model and convergence analysis based on cuckoo search algorithm. *Computer Engineering*, 2012, 38(11): 180 – 185.

[37] YANG X S, DEB S. Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, 2010, 1(4): 330 – 343.

[38] GANDOMI A H, YANG X S, ALAVI A H. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers*, 2013, 29(1): 17 – 35.

[39] WOLPERT D H, MACREADY W G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 67 – 82.

[40] ISLAM S M, DAS S, GHOSH S, et al. An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2012, 42(2): 482 – 500.

[41] LIN J, CHEN C. Parameter estimation of chaotic systems by an oppositional seeker optimization algorithm. *Nonlinear Dynamics*, 2014, 76(1): 509 – 517.

[42] SAMKO S G, KILBAS A A, Marichev O I. *Fractional Integral and Derivatives Theory and Applications*. New York: Gordon and Breach, 1993.

[43] LUCHKO Y, GORENFLO R. An operational method for solving fractional differential equations with the Caputo derivatives. *Acta Mathematica Vietnamica*, 1999, 24(2): 207 – 233.

[44] DIETHELM K, FORD N J, FREED A D. A predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dynamics*, 2002, 29(1/4): 3 – 22.

[45] YANG X S. *Cuckoo Search and Firefly Algorithm: Theory and Applications*. Heidelberg: Springer, 2014.

[46] ZAHARIE D. On the explorative power of differential evolution. *Verhandlungen Der Deutschen Gesellschaft Fur Innere Medizin*, 2001, 39: 551 – 554.

[47] DASGUPTA S, DAS S, BISWAS A, et al. On stability and convergence of the population-dynamics in differential evolution. *AI Communications*, 2009, 22(1): 1 – 20.

[48] ZHANG J, SANDERSON A C. JADE: adaptive differential evolution with optional external archive. *IEEE Transactions on evolutionary computation*, 2009, 13(5): 945 – 958.

[49] TIZHOOSH H R. Opposition-based learning: a new scheme for machine intelligence. *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce*. Vienna, Austria: IEEE, 2005: 695 – 701.

[50] HU W, YU Y, ZHANG S. A hybrid artificial bee colony algorithm for parameter identification of uncertain fractional-order chaotic systems. *Nonlinear Dynamics*, 2015, 82(3): 1441 – 1456.

[51] WANG L, YIN Y, ZHONG Y. Cuckoo search with varied scaling factor. *Frontiers of Computer Science*, 2015, 9(4): 623 – 635.

[52] WANG Y, CAI Z, ZHANG Q. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1): 55 – 66.

[53] TURGAY K, HAHN J, BURGHOORN J, et al. Competence in bacillus subtilis is controlled by regulated proteolysis of a transcription factor. *The EMBO Journal*, 1998, 17(22): 6730 – 6738.

[54] OGURA M, LIU L, LACELLE M, et al. Mutational analysis of Com S: Evidence for the interaction of Com S and MecA in the regulation of competence development in Bacillus subtilis. *Molecular Microbiology*, 1999, 32(4): 799 – 812.

[55] NAIK M K, PANDA R. A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition. *Applied Soft Computing*, 2016, 38: 661 – 675.

[56] VALIAN E, MOHANNA S, TAVAKOLI S. Improved cuckoo search algorithm for feedforward neural network training. *International Journal of Artificial Intelligence & Applications*, 2011, 2(3): 36 – 43.

[57] WANG L, ZHONG Y, YIN Y. Nearest neighbour cuckoo search algorithm with probabilistic mutation. *Applied Soft Computing*, 2016, 49(): 498 – 509.

[58] TANG Y, GUAN X. Parameter estimation for time-delay chaotic system by particle swarm optimization. *Chaos, Solitons & Fractals*, 2009, 40(3): 1391 – 1398.

作者简介:

卫佳敏 博士研究生, 目前研究方向为分阶系统建模与控制优化, E-mail: 16118414@bjtu.edu.cn;

于永光 教授, 目前研究方向为分数阶微分方程及其应用和混沌控制与同步, E-mail: ygyu@bjtu.edu.cn;

张 硕 副教授, 目前研究方向为分数阶神经网络动力学分析及控制, E-mail: zhangshuo1018@nwpu.edu.cn.