Egyptian Informatics Journal 19 (2018) 151–163

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

Chaotic based differential evolution algorithm for optimization of baker's yeast drying process

Uğur Yüzgeç *, Mehmet Eser

Department of Computer Engineering, Bilecik Seyh Edebali University, Bilecik, Turkey

A R T I C L E I N F O

Article history:
Received 5 September 2017
Revised 11 January 2018
Accepted 21 February 2018
Available online 27 February 2018

Keywords:
Chaotic
Differential evolution
Optimization
Drying process

A B S T R A C T

Chaotic based Differential Evolution (CDE) algorithm is presented to determine the optimal control variables for the optimization of Baker's Yeast drying process. The chaotic system is proposed to determine the initial population, to select the trial individuals from the population in the mutation operation instead of the random number generator. The random values produced by the random number generator are likely to be similar or same values with each other. In this study, four different chaotic systems, such as Lorenz attractor, Rössler attractor, Chua circuit and Mackey-Glass equation, are solved by Runge-Kutta method to produce the random values of the initial individuals. To demonstrate the performance of the CDE algorithms, ten optimization problems are taken from the literature. Furthermore, the performances of the proposed CDE algorithms are compared with the classic Differential Evolution (DE) algorithm, Particle Swarm Optimization (PSO) algorithm, Artificial Bee Colony (ABC) algorithm, Simulated Annealing (SA) algorithm, Touring Ant Colony Optimization (TACO) algorithm in terms of the mean best solution, the number of function evaluations (NFE) and CPU-time metrics. At the same time, the proposed CDE algorithms are implemented for numerical optimization problems based on the IEEE Congress on Evolutionary Computation (CEC) 2014 test suite. For the optimization of baker's yeast drying process, there are four significant parameters, such as product quality, drying total time, energy cost of air and the final moisture content. The proposed CDE algorithms and classic DE algorithm are applied for the same optimization problem that is taken from a baker's yeast producer in Turkey. The experimental results prove that the proposed CDE algorithms are able to provide very competitive results.

1. Introduction

Differential Evolution (DE) algorithm is a powerful heuristic method for global optimization problems, was introduced by Storn and Price [31,32,37]. This population based heuristic optimization algorithm has drawn the interest of researchers in many scientific fields. The DE algorithm has happened to more popular step by step and it has been used in a lot of useful cases due to ease and the good convergence in the optimization problems [4].

The principle of DE algorithm is basically based on adding the difference between two individuals to a third individual in population. It differs from other heuristic algorithms in the mutation, crossover and selection stages. Unlike the procedures based on random number generator in evolutionary algorithms such as genetic algorithms, DE algorithm uses the differences between individuals in the population to form the next generation [10]. Furthermore, DE algorithm has got few control parameters, such as scaling factor, crossover probability constant and population size, which are used during the optimization process like the other evolutionary algorithms. These control parameters have to be determined carefully to increase the solution quality and the algorithm efficiency. The robustness and effectiveness of DE algorithm are based on the suitable settings of the control parameters [44].

In addition to these parameters, the other important thing is determining the initial population by random number generator. In DE algorithm, the individuals' initial values in the population which are produced by the random number generator are likely

* Corresponding author.
E-mail address: ugur.yuzgec@bilecik.edu.tr (U. Yüzgeç).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2018.02.001
1110-8665/© 2018 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.

to be similar or same values with each other. This is an undesirable situation because of reducing the diversity in the population. In this paper, the new methods based on the chaotic functions were proposed instead of the classic random procedure. Chaos functions have got applications, such as observing the weather in meteorology area [38], cryptography in computer science area [43], predicting gas solubility in chemical engineering [39], finance modeling in economics area [14] and hydrology in biology area [40]. Chaotic functions have the behavior of dynamic systems which are highly sensitive to initial conditions. Each point in a chaotic system is arbitrarily close to other points with different future trajectories. As a result, an small change in the existing trajectory can lead to considerably different behavior [11,12].

In the literature, there have been a large number of publications regarding improvements and applications of the DE algorithm in many fields, such as chemical optimization [44], image segmentation [27], human detection [5], economic dispatch optimization [37], shape matching problem [1], object detection [42], among the others [48]. Babu and Angira [2] proposed the modified selection procedure that was used for a single array, for the optimization of non-linear chemical processes. The proposed modified DE algorithm was compared to classic DE algorithm for optimization problems of benchmark test functions and selected non-linear chemical processes. Babu and Munawar [3] introduced DE algorithm's ten different strategies for the optimal design of shell-and-tube heat exchangers. In [6], the ranking-based mutation operator was integrated into the original DE algorithm to accelerate the convergence rate for multi objective optimization problems. Draa et al. [10] presented the idea that is about the tuning of the DE's parameters using sinusoidal function. There were six different configurations of this sinusoidal function based parameter adjustment for both scale factor and crossover constant between the upper and lower bounds of these parameters. The application of this proposed strategies is very hard for the real time microcontroller based implementations. Gong et al. [13] proposed two different adaptive strategy selection, namely probability matching and adaptive pursuit to select the most suitable strategy during the optimization process. Although the proposed selection methods brought some advantages, time complexity of the DE algorithm including these methods increases. In [26], a new mutation strategy that is based on the weighted difference vector between the best and the worst individuals was introduced. The authors presented the performance and the comparison results in their paper. A concept which is called opposition-based differential evolution (ODE) to accelerate the convergence rate of DE algorithm was presented by Rahnamayan et al. [34,35]. ODE uses the opposite numbers during the population initialization and also during generation jumping. The proposed algorithm considers an individual in population and its corresponding opposite individual is calculated to accomplish a better solution at each generation according to the jumping rate [34,35]. Zhang and Yuen [50] introduced the new method about the mutation operator to accelerate the convergence rate. In the study by Poikolainen et al. [29], a software module consists of three stages was presented to determine the most interesting areas of the search domain. But there was no discussion about the run times of the proposed algorithms. Qi et al. [33] proposed the a hybrid immune multi-objective optimization algorithm with differential evolution inspired recombination. In the proposed recombination operator, two types of search directions were determined according to the other two neighboring individuals in the current population. Mohamed [52] introduced a new triangular mutation rule for DE algorithm. In [53] and [55], the DE algorithm based on a new mutation rule was proposed. It utilizes the information of good and bad vectors in the population. Mohamed and Suganthan [54] presented a new triangular mutation operator for solving global numerical optimization problems.

In the studies regarding combination with chaotic dynamic systems and DE algorithm, a logistic map based DE algorithm was presented by Zhang et al. [49] for short-term scheduling and a self-adaptive chaotic DE algorithm using gamma distribution was introduced by Coelho et al. [9]. Both of these papers include the logistic map as the chaos function. The logistic map is a polynomial mapping of second degree in discrete time. In terms of ease solution, the authors generally prefer the chaotic maps, such as logistic map, baker's map, Henon map, etc., instead of chaotic systems in continuous time.

In this study, the new random number generator based on the solution of the chaotic functions was proposed for selection of the candidates from population at the mutation, crossover operations and for the initialization of the population in DE algorithm. There are four chaotic systems, such as Lorenz attractor [11,12], Rössler attractor [45], Chua circuit and Mackey-Glass system to use in the random number generator procedure. To evaluate the performances of the proposed CDE algorithms, ten benchmark functions are taken from the literature and the popular heuristic algorithms, such as DE, PSO, ABC, SA and TACO algorithms, are compared with the proposed CDE algorithms. Besides, we used the CEC 2014 test suite benchmark problems to evaluate the performance of proposed CDE algorithms. For the CDE algorithms and classic DE algorithm, the results obtained during baker's yeast drying optimization process are compared with each others.

The paper is organized as follows. The differential evolution algorithm is briefly presented in Section 2. Section 3 gives information about the four different chaotic system definition being used to generate random number in the proposed CDE algorithm. The concept of the proposed CDE algorithms is presented in Section 4. The next section includes the information about the benchmark functions used to evaluate the performances of the CDE algorithms. In Section 6, there is short information regarding the optimization problem in the baker's yeast drying process. In Section 7, the performances of the CDE algorithms for optimization problems are discussed according to the mean best solution, the number of function evaluations (NFE) and CPU-time metrics. Besides, the comparison between the proposed CDE algorithms and the classic DE algorithm is presented for optimization of the baker's yeast drying process. Finally, the paper is concluded in Section 8.

2. Differential evolution algorithm

Differential evolution (DE) algorithm is a simple powerful and influential evolutionary algorithm for solution of the global optimization, introduced by Price and Storn [32]. On the contrary simple genetic algorithm which uses binary coding to represent the individuals in the population, DE algorithm uses floating point coding to stand for each individuals. The important idea of DE algorithm is based on generating trial parameter vectors. These vectors are obtained by adding the difference between two individuals to a third individual in population. Mutation and crossover operators are used to generate new individuals, and then selection operator determines which of the vectors will carry on into the next generation [31].

The structure of the DE algorithm resembles the structures of other population-based optimization algorithms. DE consists of three important parameters, such as scaling factor (SF), crossover constant (CR) and population size (PS). A population includes the PS individuals, each of which comprises the value of the variable in the feasible region of the optimization problem [44]. At the beginning of DE algorithm, PS is determined as depend on optimization parameters and it is not changed during the optimization process. The initial population can be chosen randomly. DE algorithms have got three genetic operators, such as mutation, crossover and selection [31,41].

Mutation and crossover operators generate new trial individuals and selection operator determines suitable individuals which have got maximum/minimum fitness values and in this way population consists of the better individuals in that generation [44]. For mutation procedure, there are ten strategies that used in the different DE algorithms. A strategy that works out to be the best for a given problem may not work well when applied for a different problem. The strategy to be adopted for each problem is to be determined separately by trial and error. Five DE strategies used for mutation process are given below:

$$v_{i,g+1} = x_{i,g} + F(x_{b,g} - x_{i,g}) + F(x_{r1,g} - x_{r2,g}) \quad (1)$$

$$v_{i,g+1} = x_{r3,g} + F(x_{r1,g} - x_{r2,g}) \quad (2)$$

$$v_{i,g+1} = x_{b,g} + F(x_{r1,g} - x_{r2,g}) \quad (3)$$

$$v_{i,g+1} = x_{b,g} + F(x_{r1,g} - x_{r2,g}) + F(x_{r3,g} - x_{r4,g}) \quad (4)$$

$$v_{i,g+1} = x_{r1,g} + F(x_{r2,g} - x_{r3,g}) + F(x_{r4,g} - x_{r5,g}) \quad (5)$$

In this equations, $v_{i,g+1}$ denotes the mutant individuals for the next generation, $x_{i,g}$ is the individual with the running index (i), $x_{b,g}$ is the individual which has got the best fitness value in the population, $x_{r1,g}, x_{r2,g},\ldots, x_{r5,g}$ represent the individuals that chosen randomly from the population. The scale factor (F) is a constant value that is in the range from 0 to 2.

In the crossover procedure, according to the comparison of the random number and crossover constant (CR), the trial individual $u_{i,g+1}$ is selected from the current individual or the mutant individual. The crossover equation is given by Eq. (6),

$$u_{i,g+1} = \begin{cases} v_{i,g+1}, & \text{if } r \leqslant CR \\ x_{i,g}, & \text{if } r > CR \end{cases} \quad (6)$$

where r denotes a random number which is in range [0 1]. At the end of mutation and crossover procedures, individuals of next generation are selected from current population by the selection procedure. Selection procedure for the minimization problem is given below:

$$x_{i,g+1} = \begin{cases} u_{i,g+1}, & \text{if } f(u_{i,g+1}) < f(x_{i,g}) \\ x_{i,g}, & \text{otherwise} \end{cases} \quad (7)$$

where $x_{i,g+1}$ denotes the individual with the running index (i) at the next generation and f represents the fitness value. According to the comparison between the fitness value of the trial individual $u_{i,g+1}$ and the target individual with the running index (i) $x_{i,g}$, the individual with the minimum fitness value is selected for the next generation.

According to Storn and Price [41], the selection of scaling factor SF is more sensitive than that of crossover probability constant CR for DE algorithms. In the optimization process, DE algorithm repeats the above three operators until a stop criterion is reached.

3. Chaotic systems

This section consists of the four different chaotic system definition that are used for random number generator in the DE algorithm and the mathematical formulations of these systems. In this study, Lorenz attractor, Rössler attractor, Chua circuit and Mackey-Glass equation were selected among the chaotic systems.

3.1. Lorenz attractor

The Lorenz attractor studied by Edward Lorenz is a chaotic system that includes ordinary differential equations [24]. This attractor presents chaotic attributes for certain parameter values and initial conditions. The model is a chaotic system with three ordinary differential equations known as the Lorenz equations:

$$\begin{aligned} \dot{x} &= s \cdot (y - x) \\ \dot{y} &= x \cdot (r - z) - y \\ \dot{z} &= x \cdot y - b \cdot z \end{aligned} \quad (8)$$

where x, y, z represent the system states and s, r, b denote the system parameters. In this study, these system parameters were determined such as $s = 11, r = 25, b = 8/3$. These equations were obtained from simplified mathematical model developed for atmospheric convection [24].

3.2. Rössler attractor

Rössler attractor is formed by three differential equations that are useful in modeling equilibrium in chemical reactions. The Rössler attractor behaves similarly to the Lorenz attractor, but also be easier to analyze [36]. The equations of the Rössler attractor are given below:

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + a \cdot y \\ \dot{z} &= b + z \cdot (x - c) \end{aligned} \quad (9)$$

where x, y, z denote the system states and a, b, c are the system parameters. In this study, these parameters were selected as $a = 0.2, b = 0.2, c = 5$.

3.3. Chua circuit

Chua's circuit is a electronic circuit known as nonperiodic oscillator. This circuit produces an oscillating waveform that exhibits classic chaos behavior [7,8]. Chua circuit consists of two capacitors, one inductance, one resistance and one Chua diode. As the results of analyzing the Chua circuit, three ordinary differential equations are found as below:

$$\begin{aligned} \dot{x} &= \alpha[y - x - f(x)] \\ \dot{y} &= x - y + z \\ \dot{z} &= -\beta y \end{aligned} \quad (10)$$

where z denotes inductance current, x and y represent the voltage of the C_1 and C_2 capacities. α and β are the parameters determined by the particular values of the circuit components. The function f(x) defines the response of the nonlinear resistance and it's equation is given below

$$\mathrm{f(x)} = m_0 \cdot V_{C_1} + \frac{1}{2} \cdot (m_1 - m_0)(|V_R + B_P| - |V_R - B_P|) \quad (11)$$

In Chua circuit, the parameters were determined as $\alpha = 15.6$, $\beta = 28$, $m_0 = -1.143$, $m_1 = -0.714$.

3.4. Mackey-glass system

Mackey-Glass system exhibits the chaotic behaviors based on the complex rhythms observed in physiology control systems [25]. Mackey-Glass system has got one nonlinear delay-differential equation given below.

$$\dot{x} = \frac{ax(t - \tau)}{1 + x^c(t - \tau)} - bx(t) \quad (12)$$

a, b, c are real numbers and τ denotes the time delay in the Eq.12. In this study, these coefficients are used as $a = 0.2, b = 0.1, c = 10$ and $\tau = 17$.

4. Chaotic based differential evolution algorithm

In this study, the chaotic systems such as Lorenz, Rössler attractors were proposed to generate the individuals in the initial population and the random values (Eq. 1–6) in the mutation and crossover procedures instead of the random number generator. In the optimization process, the initial values produced by the random number generator can be similar or same values with each other. This is undesirable condition because of reducing the diversity in the initial population. In chaotic based differential evolution algorithm, chaotic systems are solved for different initial seed values to obtain different number series at each runs [11,12]. These initial values are the differences between the running/compiling times and the predefined default time. Fig. 1 shows the pseudo code of chaotic DE algorithm.

5. Benchmark functions

Ten benchmark functions from literature [15] were used to test the performance of the proposed chaotic DE algorithm. The information regarding the selected benchmark functions are given in Table 1. These functions have different characteristic. Ackley function is characterized by a nearly flat outer region, and a large hole at the center, Holder table function has many local minimal points and four global minimal points at the corners. Rastrigin is highly multimodal function, but locations of the minimal points are distributed. Rosenbrock is unimodal function, and the global minimum lies in a narrow, parabolic valley. Guinta and Himmelblau functions have not local minimal points and are characterized by almost flat area. Pen holder and Test tube holder functions have many local minimal points. Schweffel is complex function with many local minimal points. Six-hump camel function has got smooth surface and two global minimal points.

6. Optimization of baker's yeast drying process

The fluidized-bed drying technique plays an important role among modern drying methods. It is used mainly for granular materials; on the other hand it is applicable also in the drying of solutions, pastes and liquid sprayed onto the fluidized inert bed [46,47]. The drying method is based on passing hot air through

```
If (individual < min limit value)
        individual ← min limit value
Else If (individual < max limit value)
        individual ← max limit value
End
```

Fig. 2. The boundary strategy of chaotic DE algorithm.

```
Initial individuals_chaotic(Population size, problem dimension, limit values)
Calculate cost function(Population)
Best individual←Individual that has got the best cost function in the population
Best cost function value← Calculate cost function(Best individual)
While(Stop criteria)
            Mutation individual_chaotic (Scale factor, DE strategy)
            Crossover procedure_ chaotic (Crossover constant)
            Calculate cost function(New Population)
               If (Cost(New Individual)≤ Cost(Old Individual))
                     Population← New Individual
               Else
                     Population← Old Individual
               End
Update Best individual and Best cost function value
```

Fig. 1. The pseudo code of chaotic DE algorithm.

Table 1
Benchmark Functions.

Function name	Problem
Ackley (FN1)	$f(x) = -a\exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$
Holder table (FN2)	$f(x) = -\left\lvert\sin(x_1)\cos(x_2)\exp\left(\left\lvert 1 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}\right\rvert\right)\right\rvert$
Rastrigin (FN3)	$f(x) = 10d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right]$
Rosenbrock (FN4)	$f(x) = \sum_{i=1}^{d-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$
Giunta (FN5)	$f(x) = 0.6 + \sum_{i=1}^{d}\left[\sin^2\left(1 - \frac{16}{15}x_i\right) - \frac{1}{50}\sin\left(4 - \frac{64}{15}x_i\right) - \sin\left(1 - \frac{16}{15}x_i\right)\right]$
Penholder (FN6)	$f(x) = -\exp\left\lvert\exp\left(\left\lvert -\frac{\sqrt{x_1^2+x_2^2}}{\pi} + 1\right\rvert\right)\cos(x_1)\cos(x_2)\right\rvert^{-1}$
Himmelblau (FN7)	$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$
Schweffel (FN8)	$f(x) = 418.9829d - \sum_{i=1}^{d}x_i\sin\sqrt{\lvert x_i\rvert}$
Six-hump Camel (FN9)	$f(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$
Testtubeholder (FN10)	$f(x) = -4\left\lvert\exp\left(\left\lvert\cos\left(\frac{1}{200}x_1^2 + \frac{1}{200}x_2^2\right)\right\rvert\right)\sin(x_1)\cos(x_2)\right\rvert$

Table 2
Experimental results (Mean Best & Std Dev.) with 50 independent runs of Chaotic based DE algorithms (CDE_1, CDE_2, CDE_3, CDE_4), DE, PSO, ABC, SA and TACO algorithms. CDE_1: Lorenz, CDE_2: Rossler CDE_3: Chua CDE_4: Mackey-Glass.

FN	Mean Best (Std Dev)								
No	DE	PSO	ABC	SA	TACO	CDE_1	CDE_2	CDE_3	CDE_4
FN1	8.49e−8 (6.03e−8)	2.59e−1 (5.85e−1)	**1.90e−9 (7.68e−9)**	2.7630 (1.8368)	8.69e−1 (1.066)	1.02e−7 (7.11e−8)	6.92e−1 (3.42)	9.81e−8 (6.77e−8)	7.99e−1 (3.95)
FN2	−**19.208 (6.83e−9)**	−15.659 (4.336)	−**19.208 (5.89e−4)**	−19.168 (6.69e−2)	−18.887 (6.33e−1)	−**19.208 (9.89e−9)**	−18.197 (4.045)	−11.777 (7.451)	−**19.208 (2.61e−9)**
FN3	1.99e−2 (1.41e−1)	9.79e−1 (8.17e−1)	**9.29e−5 (4.57e−4)**	8.97e−1 (6.26e−1)	4.39e−1 (5.37e−1)	3.98e−2 (1.97e−1)	5.57e−1 (1.919)	9.95e−1 (1.463)	1.035 (4.919)
FN4	2.15e−7 (7.10e−7)	9.36e−3 (1.68e−2)	1.09e−2 (1.06e−2)	2.97e−2 (4.32e−2)	3.35e−2 (5.47e−2)	**1.00e−8 (1.12e−8)**	1.27e−8 (1.32e−8)	1.09e−8 (1.02e−8)	3.99e−2 (1.14e−1)
FN5	**6.44e−2 (8.14e−9)**	6.45e−2 (1.01e−5)	6.45e−2 (3.07e−10)	6.48e−2 (6.37e−4)	6.45e−2 (3.34e−5)	**6.44e−2 (1.39e−8)**	**6.44e−2 (1.47e−8)**	**6.44e−2 (1.39e−8)**	6.73e−2 (2.03e−2)
FN6	−**9.64e−1 (1.35e−8)**	−9.53e−1 (1.93e−2)	−**9.64e−1 (2.51e−7)**	−9.63e-1 (5.69e−4)	−9.53e−1 (1.82e−2)	−**9.64e−1 (9.18e−9)**	−9.21e−1 (1.31e−1)	−8.35e−1 (1.73e−1)	−9.55e−1 (6.08e−2)
FN7	9.83e−6 (6.36e−5)	1.85e−2 (1.18e−1)	3.14e−4 (7.29e−4)	6.94e−2 (1.33e−1)	8.98e−3 (1.22e−2)	1.28e−8 (1.33e−8)	**1.03e−8 (1.24e−8)**	1.43e−8 (2.38e−8)	8.00e−4 (2.38e−3)
FN8	−8.37e+2 (8.45e−9)	−6.98e+2 (9.95e+1)	−**8.38e+2 (5.93e−1)**	−8.04e+2 (5.16e+1)	−7.05e+2 (9.21e+1)	−7.59e+2 (9.75e+1)	−6.95e+2 (1.82e+2)	−4.18e+2 (2.6e+2)	−7.82e+2 (1.17e+2)
FN9	−**1.0316 (1.38e−8)**	−1.0151 (1.15e−1)	−**1.0316 (6.34e−8)**	−1.0151 (2.76e−2)	−5.50e−1 (3.73e−1)	−**1.0316 (1.39e−8)**	−**1.0316 (1.13e−8)**	−**1.0316 (9.61e−9)**	−**1.0316 (1.01e−5)**
FN10	−**1.087e+1 (3.92e−3)**	−1.081e+1 (9.11e−2)	−**1.087e+1 (1.03e−3)**	−1.082e+1 (6.04e−2)	−1.076e+1 (1.39e−1)	−**1.087e+1 (6.94e−3)**	−**1.087e+1 (3.92e−3)**	−1.083e+1 (4.78e−2)	−1.063e+1 (1.09)

Table 3
Experimental results (NFE^{50} & CPU-time^{50}) with 50 independent runs of Chaotic based DE algorithms (CDE_1, CDE_2, CDE_3, CDE_4), DE, PSO, ABC, SA and TACO algorithms. CDE_1: Lorenz, CDE_2: Rossler CDE_3: Chua CDE_4: Mackey-Glass.

FN	NFE^{50} (CPU-time^{50} sec)[a]								
No	DE	PSO	ABC	SA	TACO	CDE_1	CDE_2	CDE_3	CDE_4
FN1	2968 (0.1682)	2780 (0.1797)	4056 (0.3147)	4000 (0.3305)	4000 (3.6866)	2380 (0.1262)	**2124 (0.1085)**	3156 (0.1825)	3980 (0.1833)
FN2	2760 (0.1466)	2272 (0.1272)	4193 (0.2731)	4000 (0.2687)	3184 (2.2134)	2132 (0.1029)	**1320 (0.0881)**	1508 (0.1077)	2780 (0.1474)
FN3	2144 (0.1139)	2356 (0.1372)	4200 (0.2679)	4000 (0.2674)	1972 (2.4539)	2124 (0.0951)	**1676 (0.0782)**	1920 (0.1015)	2104 (0.1308)
FN4	3912 (0.1969)	2276 (0.1149)	4198 (0.2686)	4000 (0.2698)	3708 (2.1505)	2056 (0.0907)	**1892 (0.0726)**	2208 (0.1225)	4000 (0.2064)
FN5	1152 (0.0649)	1476 (0.0867)	1692 (0.1347)	4000 (0.2900)	4000 (2.6088)	964 (0.0486)	**876 (0.0483)**	1560 (0.0571)	2308 (0.0687)
FN6	2156 (0.1089)	1700 (0.0923)	4194 (0.2609)	4000 (0.2685)	**596** (0.6770)	1912 (0.0824)	1792 **(0.0579)**	1460 (0.0824)	2152 (0.0972)
FN7	3456 (0.1472)	2336 (0.1087)	4193 (0.2369)	4000 (0.2544)	3728 (3.0529)	1896 (0.0792)	**1580 (0.0754)**	1884 (0.0904)	4000 (0.1682)
FN8	2240 (0.1266)	2680 (0.1526)	4205 (0.2856)	4000 (0.2880)	2936 (2.6051)	1620 (0.0969)	**1440 (0.0884)**	1996 (0.1040)	2004 (0.1306)
FN9	2544 (0.1348)	2152 (0.0986)	4224 (0.2432)	4000 (0.2548)	**904** (1.2308)	1580 (0.0988)	1348 **(0.0851)**	1860 (0.1153)	4000 (0.2279)
FN10	2684 (0.1570)	2116 **(0.1089)**	4211 (0.2571)	4000 (0.2557)	3792 (2.0574)	2644 (0.1372)	**2012** (0.1209)	2244 (0.1159)	2708 (0.1540)

[a] NFE^n: Number of function evaluations, CPU-time^n: time taken by CPU per execution (average of 'n' executions).

the fluidized bed. The fluid bed consists of centrifugal fan to supply air flow from ambient air. There are two essential output parameters known as the moisture content and the product temperature in drying process [21]. In general, the drying process has got three phases. In the first phase is loaded with granulated material to be dried. Then drying temperature is increased to initiate constant drying phase. Third is reduced drying phase or called falling rate period. Finally dried material discharged from the dryer when the desired end dry matter was reached [22].

The main target of the drying process optimization is to improve the efficiency in the fluidized bed dryer in terms of energy consumption and quality loss. In order to do this, the production has to be performed minimum energy consumption and maximum quality together [23]. A multi-objective function can be described by total energy, product quality and moisture content. This objective function is given as Eq. (13),

$$J = \alpha u_a T_a (c_{p,a} + c_{p,wv} Y_a) + \beta(\bar{X} - X_d) + \gamma(Q_d - Q) \tag{13}$$

where α, β, γ denote the weighting factor in the objective function, u_a is the air flow rate (kg/s), T_a represents the air temperature (K), $c_{p,a}$ and $c_{p,w}$ represent air heat capacity and water vapor (J/kg K) respectively, Y_a is humidity of air (kg water vapor/kg dry air), $\bar{X}$ is average moisture content (kg water/kg dry solid), X_d is desired moisture content, Q represents product quality or the loss of product activity, Q_d is desired quality value (%100).

7. Results and discussion

In this section, the chaotic based DE algorithms were firstly run for different optimization problems to evaluate their performances. In addition to these simulations, for optimization of the baker's yeast drying process, the proposed chaotic based DE algorithms were compared with classic DE algorithm, PSO algorithm [19,30], ABC algorithm [16–18], SA algorithm [20], TACO algorithm [28].

7.1. Benchmark tests

All chaotic based DE algorithms were coded on PC with Intel(R) Core(TM) i5-3230 M CPU 2.60 GHz/8 GB RAM. The DE strategy used in these algorithms was selected as DE/rand/1/bin and the DE parameters were used as CR = 0.5, F = 0.8. For PSO algorithm, the learning factors (c_1,c_2) were selected as 2.05 and the evaporation coefficient for TACO algorithm was used as 0.1 according to the studies from literature. In all algorithms, population size was determined as 20. The termination criterion was determined as iteration or generation reaches the maximum number of iteration or generation and $|fitness(best) - fitness(worst)| = VTR$. VTR represents the value to reach and it was used as 1×10^{-6}. The maximum number of iteration was used as 200 for all optimization problems. All bound violation were repaired by holding at the boundary strategy given in Fig. 2. Table 2 summarizes the average results of 50 independent runs of the proposed chaotic based DE algorithms and DE, PSO, ABC, SA, TACO algorithms consecutively.

In Table 2, mean best indicates the average of minimum values obtained by the proposed chaotic DE algorithms and the other heuristic algorithms. This indicator represents with the standard deviation (std dev) to evaluate the performances of the algorithms. As can be seen from this table, Lorenz based DE algorithm (CDE_1) that reaches the best minimum values in 60% of all test functions. Rossler based DE algorithm (CDE_2) found the best global values in 40% of all the test functions. Chua based DE algorithm (CDE_3) has got the performance with the best minimum values in 20% of all test functions. Mackey-Glass based DE algorithm (CDE4) is

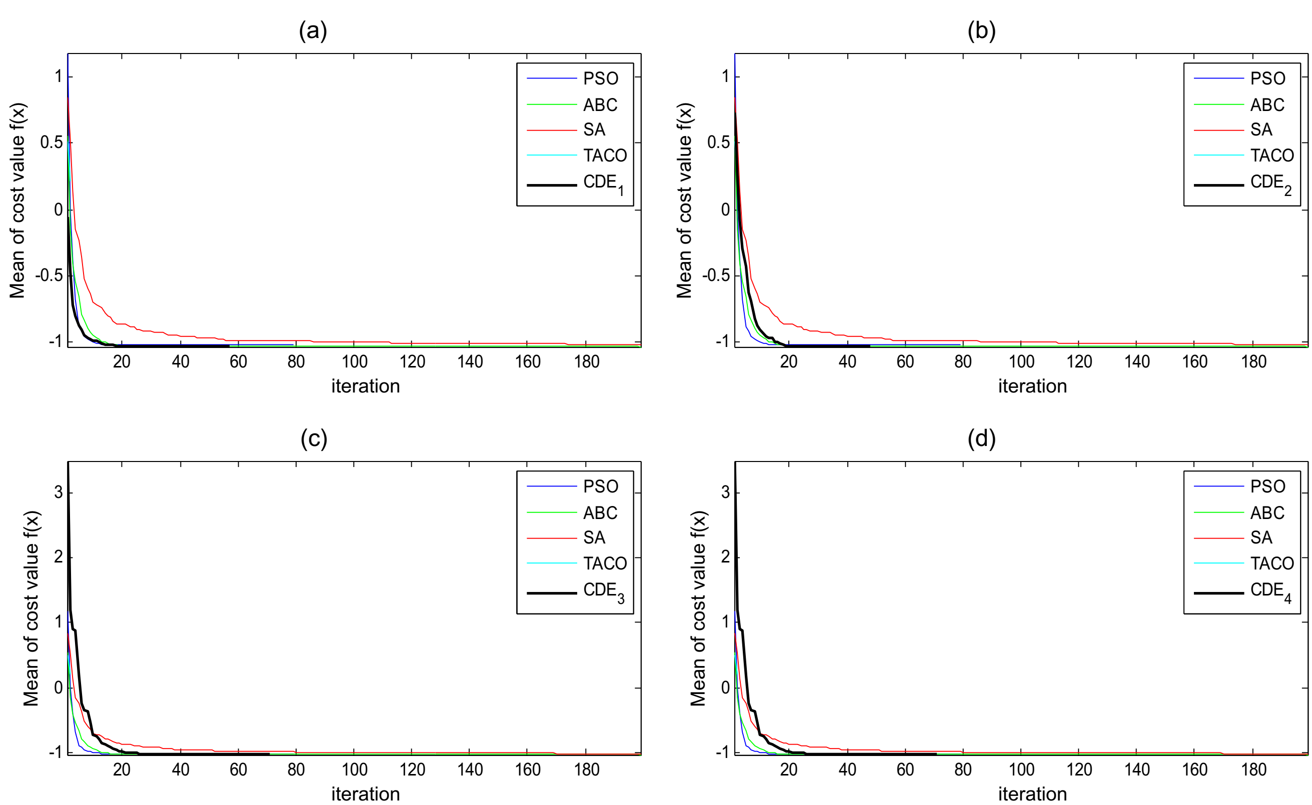

Fig. 3. Average best fitness curves of CDE Algorithms and PSO, ABC, SA and TACO algorithms for **test function FN9**. All experimental results are means of 50 independent runs. (a) Lorenz based DE Algorithm. (b) Rossler based DE Algorithm. (c) Chua based DE Algorithm. (d) Mackey-Glass based DE Algorithm.

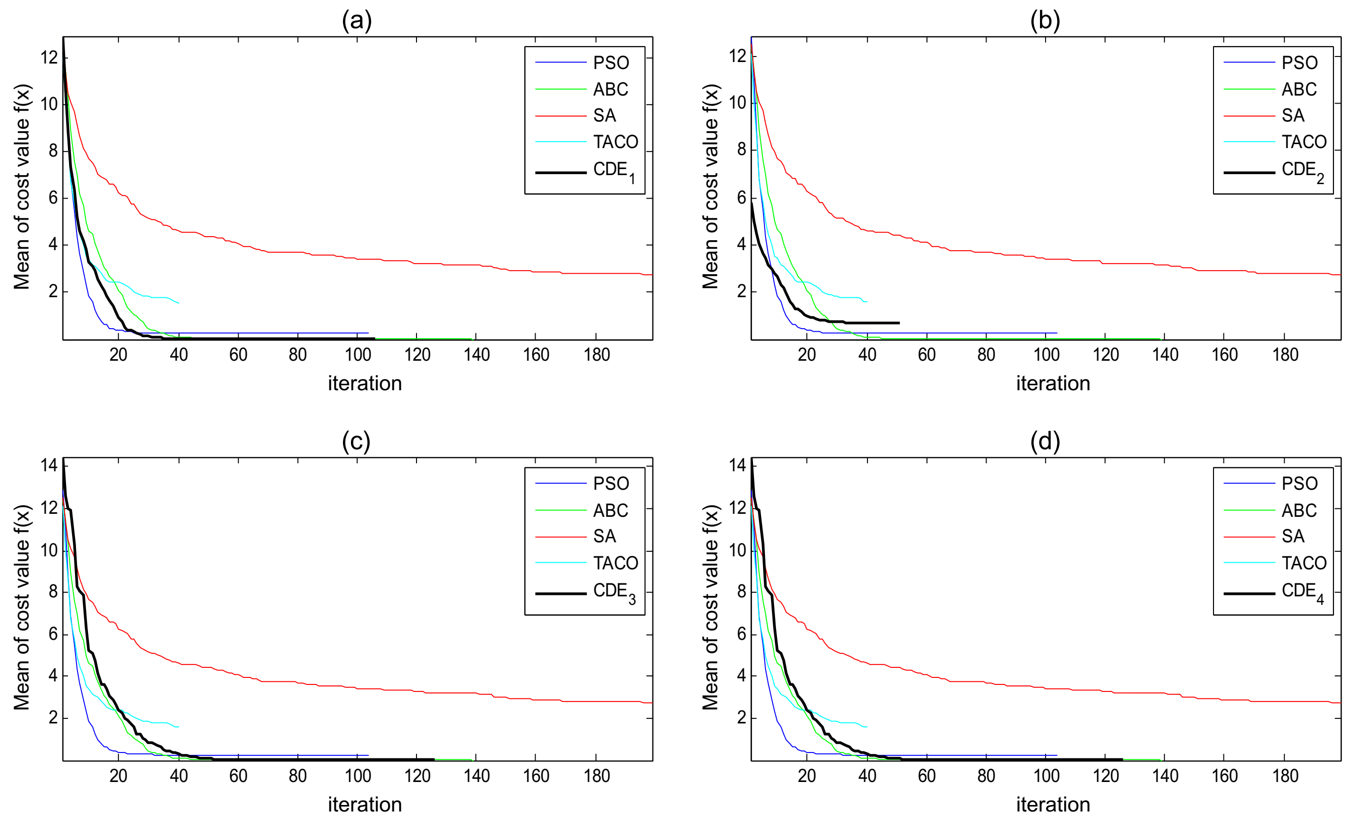

Fig. 4. Average best fitness curves of CDE Algorithms and PSO, ABC, SA and TACO algorithms for test function FN1. All experimental results are means of 50 independent runs. (a) Lorenz based DE Algorithm. (b) Rossler based DE Algorithm. (c) Chua based DE Algorithm. (d) Mackey-Glass based DE Algorithm.

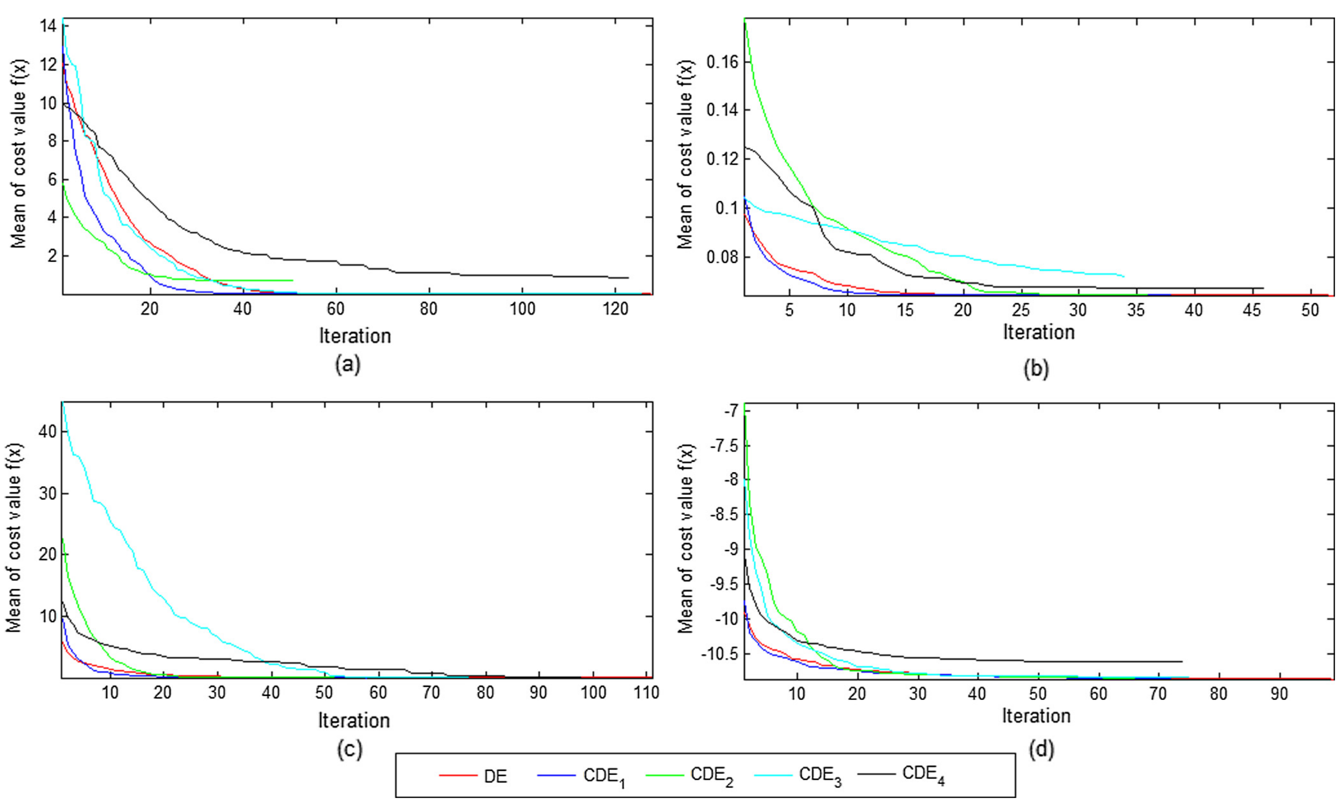

Fig. 5. Average best fitness curves of CDE Algorithms and Differential Evolution Algorithm for selected benchmark functions. All experimental results are means of 50 independent runs. (a) Test function FN1. (b) Test function FN5. (c) Test function FN7. (d) Test function FN10.

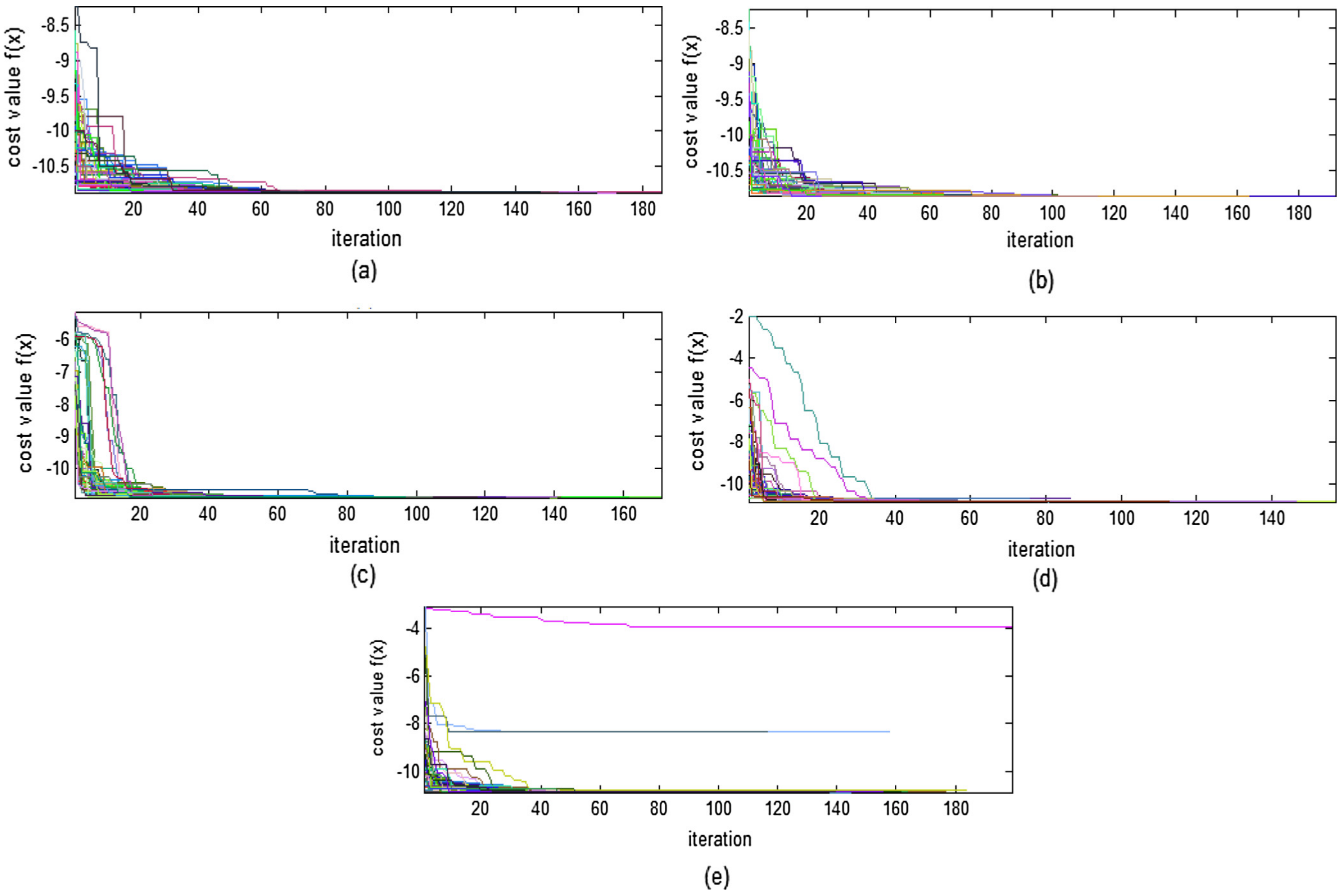

Fig. 6. The fitness curves for test function FN10. (a) DE Algorithm. (b) Lorenz based DE Algorithm. (c) Rossler based DE Algorithm. (d) Chua based DE Algorithm. (e) Mackey-Glass based DE Algorithm.

successful for only two functions. Finally, ABC algorithm is the best algorithm in terms of the mean value/standard deviation results. This table result shows that Lorenz based DE algorithm (CDE_1) and Rossler based DE algorithm (CDE_2) provide very competitive results.

In terms of the number of function evaluations (NFE) and CPU-time, Table 3 summarizes the results obtained by CDE algorithms and DE, PSO, ABC, SA, TACO algorithms. In Table 3, NFE indicator represents with the CPU-time (second) to compare the performances of all algorithms. Lorenz and Rossler based DE algorithms (CDE_1 and CDE_2) have the most minimum values among the NFE and CPU-time indicators. Both of CDE algorithms are faster than other heuristic algorithms. Chua based DE algorithm (CDE_3) has got the performance with the best NFE and CPU-time values in 50% of all test functions. The comparison shows that the chaotic based DE algorithms give better results than DE algorithm according to the CPU-time and NFE indicators. The CDE_2 algorithm has the best performance according to the NFE and CPU values.

Fig. 3 shows average best fitness curves for the chaotic based DE algorithms and PSO, ABC, SA and TACO algorithms for the test function FN9. For the function FN1, the average best fitness curves with 50 independent runs are plotted in Fig. 4 for each CDE algorithms with the other heuristic algorithms. As can be seen from these figures, it can be said that the proposed CDE algorithms are successful for the both test functions. Fig. 5 shows the average best fitness curves for the CDE algorithms and DE algorithm with 50 independent runs for the benchmark functions FN1, FN5, FN7, FN10. For the function FN10, the fitness curves with 50 independent runs are plotted in Fig. 6.

7.2. CEC 2014 benchmark tests

In the CEC 2014 tests, there are 30 numerical minimization problems. They consist of the four groups: unimodal functions (F1–F3), simple multimodal functions (F4–F16), hybrid function (F17–F22) and composition functions (F23–F30). The detailed information about the CEC 2014 test functions can be found in [51].

Table 4 gives the experimental results of DE algorithm and proposed CDE algorithms for 10D CEC 2014 all test functions. This table presents the best, worst, mean, median and standard variance values of function error values for the 51 runs. In Fig. 7, the boxplots are shown for mean values of function error of the proposed CDE algorithms and DE algorithm on 10D. According to this figure, especially CDE_1 and CDE_2 algorithms present competitive results for all test functions. The performance of the other chaotic algorithms (CDE_3 and CDE_4) are worse than those of DE, CDE_1 and CDE_2 algorithms.

We used a non-parametric Wilcoxon ranksum test to determine if all sets of solutions are different statistically significant or not. This statistical test returns a value that is called *p*-value. In this study, CDE algorithms and DE algorithm are statistically tested. Table 5 shows the *p*-values of the Wilcoxon ranksum test over 10D CEC 2014 functions. If the algorithm's p-value is less than

Table 4
Experimental results of DE algorithm and proposed Chaotic based DE algorithms in for 10D CEC 2014 test functions.

	DE					CDE_1					CDE_2					CDE_3					CDE_4				
Fn	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std	Best	Worst	Median	Mean	Std
1	440.5	2804.1	1543.1	1530.8	474.8	0.0	9934.8	119.5	1285.0	2447.2	0.0	4412.3	545.7	948.2	1193.6	30649.5	245376.2	77282.7	83482.1	38742.0	57707.8	1532580.2	294622.3	342183.3	243587.2
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	25.8	365211.5	439.0	15266.0	58769.7
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.8	3601.3	746.9	990.6	796.5	0.0	1889.7	79.4	226.4	330.0
4	0.0	34.8	0.0	1.0	4.9	0.0	34.8	0.0	2.2	8.3	0.0	34.8	0.0	0.7	4.9	0.0	34.8	0.4	5.0	11.1	0.2	56.2	0.6	3.3	8.5
5	17.5	20.2	20.1	20.0	0.5	19.5	20.3	20.3	20.2	0.1	0.0	20.4	20.3	18.9	4.9	19.4	20.3	20.1	20.1	0.1	18.2	20.2	20.1	20.0	0.3
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.2	0.0	0.3	0.5	0.0	3.8	0.2	0.7	1.0
7	0.1	0.3	0.2	0.2	0.0	0.1	0.5	0.4	0.4	0.1	0.0	0.6	0.4	0.4	0.1	0.2	0.4	0.3	0.3	0.0	0.1	0.9	0.2	0.2	0.1
8	0.0	0.0	0.0	0.0	0.0	0.0	6.5	0.0	0.9	1.8	0.0	34.8	5.8	7.4	7.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	7.3	13.3	10.7	10.5	1.7	10.6	25.9	20.0	19.8	3.0	4.0	35.7	19.8	19.4	7.3	11.6	25.5	19.0	18.6	3.1	6.6	25.2	12.4	12.9	3.6
10	0.0	0.1	0.0	0.0	0.0	44.9	236.7	115.2	116.2	44.6	0.2	770.6	141.3	172.1	153.8	0.0	7.5	0.1	0.6	1.3	0.0	0.2	0.0	0.0	0.0
11	279.9	645.3	457.7	469.0	77.4	569.0	1021.6	800.9	809.6	114.0	10.4	1201.1	823.2	748.0	284.7	218.9	670.1	460.4	463.5	112.7	290.2	827.0	624.9	621.1	128.1
12	0.2	0.5	0.4	0.4	0.1	0.3	1.0	0.7	0.7	0.1	0.0	1.3	0.9	0.8	0.3	0.5	1.0	0.8	0.8	0.1	0.2	0.5	0.3	0.4	0.1
13	0.1	0.2	0.2	0.1	0.0	0.1	0.2	0.2	0.2	0.0	0.0	0.3	0.2	0.2	0.1	0.1	0.2	0.2	0.2	0.0	0.1	0.4	0.2	0.2	0.0
14	0.0	0.1	0.1	0.1	0.0	0.1	0.2	0.1	0.1	0.0	0.0	0.3	0.1	0.1	0.0	0.1	0.2	0.1	0.1	0.0	0.0	0.3	0.1	0.1	0.0
15	0.9	1.8	1.3	1.4	0.2	1.4	2.5	2.0	2.0	0.3	0.4	3.2	2.1	2.0	0.6	1.2	2.3	1.8	1.8	0.3	0.9	5.3	1.5	1.7	0.8
16	1.6	2.6	2.2	2.2	0.2	1.9	2.9	2.5	2.5	0.2	0.9	2.8	2.5	2.4	0.4	1.7	2.7	2.4	2.4	0.2	1.6	2.7	2.3	2.3	0.2
17	10.9	31.2	19.7	19.9	4.8	2.7	65.4	14.2	16.5	12.3	0.0	26.6	3.8	5.4	6.3	491.4	4874.1	1164.0	1292.4	659.2	416.4	16065.1	2168.1	3084.1	2897.9
18	0.6	1.9	1.2	1.2	0.4	0.1	4.9	1.5	1.9	1.5	0.0	4.8	0.6	1.6	1.7	2.0	8.4	4.1	4.3	1.1	4.7	342.7	40.7	62.1	67.2
19	0.1	0.4	0.2	0.2	0.1	0.3	1.0	0.6	0.6	0.2	0.0	1.0	0.6	0.5	0.3	0.6	1.9	1.7	1.6	0.2	0.7	1.6	1.1	1.2	0.2
20	0.0	0.1	0.0	0.0	0.0	0.0	0.6	0.0	0.1	0.1	0.0	0.6	0.0	0.1	0.1	0.8	64.4	14.9	17.2	16.1	1.8	33.6	6.9	9.8	7.4
21	0.0	0.4	0.1	0.1	0.1	0.0	1.2	0.2	0.3	0.2	0.0	0.7	0.1	0.1	0.2	1.3	62.3	2.4	4.8	9.0	23.6	1955.7	164.3	265.7	342.1
22	0.0	0.3	0.0	0.0	0.0	0.0	1.3	0.2	0.3	0.3	0.0	1.2	0.1	0.2	0.3	0.1	1.0	0.4	0.4	0.2	0.2	15.2	2.5	3.8	3.9
23	329.5	329.5	329.5	329.5	0.0	329.5	329.5	329.5	329.5	0.0	200.0	329.5	200.0	248.2	63.2	218.8	329.5	329.5	317.8	30.1	246.1	330.9	329.5	325.7	16.1
24	113.9	123.4	119.0	118.8	2.1	120.6	134.7	127.4	127.7	3.1	110.6	200.5	127.3	132.2	20.9	115.6	128.7	122.4	122.2	2.7	114.2	131.8	122.4	122.7	3.8
25	122.7	201.4	132.5	137.1	17.8	114.7	201.2	132.1	151.5	32.1	100.0	199.6	128.3	125.7	16.6	127.8	147.1	135.9	136.5	4.3	122.2	157.2	142.2	143.0	5.7
26	100.1	100.2	100.2	100.2	0.0	100.1	100.2	100.2	100.2	0.0	100.0	100.3	100.2	100.2	0.0	100.1	100.2	100.2	100.2	0.0	100.1	100.3	100.2	100.2	0.0
27	1.5	300.0	2.2	43.0	103.5	1.4	400.1	2.4	78.2	139.8	1.2	400.1	2.5	57.0	119.6	1.7	338.9	300.3	177.4	145.0	2.8	400.8	8.1	108.3	154.3
28	356.8	362.2	356.8	356.9	0.8	356.8	455.7	356.8	372.3	36.3	100.0	478.0	455.7	398.5	116.2	114.6	390.0	356.9	347.8	47.0	358.5	452.2	376.5	382.1	18.3
29	228.3	239.7	232.3	233.0	2.4	100.1	246.9	223.3	207.2	41.8	100.0	223.4	221.8	187.7	52.4	343.2	775.1	512.6	506.5	101.1	263.9	543.1	382.4	389.1	65.0
30	476.2	527.5	497.7	498.3	8.8	463.3	508.0	469.4	473.4	10.7	462.3	546.3	466.4	479.2	21.9	630.4	1123.1	760.6	778.8	121.3	525.6	955.3	624.0	658.4	100.3

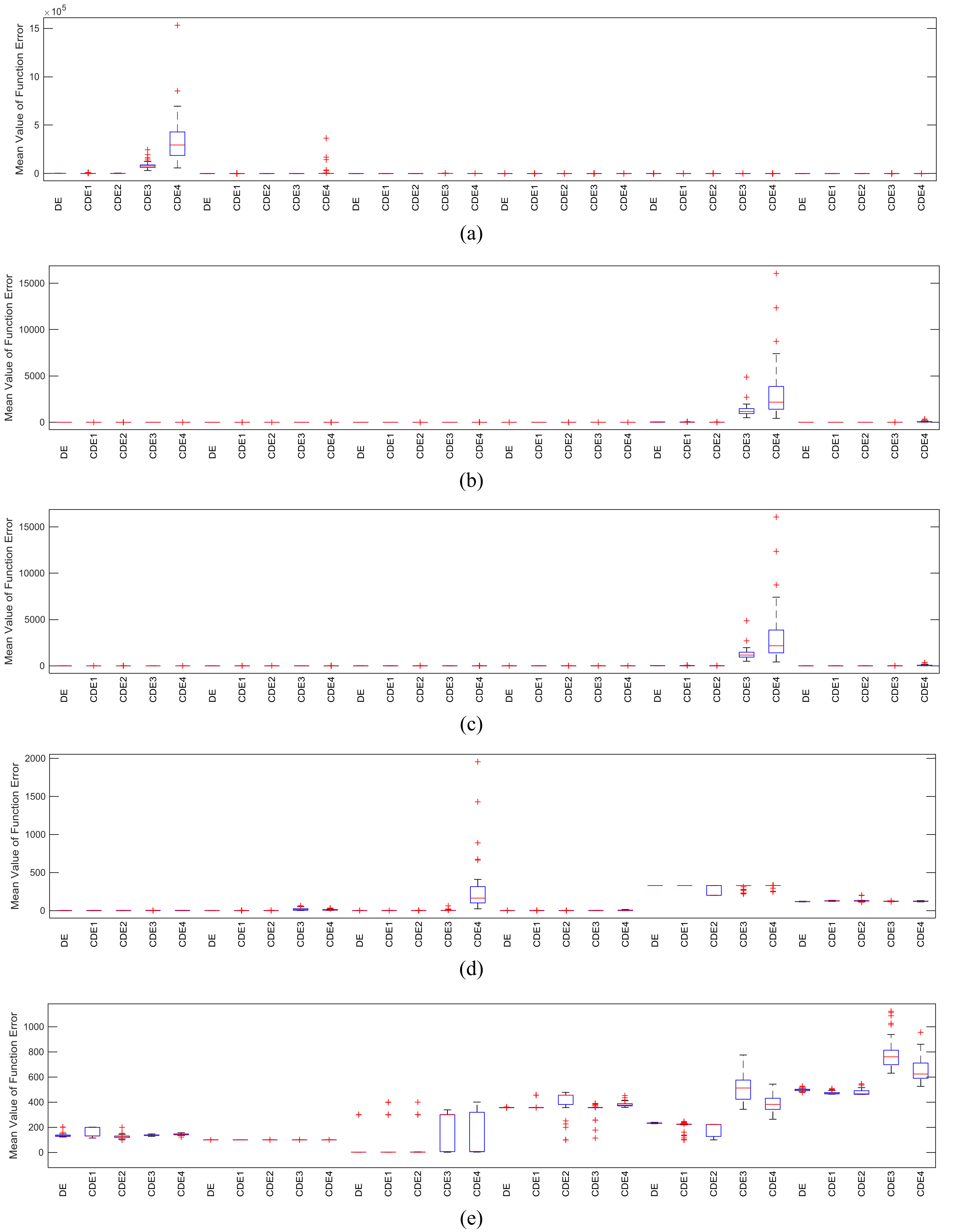

Fig. 7. Boxplot of comparative convergence for all CEC 2014 test functions (a) Func. No. 1–6, (b) Func. No. 7–12, (c) Func. No. 13–18, (d) Func. No. 19–24, (e) Func. No. 25–30.

Table 5
p-Values of the Wilcoxon ranksum test over 10D CEC 2014 functions.

FN	DE	CDE_1	CDE_2	CDE_3	CDE_4
1	5.145E-10	5.145E-10	4.004E-10	5.145E-10	5.145E-10
2	9.237E-13	2.317E-11	9.237E-13	9.237E-13	5.145E-10
3	9.237E-13	9.237E-13	9.237E-13	5.145E-10	5.145E-10
4	5.141E-10	5.140E-10	2.877E-10	5.134E-10	5.145E-10
5	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
6	9.237E-13	9.237E-13	9.237E-13	4.659E-10	5.145E-10
7	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
8	9.237E-13	3.475E-10	5.139E-10	9.237E-13	9.237E-13
9	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
10	2.244E-11	5.145E-10	5.145E-10	5.039E-10	2.321E-11
11	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
12	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
13	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
14	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
15	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
16	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
17	5.145E-10	5.145E-10	4.963E-10	5.145E-10	5.145E-10
18	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
19	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
20	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
21	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
22	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
23	9.237E-13	9.237E-13	3.756E-10	1.783E-11	4.747E-10
24	5.145E-10	5.145E-10	5.144E-10	5.145E-10	5.145E-10
25	5.145E-10	5.134E-10	5.141E-10	5.145E-10	5.145E-10
26	5.140E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10
27	5.145E-10	5.082E-10	5.052E-10	5.145E-10	5.145E-10
28	5.144E-10	5.072E-10	4.433E-10	5.145E-10	5.145E-10
29	5.145E-10	5.145E-10	5.111E-10	5.145E-10	5.145E-10
30	5.145E-10	5.145E-10	5.145E-10	5.145E-10	5.145E-10

0.05 then it is statistically significant. The statistical tests show that the results are statistically significant for all CDE algorithms.

7.3. Optimization of baker's yeast drying process

The optimization problem of baker's yeast drying process is given below

$$\min_{T_a, Y_a} J(T_a, Y_a) \tag{14}$$

$$293\ \text{K} \leqslant T_a \leqslant 373\ \text{K},\ 0 \leqslant Y_a \leqslant 5.10^{-3}\ \text{kg}\ \textit{water vapor}/\ \text{kg}\ \textit{dry air} \tag{15}$$

The air temperature (T_a) and the humidity of air (Y_a) are the manipulated variables regarding optimization process. The parameters of drying process of the baker's yeast were determined as initial moisture content equals 1.563 kg water/kg dry solid, the air flow rate equals 12000 kg air h^{-1} for cylindrical granule. Fig. 8 shows the optimization results obtained by chaotic based DE algorithms and DE algorithm. The trends of the moisture content, temperature and product quality during drying process can be seen from these figures.

According to the final moisture content value at the end of the drying process, all chaotic based DE algorithms have got the same performances approximately, but DE algorithm is better than the proposed chaotic based DE algorithms. The shortest drying time (25 min) belongs to the Mackey-Glass based DE algorithm and all chaotic based DE algorithms have better drying time than classic

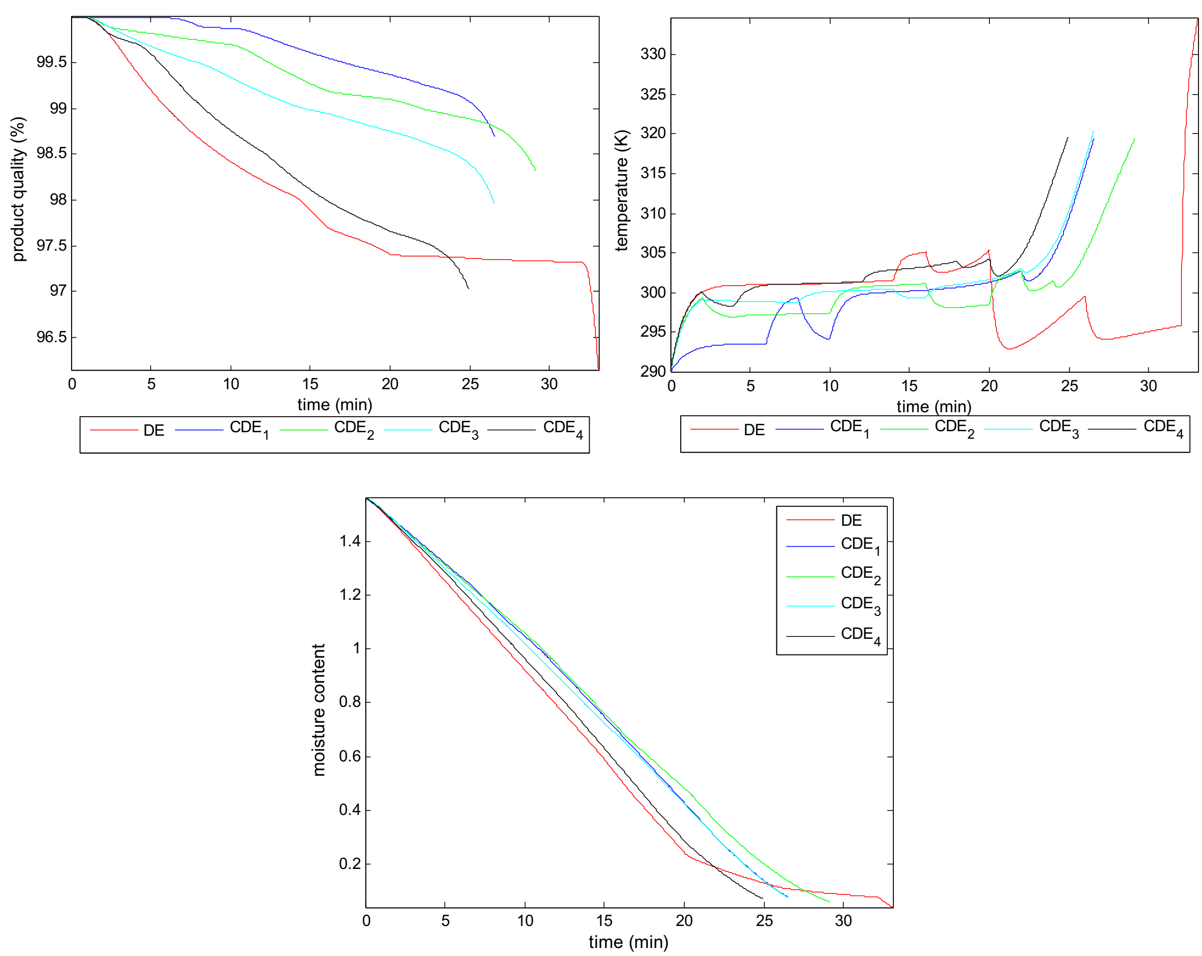

Fig. 8. The moisture content, temperature and product quality during optimization process by chaotic based DE algorithms. CDE_1: Lorenz based DE algorithm, CDE_2: Rossler based DE algorithm, CDE_3: Chua based DE algorithm, CDE_4: Mackey-Glass based DE algorithm.

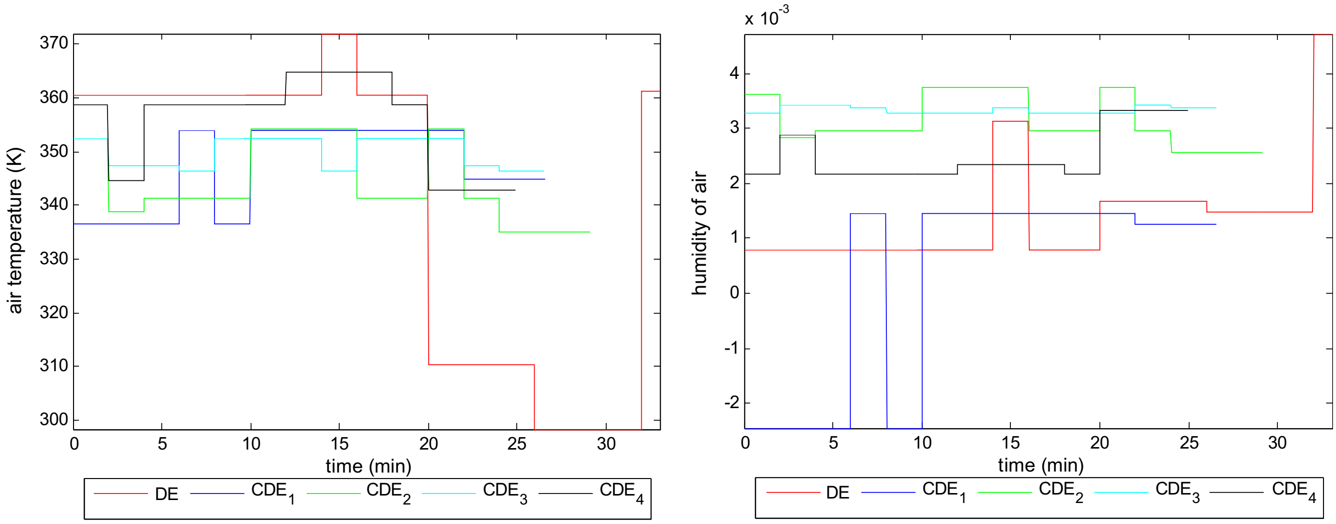

Fig. 9. The optimization results (air temperature and humidity of air) solved by Chaotic based DE algorithms. CDE_1: Lorenz based DE algorithm, CDE_2: Rossler based DE algorithm, CDE_3: Chua based DE algorithm, CDE_4: Mackey-Glass based DE algorithm.

Table 6
Optimization results of chaotic based differential evolution algorithms.

	DE	CDE_1	CDE_2	CDE_3	CDE_4
Q (%)	96.133	98.685	98.320	97.964	97.037
X_f (kg/kg)	0.04148	0.07820	0.06192	0.077676	0.074227
J_a (kJ)	2.353.000	1.965.700	2.098.900	1.991.900	1.877.100
t (sec)	1983	1594	1747	1592	1495

DE algorithm. The product quality is the important parameter in the biomass drying process especially. The result obtained by Lorenz based DE algorithm is the best final value in terms of the product quality. As can be seen from the temperature profiles found by all chaotic based DE and classic DE algorithms, the fluidized bed temperature value increases at the beginning of the process, then it follows to the fixed stable value, it has rising trend at the end of the process. The profiles of air temperature (T_a) and the humidity of air (Y_a) obtained by the proposed algorithms are shown in Fig. 9. The product quality (Q), the energy cost of air (J_a), the final moisture content (X_f) and the total drying period (t) at the end of the drying process are given in Table 6. The energy cost of air was given as $J_a = \alpha u_a T_a (c_{p,a} + c_{p,wv} Y_a)$ in Eq. (13). According to the product quality, the best value was observed by Lorenz based DE algorithm (CDE_1) as%98.685. All of the chaotic based DE algorithms have better drying time than the time of classic DE algorithm.

As can be seen from the energy cost values, Mackey-Glass based DE algorithm (CDE_4) has got the best minimum cost at the end of the process and the performance of the classic DE algorithm is worse than the performances of the chaotic based DE algorithms. The baker yeast, the microorganism *Saccharomyces cerevisiae* was used for experimental data of the drying process in this study. In the experimental data for cylindrical granules, the total drying time was measured as 27 min without loading period, the product quality was obtained as%89.6, the moisture content was measured as 0.069 kg/kg and the energy cost of air in drying process was calculated as 1.944.500 kJ respectively. The product quality value at the end of the drying process was held on the higher level than the value taken from experimental data. In this way, the dry biomass product that has more quality has been obtained by both DE algorithm and chaotic based DE algorithms. Besides, total drying time has been decreased by the chaotic based DE algorithms without CDE_2. It is clear that no algorithm's perform shows superiorly than the experimental data in terms of moisture content and energy cost of air.

8. Conclusion

In this paper, Chaotic based Differential Evolution (CDE) algorithm has been introduced and compared to classic Differential Evolution (DE) for optimization of benchmark test functions and optimization of baker's yeast drying process. The chaotic based structure were proposed to generate the individuals in the population instead of the random number generator. CDE includes four different chaotic systems such as Lorenz, Rossler, Chua and Mackey-Glass functions. The proposed CDE algorithms has been implemented and tested on benchmark optimization problems taken from the literature. The popular heuristic algorithms (DE, PSO, ABC, SA and TACO) have been used for the performance evaluation works with the proposed CDE algorithms. The comparison results with 50 independent runs show that the performances of the proposed CDE_1 and CDE_2 algorithms are better than the other heuristic algorithms in terms of the mean best value and standard deviation. According to the CEC 2014 test results, the CDE_1 and CDE_2 algorithms provide the competitive results.

In this paper, applying DE and CDE algorithms to the optimization of baker's yeast drying process was focused. In biomass drying process, there are four important parameters, such as product quality, drying total time, energy cost of used hot air and final moisture content. In comparison with the data taken from a baker's yeast producer in Turkey, especially the improvement on the product quality has been provided by CDE algorithms. Besides, it is obvious that the results of CDE algorithms were better than the results of DE algorithm in terms of the process output values except for only moisture content. In the future works, the proposed CDE algorithms will be implemented for different processes and their performances will be evaluated and compared with the other heuristic methods.

References

[1] Asafuddoula M, Ray T, Sarker R. An adaptive hybrid differential evolution algorithm for single objective optimization. Appl Math Comput 2014;231:601–18.
[2] Babu B, Angira R. Modified differential evolution (MDE) for optimization of non-linear chemical processes. Comput Chem Eng 2006;30:989–1002.

[3] Babu B, Munawar S. Differential evolution strategies for optimal design of shell-and-tube heat exchangers. Chem Eng Sci 2007;62:3720–39.
[4] Brest J, Greiner S, Boskovic B, Mernik M, Zumer V. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans Evol Comput 2006;10(6):646–57.
[5] Chen N, Chen W-N, Zhang J. Fast detection of human using differential evolution. Signal Process 2015;110:155–63.
[6] Chen X, Du W, Qian F. Multi-objective differential evolution with ranking-based mutation operator and its application in chemical process optimization. Chemomet Intelligent Lab Syst 2014;136:85–96.
[7] Chua LO. The Genesis of Chua's Circuit. California: Electronics Research Laboratory, College of Engineering, University of California; 1992.
[8] Chua LO, Itoh M, Kocarev L, Eckert K. Chaos synchronization in Chua's circuit. J Circuits, Syst, Comput 1993;3(1):93–108.
[9] Coelho Ld, Ayala HV, Mariani VC. A self-adaptive chaotic differential evolution algorithm using gamma distribution for unconstrained global optimization. Appl Math Comput 2014;234:452–9.
[10] Draa A, Bouzoubia S, Boukhalfa I. A sinusoidal differential evolution algorithm for numerical optimisation. Appl Soft Comput 2015;27:99–126.
[11] Eser M, Yüzgeç U. Chaotic based differential evolution algorithm. 2nd International Symposium on Innovative Technologies in Engineering and Science (pp. 201–210) (2014b) Karabük: ISITES 2014.
[12] Eser M, Yüzgeç, U. Comparison of Lorenz and Rossler based differential evolution algorithms. Electrical, Electronics and Computer Engineering Symposium (pp. 713–717). Bursa: ELECO 2014.
[13] Gong W, Fialho Á, Cai Z, Li H. Adaptive strategy selection in differential evolution for numerical optimization: an empirical study. Inf Sci 2011;181:5364–86.
[14] Guégan D. Chaos in economics and finance. Annual Rev Control 2009;33 (1):89–93.
[15] Jamil M, Yang XS. A literature survey of benchmark functions for global optimisation problems. Internat J Mathemat Modell Numer Opt 2013;4 (2):150–94.
[16] Karaboga D, Akay B. A comparative study of Artificial Bee Colony algorithm. Appl Math Comput 2009;214(1):108–32.
[17] Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput J 2008;8(1):687–97.
[18] Karaboga D, Ozturk C. A novel clustering approach: Artificial Bee Colony (ABC) algorithm. Appl Soft Comput J 2011;11(1):652–7.
[19] Kennedy J, Eberhart R. Particle swarm optimization. Neural Networks. Proceedings IEEE International Conference on 1995: pp. 1942–1948.
[20] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. Sci 1983;220(4598):671–80.
[21] Köni M, Türker M, Yüzgeç U, Dinçer H, Kapucu H. Adaptive modeling of baker's yeast drying in batch fluidized bed. Control Eng Pract 2009;17:503–17.
[22] Köni M, Yüzgeç U, Türker M, Dinçer H. Adaptive fuzzy control of baker's yeast drying in batch fluidized bed. Drying Technol 2010;28:205–13.
[23] Köni M, Yüzgeç U, Türker M, Dinçer H. Optimal quality control of baker's yeast drying in large scale batch fluidized bed. Chem Eng Proc 2009;48:1361–70.
[24] Lorenz EN. Deterministic Nonperiodic Flow. J Atmos Sci 1963;20:130–41.
[25] Mackey M, Glass L. Oscilliation and Chaos in phyiological control systems. Sci New Series 1977;197:287–9.
[26] Mohamed AW, Sabry HZ, Khorshid M. An alternative differential evolution algorithm for global optimization. J Adv Res 2012;3:149–65.
[27] Novo J, Santos J, Penedo M. Multiobjective differential evolution in the optimization of topological active models. Appl Soft Comput 2013;13:3167–77.
[28] Özbakir L, Baykasoğlu A, Kulluk S, Yapici H. TACO-miner: an ant colony based algorithm for rule extraction from trained neural networks. Expert Syst Appl 2009;36(10):12295–305.
[29] Poikolainen I, Neri F, Caraffini F. Cluster-based population initialization for differential evolution frameworks. Inf Sci 2015;297:216–35.
[30] Poli R, Kennedy J, Blackwell T. Particle swarm optimization. Swarm Intell 2007;1(1):33–57.
[31] Price K. An introduction to differential evolution. In: Corne D, Dorigo M, Glover F, editors. New Ideas Opt. London: McGraw-Hill; 1999. p. 79–108.
[32] Price K, Storn R. Differential evolution: a simple evolution strategy for fast optimization. Dr Dobb's J Software Tools 1997;22(4):18–24.
[33] Qi Y, Hou Z, Yin M, Sun H, Huang J. An immune multi-objective optimization algorithm with differential evolution inspired recombination. Appl Soft Comp 2015;29:395–410.
[34] Rahnamayan S, Tizhoosh H, Salama M. Opposition-based differential evolution. IEEE Trans Evol Comput 2008;12(1):64–79.
[35] Rahnamayan S, Tizhoosh H, Salama M. Opposition-based differential evolution. IEEE Transact Evolut Comput 2008;12(1):64–79.
[36] Rössler O. An equation for continuous chaos. Phys Lett A 1976;57:397–8.
[37] Sayah S, Hamouda A. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. Appl Soft Comput 2013;13:1608–19.
[38] Shen CY, Evans TE, Finette S. Polynomial chaos quantification of the growth of uncertainty investigated with a Lorenz model. J Atmos Oceanic Technol 2010;27(6):1059–71.
[39] Shengrui Z, Baisheng N, Shuiwen L, Hui W. Model of gas concentration forecast based on chaos theory. Procedia Eng 2011;26:211–7.
[40] Sivakumar B. Chaos theory in hydrology: important issues and interpretations. J Hydrol 2000.
[41] Storn R, Price K. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11:341–59.
[42] Ugolotti R, Nashed YS, Mesejo P, Ivekovic S, Mussi L, Cagnoni S. Particle Swarm optimization and differential evolution for model-based object detection. Appl Soft Comput 2013;13:3092–105.
[43] Wu CW, Chua LO. A simple way to synchronize chaotic systems with applications to secure communication systems. Int J Bifurcation Chaos 1993;3(6):1619–27.
[44] Yüzgeç U. Performance comparison of differential evolution techniques on optimization of feeding profile for an industrial scale fed-batch baker's yeast fermentation process. ISA Trans 2010;49:167–76.
[45] Yüzgeç U, Eser M. Rossler based chaotic differential evolution algorithm. Bilecik Seyh Edebali University Science Journal 2014;1(2):9–15.
[46] Yüzgeç U, Becerikli Y, Türker M. Dynamic neural network based model predictive control of an industrial baker's yeast drying process. IEEE Trans Neural Net 2008;19(7):1231–42.
[47] Yüzgeç U, Becerikli Y, Türker M. Nonlinear predictive control of a drying process using genetic algorithms. ISA Trans 2006;45:589–602.
[48] Zhang G, Cheng J, Gheorghe M, Meng Q. A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems. Appl Soft Comput 2013;13:1528–42.
[49] Zhang J, Lin S, Qiu W. A modified chaotic differential evolution algorithm for short-term optimal hydrothermal scheduling. Elect Power Energy Sys 2015;65:159–68.
[50] Zhang X, Yuen SY. A directional mutation operator for differential evolution algorithms. Appl Soft Comput 2015;30:529–48.
[51] Liang JJ, Qu BY, Suganthan PN. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real parameter numerical optimization. Technical Report 201311; 2013.
[52] Mohamed AW. A novel differential evolution algorithm for solving constrained engineering optimization problems. J Intell Manuf 2017:1–34.
[53] Mohamed AW. Solving large-scale global optimization problems using enhanced adaptive differential evolution algorithm. Complex Intelligent Syst 2017;3:205–31.
[54] Mohamed AW, Suganthan PN. Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation. Soft Comput 2017:1–21.
[55] Mohamed AW, Mohamed AK. Adaptive guided differential evolution algorithm with novel mutation for numerical optimization. Int J Mach Learn Cybern 2017:1–25.