Applied Soft Computing 71 (2018) 1161–1175

Contents lists available at ScienceDirect

Applied Soft Computing

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

Chaotic crow search algorithm for fractional optimization problems

Rizk M. Rizk-Allah[a,1], Aboul Ella Hassanien[b,1], Siddhartha Bhattacharyya[c,*]

[a] Department of Basic Engineering Science, Faculty of Engineering, Menoufia University, Shebin El-Kom, Egypt
[b] Faculty of Computers and Information, Cairo University, Egypt
[c] Department of Computer Application, RCC Institute of Information Technology, Kolkata, India

ARTICLE INFO

Article history:
Received 1 September 2017
Received in revised form 14 March 2018
Accepted 15 March 2018
Available online 22 March 2018

Keywords:
Crow search algorithm
Chaos
Fractional programming
Economic environmental power dispatch

ABSTRACT

This paper presents a chaotic crow search algorithm (CCSA) for solving fractional optimization problems (FOPs). To refine the global convergence speed and enhance the exploration/exploitation tendencies, the proposed CCSA integrates chaos theory (CT) into the CSA. CT is introduced to tune the parameters of the standard CSA, yielding four variants, with the best chaotic variant being investigated. The performance of the proposed CCSA is validated on twenty well-known fractional benchmark problems. Moreover, it is validated on a fractional economic environmental power dispatch problem by attempting to minimize the ratio of total emissions to total fuel cost. Finally, the proposed CCSA is compared with the standard CSA, particle swarm optimization (PSO), firefly algorithm (FFA), dragonfly algorithm (DA) and grey wolf algorithm (GWA). Additionally, the efficiency of the proposed CCSA is justified using the non parametric Wilcoxon signed-rank test. The experimental results prove that the proposed CCSA outperforms other algorithms in terms of quality and reliability.

1. Introduction

The Fractional programming (FP) problem is a very important topic in operations research. An FP problem is a specific type of optimization programming problem that aims to optimize one or various ratios of the objective functions [1]. It has attracted the attention of numerous researchers due to its usage in several fields, such as economics, engineering, environmental science, finance, business and administration [1]. FP problems emerge naturally in decision-making models that require some degree of efficiency, such as output/input, income/investment, performance/cost and return/risk [1].

Conventional methods for solving the FP problem attempt to find the solutions using transformation-based algorithms [2–5]. Isbell et al. [2] solved the FP problem by transforming it into a sequence of linear programs (LPs). Charnes-Copper [3] transformed the linear FP (LFP) problem into an equivalent form of the LP problem by incorporating one extra variable and one extra constraint, with the optimum solution being computed using the simplex algorithm. Pandey et al. [4] solved the linear FP problem using different strategy based on the simplex method. Mojtaba Borza et al. [5] proposed a novel technique for solving the linear FP problem: the coefficients of the objective function are expressed in interval form. Additionally, some other works on FP theory have been presented in the literature [1].

Traditional techniques for solving FP problems have become more tedious and time-consuming, especially for high-dimensional problems, due to their reliance on some transformations and approximations. Furthermore, most of the transformations are manually conducted. Thus, errors may occur. Consequently, to overcome these limitations, metaheuristic algorithms (MHAs) have been developed to address these complex optimization tasks. MHAs have been widely adopted to solve several real-world problems in different disciplines [6,7] due to their flexibility, simplicity, and adaptability. They also have robust performances and are reliable in obtaining global solutions.

Accordingly, metaheuristic optimization algorithms have also been extended to solve FP problems [8–12]. Sameeullah et al. [8] proposed using the genetic algorithm to address linear FP problems. In their method, a set of feasible solutions is obtained using random numbers, and then the fitness of these solutions is determined. Moreover, the best feasible solution is detected, which then replaces the worst one. The crossover operator is adopted for pair wise solutions to obtain new solutions. These procedures are carried out until the maximum iteration is satisfied, and then the best solution is recorded. Unfortunately, this approach was designed to determine the weights and aspiration for solving specific problems

* Corresponding author.
E-mail address: dr.siddhartha.bhattacharyya@gmail.com (S. Bhattacharyya).
[1] Scientific Research Group in Egypt http://www.egyptscience.net.

https://doi.org/10.1016/j.asoc.2018.03.019

in which the functions of the numerators and denominators are expressed as affine functions. Additionally, comparisons showcasing the superior efficiency of this approach are lacking. In 2009, Calvete et al. [9] presented a genetic algorithm for solving bi-level problems where, in each level, the objective function is formulated as a linear fractional and the feasible region is a polyhedron. Their algorithm searches among extremes of a polyhedron. Moreover, the crossover operator and mutation operator are considered. Unfortunately, this algorithm addresses specific cases by considering the linearity of the numerator and denominator in each level. Furthermore, it has not been applied in practice. Jaberipour et al. [10] calculated the sum of ratio terms using the harmony search algorithm, in which the parameters are adjusted by using the central difference formula to enhance the convergence rate of the algorithm. Unfortunately, their algorithm requires a considerable number of iterations and did not yield the global optimal solution for the Himmelblau function. It also fails to consider any statistical tests and highly depends on the chosen parameters. Abdel-Baset et al. [11] presented an improved flower pollination algorithm for solving FP problems that is based on chaos theory (CT). Unfortunately, this approach was investigated on only a few test functions, and an exhibition of its convergence behavior and some statistical tests are lacking. Raouf et al. [12] proposed solving FP problems using swarm intelligence. They introduced two kinds of swarms, i.e., PSO and FFA. Unfortunately, they did not conduct statistical tests to validate the stability of the obtained results and to exhibit the convergence behavior of their algorithms.

Recently, the economic environmental power dispatch (EEPD) has been formulated mathematically as a fractional optimization problem (FOP) [13]. This formulation is becoming more and more significant as the demand for electric power increases at an alarming rate, especially in developing countries. The main objective of the EEPD is to schedule generation unit outputs such that the total fuel cost and total pollutant emissions are minimized simultaneously while operational constraints are satisfied.

Many researchers have attempted to solve the EEPD problem [14]. In [15], the problem was transformed into an equivalent single objective via the goal programming approach. Unfortunately, this approach requires additional time to construct models to determine the weights and aspiration levels. Therefore, the goal programming approach cannot depict any information regarding the trade-off between two objectives, namely, fuel cost and emissions. In another research direction, weighted sum methods have been developed to achieve the compromise solutions [16]. In these methods, each objective is associated with weighting values determined by the decision maker (DM); therefore, the solutions for the EEPD greatly depend on the subjectivity of the DM [16]. In this sense, the author of [16] proposed a hybrid ant optimization system for solving the EEPD, in which the DM relies on the TOPSIS method to obtain the best alternative from a set of alternatives. In this work, six weights per objective function are assigned and six solutions are obtained according to these weights. Consequently, the DM specifies input values (namely, the weight values) according to his needs; thus the subjectivity of the DM is included.

The DM plays a major role in multiobjective optimization problems [17], as he/she is responsible for determining the priority of each objective as an ordinal rank based on some preferences. The more common preferences are the weight distributions, which assign these weights according to the importance of objectives. Some works on competing objectives that incorporated the DM preferences have been carried out [17,18]; the interested readers are referred to Chiandussi et al. [18]. Therefore, it is very useful to solve the EEPD problem as an FOP to overcome the reliance on DM pReferences

The crow search algorithm (CSA) is one of the latest MHAs proposed in the literature. It was developed by Askarzadeh [19] in 2016 to solve global optimization tasks. The CSA is inspired by the intelligent behavior of crows in nature. Crows are ravening birds, as they keep track of each other to find better food sources. Their capacity to remember where food is stored across seasons is superior to that of other birds. The process of obtaining food resources unobserved by other crows is a difficult task. Crows carry out the following intelligent ruse a crow attempts to cheat another crow by flying to another location if it feels that another one is following it. The main idea from the optimization point of view is based on the crow's natural behavior:(1) the crow is a searcher; (2) the environment is taken as the search space, and each potential location of a food source represents a feasible solution; (3) the quality of a food source is related to the objective function, and the best food source represents the global optimal solution. Similar to other MHAs, the determination of CSA-dependent parameters is still presented as one of the key issues that influence CSA performance. Despite the successful test of this algorithm on some unconstrained optimization problems and its high convergence speed, it suffers from premature convergence and weak diversity, which can occur especially when conducting constrained optimization and/or when the optimum solution resides in a tiny subset of the search space. These disadvantages are the main motivations of this work.

Recent directions for developing reliable and more robust MHAs involve their incorporation of CT [20,21] to replace algorithm-dependent parameters. CT is meant to be the study of chaotic systems that are a subclass of dynamical systems [21]. CT has ergodicity and non-repetition properties, which allow performing the algorithm searching process at higher speeds. Therefore, the search-based chaotic scheme is better than the standard random scheme, which is based on standard probability distributions [21].

Recently, many researchers have integrated MHAs such as PSO [22], firefly algorithm (FFA) [23], gravitational search algorithm [24], and bee colony algorithm [25] with CT. This integration of CT into MHAs yields promising performances once adequate chaotic maps (CMs) are utilized. The main reason for the incorporation of chaos into an algorithm is the high-level of search capability induced by the chaotic scheme as well as the higher acquired diversity of the generated solutions. Thus, it may be worth embedding chaos into other, especially newer, metaheuristic algorithms.

In this paper, we propose a novel chaotic-based CSA (CCSA) to solve fractional (ratio) optimization problems. The proposed approach is established by replacing some parameters that have a random scheme based on the standard distributions (i.e., uniform or Gauss distributions) with a dynamical scheme based on the CT scheme, where four chaotic variants of the CSA are validated and the best one is chosen. The incorporation of CT can improve the performance of the proposed CCSA and allow performing the search process at higher speeds. The proposed algorithm is tested on twenty fractional problems, and validation in terms of the ratio of total emissions to total fuel cost is performed. The experimental results prove that the performance with chaos exhibits higher diversity and mobility.

The main contributions of this study are as follows:

(i) The CCSA is proposed for FOPs. In the CCSA, the chaotic scheme is used to replace the random scheme of the CSA to improve the quality of solutions and avoid stagnation at local optima.
(ii) Four chaotic variants of the CSA are designed to elicit the superior one in terms of performance.
(iii) The performances of different CMs are validated on a set of FOPs.
(iv) The effectiveness of the CCSA is demonstrated via comprehensive experiments and comparisons with different algorithms such as the standard CSA, PSO, FFA, grey wolf algorithm (GWA) and dragonfly algorithm (DA).

(v) The significance of the simulation results is proved by applying the Wilcoxon signed-rank test (WSRT).

To this end, the novelty is contained in innovating different chaotic variants of the CSA as well as the determination of the parameters that are enhanced with chaos. Furthermore, solving the ratio optimization and the fractional EEPD problems undoubtedly becomes a true challenge for the application fields. Finally, the proposed CCSA variants can be applied to assist researchers in different areas.

The rest of the paper is organized as follows. In Section 2, the preliminaries are presented. In Section 3, the standard CSA and main idea behind CT are described. Section 4 presents the proposed CCSA in detail. The experimental results obtained for twenty benchmark fractional problems as well as EEPD problem are given in Section 5. Finally, the conclusion of this work and directions for future research are presented in Section 6.

2. Preliminaries

This section first covers the preliminaries of the FP problems; the motivation of this study is then provided.

2.1. Fractional programming (FP) problem

The FP problem arises in various practical situations when the rates of some objectives, such as (profit/time), (output/input), (profit/revenue) and (waste/total quantity), need to be optimized. The FP problem poses significant challenges because it possesses many local solutions that are not global optima. Without the loss of generality, the general model of the FP problem can be mathematically described as follows:

$$\begin{aligned}
&\text{Min/Max } z(\mathbf{x}) = \sum_{k=1}^{K} \frac{f_k(\mathbf{x})}{g_k(\mathbf{x})} \\
&\text{Subject to :} \\
&\qquad h_j(\boldsymbol{x}) \le 0, \qquad j = 1, 2, \ldots, J, \\
&\qquad q_p(\boldsymbol{x}) = 0, \qquad p = 1, 2, \ldots, P, \\
&\qquad x_i^l \le x_i \le x_i^u, \qquad i = 1, 2, \ldots, n, \\
&\qquad \boldsymbol{x} = (x_1, x_2, \ldots, x_n), \; g_k(\boldsymbol{x}) \ne 0 \; \forall k.
\end{aligned} \tag{1}$$

where f, g, h and q are linear or nonlinear functions, $\mathbf{x} = (x_1, x_2, ..., x_n)$is a vector of n variables, and x_i^l and x_i^u represent the lower and upper bounds of the ith decision variable, respectively. h inequality constraints and q equality constraints exist.

2.2. Motivation of this work

The literature reveals that a significant concern exists within the economic research community regarding the need to solve the fractional tasks of optimization. To satisfy the increasing demands of economic organizations, optimal solutions for fractional tasks must be found, which is undoubtedly becomes a true challenge. However, some of the pure MHAs suffer from premature convergence and weak diversity, particularly when handling highly nonlinear tasks, when handling high-dimensional tasks and/or when the optimum solution resides in a thin subset of the search space. These disadvantages are the main motivations of this work. Additionally, this paper is motivated by the need to introduce different variants of the CCSA to solve FOPs.

3. Materials and methods

3.1. Basic crow search algorithm

The CSA is a new metaheuristic algorithm that was proposed by Alireza Askarzadeh [19]. It was inspired by the cleverness of crows in obtaining food resources. The foraging process of the classical CSA is shown in Fig. 2 and its main steps can be explained as follows [19]:

Step 1: A swarm of crows in the n dimensional search space is initialized, with the algorithm assigning a random vector $\mathbf{x_i} = (x_{i,1}, x_{i,2}, ..., x_{i,n})$ for the ith crow,$i = 1, 2, ..., N$. Furthermore, each crow is characterized by its memory (i.e., initially, a crow's memory is filled with its initial position, $\mathbf{m_i} = (m_{i,1}, m_{i,2}, ..., m_{i,n})$, with all crows having no experience regarding food sources).

Step 2: Each crow is evaluated according to the quality of its position, which is related to the desired objective function.

Step 3: Crows create new locations inside the search space as follows: crow i selects one of the crows from the swarm randomly, i.g., crow j, and follows it to determine the location of food hidden by this crow, with the new location of crow i being generated as follows:

$$x_{i,Iter+1} = \begin{cases} x_{i,Iter} + r_i \times fl_{i,Iter}(m_{j,Iter} - x_{i,Iter}) & a_j \ge AP_{j,Iter} \\ \text{a random location} & \text{otherwise} \end{cases} \tag{2}$$

where r_i, a_j are random numbers between 0 and 1, $AP_{j,Iter}$ is the awareness probability of crow j at iteration $Iter$, $fl_{i,Iter}$ denotes the flight length of crow i at iteration $Iter$ and $m_{j,Iter}$ denotes the memory of crow j at iteration $Iter$.

Fig. 1 shows the effect of the parameter fl on the search capability, with small values of fl ($fl < 1$) leading to exploration of the new position of a crow lying on the dashed line between $m_{j,Iter}$ and $x_{i,Iter}$, as shown in Fig. 1(a), and fl values greater than 1 resulting in the crow lying at a new position along the dashed line outside the line segment between $m_{j,Iter}$ and $x_{i,Iter}$, as shown in Fig. 1(b).

Step 4: After generating the crow's location, the new locations are evaluated, and each crow updates its memory as follows:

$$m_{i,Iter+1} = \begin{cases} x_{i,Iter+1} & z(x_{i,Iter+1}) \succ z(m_{i,Iter}) \\ m_{i,Iter} & \text{otherwise} \end{cases} \tag{3}$$

where $z(.)$ is the ratio (fractional) objective function and the symbol $\succ$ denotes "better than".

Step 5: The algorithm is ended if the maximum number of iterations is met and the best location is determined from memory based on the objective function, which is then recorded as the optimal solution of the candidate problem; otherwise, return to Step 3. The flowchart diagram is shown in Fig. 2.

3.2. Chaos and chaotic map functions

Many MHAs depend on some stochastic components; this means that some probability distributions are utilized to achieve randomness. Because of the ergodicity, non-repetition and mixing properties of chaos, replacing such randomness with CMs can be advantageous. CMs follow a dynamical behavior that is characterized by Eq. (4).

$$x_{k+1} = f(x_k) \tag{4}$$

Actually, under some circumstances, Eq. (4) results in a chaotic behavior. In Eq. (4), x_{k+1} and x_k are the $(k+1)$th and $k$thchaotic number, respectively. The behavior of the chaotic function, $f(.)$, is highly affected by the initial value, denoted as x_0. The mathematical descriptions of these maps are given in Table 1. As the initial point of the CM may have a significant effect on the fluctuation pattern,

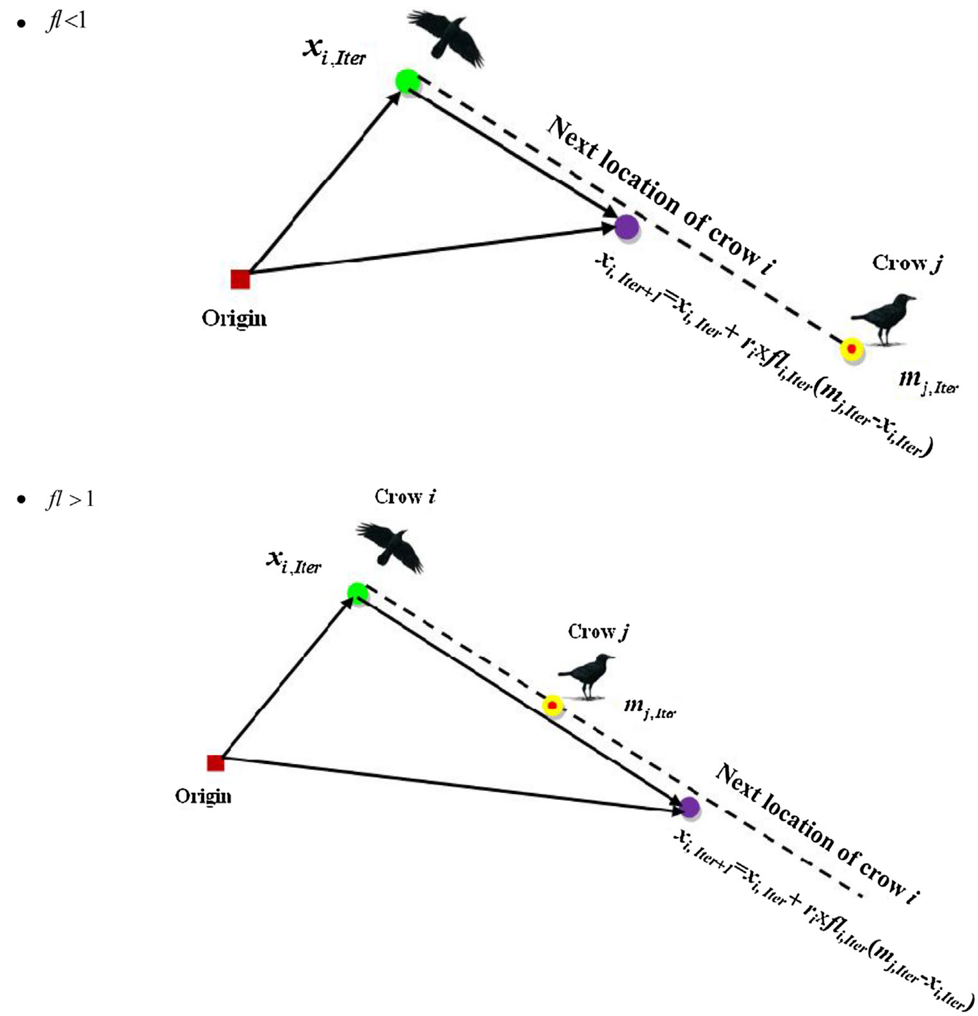

Fig. 1. Searching mechanism by the crow in two states ($fl<1$ and $fl>1$).

Table 1
representation of different chaotic map functions.

NO.	Chaotic name	Mathematical description	Chaotic interval
1	Chebyshev	$x_{i+1}=\cos(\cos^{-1}(x_i))$	[−1,1]
2	Circle	$x_{i+1}=\text{mod}\,(x_i+b-(a/2\pi)\sin(2\pi x_i),1);\ a=0.5,\ b=0.2$	[0,1]
3	Gauss/Mouse	$x_{i+1}=\begin{cases}1, & x_i=0\\ \dfrac{1}{\text{mod}\,(x_i,1)} & otherwise\end{cases}$	[0,1]
4	Iterative	$x_{i+1}=\sin\left(a\pi/x_i\right),\ a=0.7$	[−1,1]
5	Logistic	$x_{i+1}=ax_i(1-x_i),\ \ a=4$	[0,1]
6	Piecewise	$x_{i+1}=\begin{cases}x_i/p & 0\le x_i<p\\ (x_i-p)/(0.5-p) & p\le x_i<0.5\\ (1-p-x_i)/(0.5-p) & 0.5\le x_i<1-p\\ (1-x_i)/p & 1-p\le x_i<1\end{cases},\ p=0.4$	[0,1]
7	Sine	$x_{i+1}=\dfrac{a}{4}\sin(\pi x_i),\ a=4$	[0,1]
8	Singer	$x_{i+1}=\mu(7.86x_i-23.3x_i^2+28.75x_i^3-13.301875x_i^4),\ \mu=1.07$	[0,1]
9	Sinusoidal	$x_{i+1}=ax_i\sin(\pi x_i),\ a=2.3$	[0,1]
10	Tent	$x_{i+1}=\begin{cases}x_i/0.7, & x_i<0.7\\ \dfrac{10}{3}(1-x_i) & x_i\ge 0.7\end{cases}$	[0,1]

several tests in which this initial point is set are conducted. Based on the conducted experiments, the initial point is set to 0.7 for this study. In this context, the maps that do not produce values in the range of [0,1] are normalized to fit into this scale.

4. The proposed chaotic crow search algorithm

Although the MHAs have been designed to remove the computational obstacles of the existing numerical techniques (i.e.,

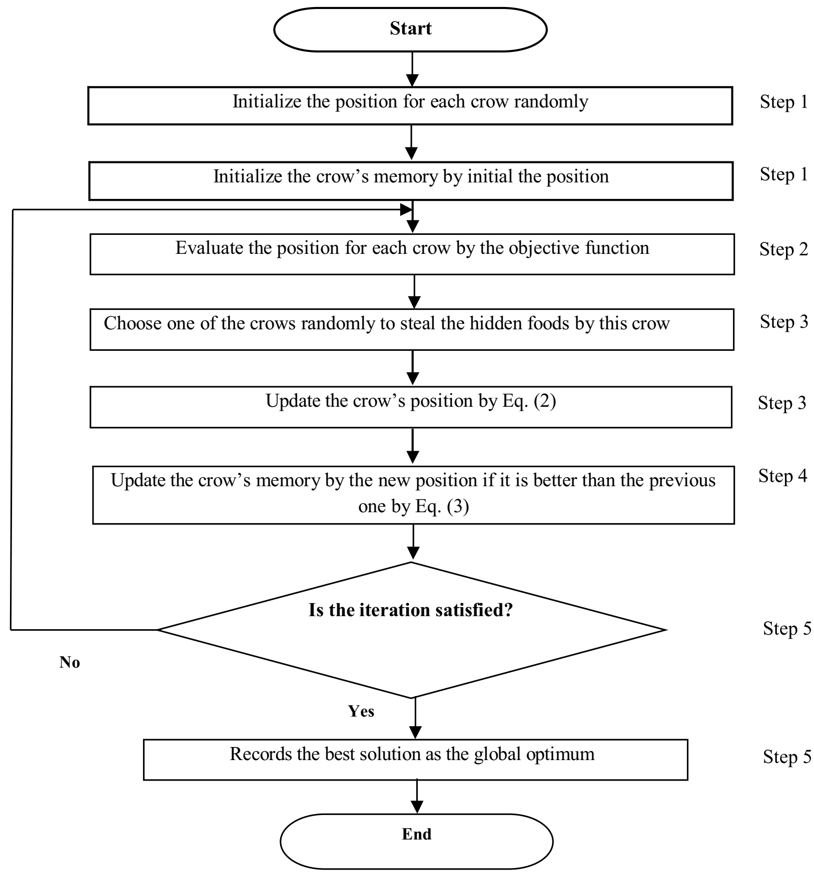

Fig. 2. The basic steps for CSA.

the complicated derivatives, sensitivity to initial values, and large amount of enumeration memory required), their conventional procedures for finding the optimal solution are iterative-based and depend on randomness to imitate natural phenomena. Clearly, the dependency on the randomness parameters may have a major effect on the convergence speed and outcome. In this regard, Hui Lu et al. [26] developed a strategy for replacing random sequences with chaotic sequences, with which they showed that the use of CMs improves the performance of the algorithm. Trelea [27] proposed a new insight regarding parameter selection, introducing a deterministic version of the algorithm by replacing the random numbers with deterministic numbers. Additionally, Trelea showed that the presence of random numbers can slow down the convergence of the algorithm (see the proof in [27]). Consequently, it is desirable to replace the standard distributions (i.e., uniform or Gauss distributions) with non-standard distributions (i.e., the chaotic-based algorithm).

The presented study focuses on hybridizing the CSA with the CMs. This hybridization is called the chaotic crow search algorithm (CCSA). The proposed algorithm operates in chaotic phases as follows: First, a population of roaming crows within the search space is initialized chaotically. During this roaming procedure, the developments of these crows are performed using the CSA procedures. Second, the behavior of the CSA is improved by replacing some of the parameters such that it changes chaotically. Thus, four variants of the CSA, namely, four CCSAs, are introduced and tested on ten different CMs. Then, the best chaotic variant with the best CM is chosen. The following subsections introduce the different variants of the CCSA according to their tuned parameters.

4.1. CCSA-I

The parameter fl of Eq. (2) is modulated using CMs. The new form of Eq. (2) is defined as:

$$x_{i,Iter+1} = \begin{cases} x_{i,Iter} + r_i \times fl_i^{\text{Chaos}(.)}(m_{j,Iter} - x_{i,Iter}) & a_j \geq AP_{j,Iter} \\ \text{a random position} & \text{otherwise} \end{cases} \quad (5)$$

where $fl_i^{\text{Chaos}()}$ is the chaotic flight length parameter, which is defined as follows: $fl_i^{\text{Chaos}(.)} = CM_i.fl_i$. In the standard CSA, fl a fixed value, while in the CCSA-I, it evolves chaotically.

4.2. CCSA-II

The crow discovers the position of the food hidden by another crow by choosing the memory of a certain crow randomly. Thus, in this version of the CCSA, the random selection is replaced with a chaotic selection. Therefore, the memory m is modulated such that it is changed chaotically using the CMs. Eq. (2) is thus modified as follows:

$$x_{i,Iter+1} = \begin{cases} x_{i,Iter} + r_i \times fl_{i,Iter}(m_j^{\text{Chaos}()} - x_{i,Iter}) & a_j \geq AP_{j,t} \\ \text{a random position} & \text{otherwise} \end{cases} \quad (6)$$

where $m_j^{\text{Chaos}()} = \text{round}(CM_j.N)$. The standard CSA generates the hidden position randomly, while CCSA-II generates it according to the CMs.

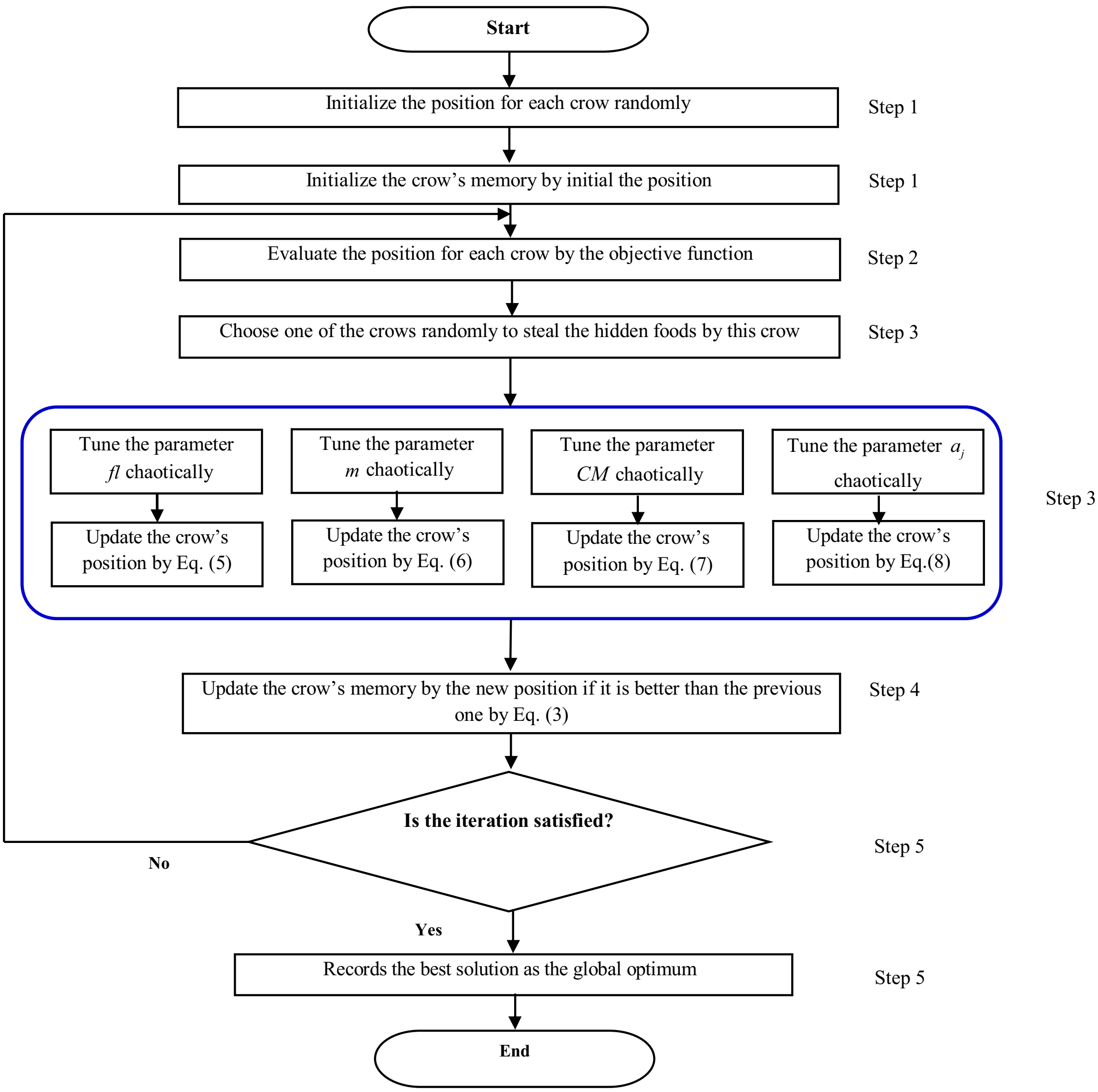

Fig. 3. Architecture of the proposed CCSA algorithm.

4.3. CCSA-III

In this version, the local scheme is presented to search for the best solution using the following equation:

$$x_{i,Iter+1} = x_{Iter}^{best} + CM(Iter).\,\text{levy}(.) \tag{7}$$

where $CM(Iter)$ is a chaotic number between 0 and 1. This version exploits the obtained best solution to improve the quality of the solution via the CM and the levy mechanism.

4.4. CCSA-IV

The parameter a_j of Eq. (2) is modulated using CMs during iterations. Thus, Eq. (2) is changed to:

$$x_{i,Iter+1} = \begin{cases} x_{i,Iter} + r_i \times fl_i(m_{j,Iter} - x_{i,Iter}) & CM_j \geq AP_{j,Iter} \\ \text{a random position} & \text{otherwise} \end{cases} \tag{8}$$

In the standard CSA, $a_j$is created randomly between 0 and 1; in CCSA-IV, it is a chaotic number between 0 and 1

Fig. 3 shows the general architecture of the proposed CCSA algorithm, where the highlighted boxes represent the new variations of the original algorithm.

5. Experiments and results

The proposed CCSA algorithm presented in this paper was validated using twenty fractional benchmark problems [12,28,29]. The mathematical description of these fractional problems and their known optimal solutions associated with their optimal values are given in the Appendix A. Since these problems involve different constraints, the penalty function method is employed [19]. By employing the penalty method, the constrained optimization problem can be transformed into an unconstrained one; then, the proposed CCSA algorithm can be implemented.

Furthermore, the proposed algorithm is validated by regarding the EEPD as an FP problem [13].The robustness and effectiveness of the proposed CCSA are validated by comparing it with prominent algorithms from the literature. The CCSA algorithm is implemented in MATLAB 7 and run on a PC with an Intel Core I 5 (1.8 GHz) processor and 4 GB of RAM. Additionally, extensive experimental tests regarding the parameter settings are performed. The superior per-

Table 2
Parameter configurations for the involved algorithms.

Algorithm	Parameters
PSO [22]	PS = 20, acceleration coefficients: $c_1 = c_2 = 2$, inertia weight w: 0.9–0.2.
FFA [23]	PS = 20, $\alpha = 0.2$, $\gamma = 1$, $\beta_0 = 1$.
GWA [30]	PS = 20, $a = 2 - 0$, $A = 2.a.r_2 - a$, $C = 2.r_1$, $r_1 = random$, $r_2 = random$.
DA [31]	PS = 20, $w = 0.9 - 0.2$, $s = 0.1$, $a = 0.1$, $c = 0.7$, $f = 1$, $e = 1$.
CSA [19]	PS = 20, $AP = 0.1$, $fl = 2$.
CCSA	PS = 20, fl, m, C, a : chaotically tuned.

formance of the proposed CCSA algorithm is validated by carrying out comparisons with other well-established algorithms such as PSO [22], FFA [23], GWA [30], DA [31] and the standard CSA [19]. For a fair comparison, all the comparative algorithms apply the same maximum number of function evaluations (Max_FES), i.e., 3000, in 20 independent runs for every problem. The parameter settings for all algorithms are configured based on the suggestions in the corresponding literature and listed in Table 2.

In order to set the initial point of the CM, different initial points of the CM (i.e., 0.1, 0.3, 0.5, 0.7, and 0.9) are addressed to study their effects on the performance of the solutions quality. In this regard, two selective problems are conducted, where the solutions of these problems and their ranks are reported in Table 3. Based on the reported ranks, we can conclude that the best setting among the different initial points of the CM is 0.7 which its mean rank is highlighted in boldface.

In this subsection, the proposed chaotic variants of the CSA are investigated using the different CM functions, and the best chaotic variant and best CM function are chosen. The best chaotic variant and best CM function is applied to solve the FOPs to validate the chosen variant. The best chaotic variant is chosen based on the mean absolute error (MAE), and the best CM function is chosen based on the closeness to the optimal value.

$$MAE = \frac{\sum_{i=1}^{Z} |A_i - G|}{Z} \tag{9}$$

where A_i represents the obtained result (i.e., the mean of optimum values), G indicates the global optimum value, and Z indicates some samples (here, Z denotes the number of different CMs).

The different chaotic CSA variants (CCSA-I, CCSA-II, CCSA-III and CCSA-IV) are applied to solve four fractional problems, namely, P_2, P_3, P_5, and P_{15}, as presented in Table 4. Obtaining the global optimum for these problems is difficult because the global optimum is inside a long narrow search space. Based on the MAE, as shown in Table 5, CCSA-II is the best chaotic variant with the circle map function. Therefore, it is employed as the proposed chaotic CSA (CCSA).

Accordingly, the performance of the proposed CCSA is compared with that of other algorithms in terms of the mean and standard deviation (Std) of the objective function value on twenty FOPs (P_1-P_{20}), as shown in Table 6. The mean value of the fractional problems quantifies the average accuracy, and Std indicates the reliability of the obtained solution. In terms of the average accuracy and reliability, Table 6 clearly reveals that the CCSA outperforms the other algorithms for most cases.

Furthermore, the number of function evaluations (NFEs) is employed as a standard test to show how fast the optimal solution is obtained among different algorithms. The NFEs is obtained for the proposed CCSA and the comparative algorithms, as listed in Table 7. The comparison of the obtained results indicates that the CCSA is much better than the others in terms of the NFEs (i.e., the CCSA evaluates fewer functions).

Additionally, a comparison of the convergence histories of PSO, FFA, GWA, DA, standard CSA and CCSA is presented in Fig. 4,

Table 3
Effect of different initial points of the CM on the performance of some selective problems.

Problem	P_1					P_{11}					P_{19}				
Initial point of CM	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
Chebyshev	3.6962 (5)	3.7041 (3)	3.7092 (2)	3.7143 (1)	3.7012 (4)	0.2106 (4)	0.2088 (3)	0.2078 (2)	0.2066 (1)	0.2088 (3)	2.8495 (4)	2.8486 (3)	2.8449 (2)	2.8344 (1)	2.8344 (1)
Circle	3.7044 (3)	3.7051 (2)	3.6929 (5)	3.7143 (1)	3.7017 (4)	0.2094 (5)	0.2085 (2)	0.2089 (3)	0.2062 (1)	0.2093 (4)	2.8397 (4)	2.8454 (5)	2.8385 (3)	2.8276 (1)	2.8326 (2)
Gauss/Mouse	3.6979 (2)	3.6960 (3)	3.6933 (5)	3.7117 (1)	3.6955 (4)	0.2083 (1)	0.2090 (3)	0.2099 (5)	0.2085 (2)	0.2098 (4)	2.8530 (5)	2.8480 (4)	2.8381 (2)	2.8326 (1)	2.8381 (2)
Iterative	3.7026 (3)	3.6985 (4)	3.7055 (2)	3.7138 (1)	3.6974 (5)	0.2090 (5)	0.2083 (3)	0.2076 (2)	0.2069 (1)	0.2088 (4)	2.8392 (3)	2.8491 (5)	2.8448 (4)	2.8326 (1)	2.8344 (2)
Logistic	3.6925 (5)	3.7077 (2)	3.6931 (4)	3.7108 (1)	3.6985 (3)	0.2098 (4)	0.2089 (3)	0.2082 (2)	0.2067 (1)	0.2098 (4)	2.8442 (4)	2.8413 (3)	2.8490 (5)	2.8407 (2)	2.8345 (1)
Piecewise	3.7015 (4)	3.6906 (5)	3.7044 (2)	3.7126 (1)	3.7040 (3)	0.2090 (4)	0.2088 (3)	0.2084 (2)	0.2064 (1)	0.2084 (2)	2.8391 (3)	2.8451 (4)	2.8503 (5)	2.8276 (1)	2.8344 (2)
Sine	3.7113 (2)	3.6991 (4)	3.6939 (5)	3.7077 (3)	3.7126 (1)	0.2087 (3)	0.2103 (5)	0.2080 (2)	0.2065 (1)	0.2091 (4)	2.8411 (4)	2.8412 (5)	2.8335 (2)	2.8326 (1)	2.8344 (3)
Singer	3.7121 (2)	3.7100 (4)	3.7112 (3)	3.7143 (1)	3.7085 (5)	0.2086 (4)	0.2093 (5)	0.2083 (2)	0.2085 (3)	0.2080 (1)	2.8379 (4)	2.8435 (5)	2.8344 (2)	2.8282 (1)	2.8346 (3)
Sinusoidal	3.7100 (4)	3.7108 (3)	3.7074 (5)	3.7134 (1)	3.7113 (2)	0.2102 (4)	0.2088 (2)	0.2080 (1)	0.2088 (2)	0.2095 (3)	2.8456 (5)	2.8393 (4)	2.8335 (2)	2.8281 (1)	2.8345 (3)
Tent	3.7000 (4)	3.6995 (5)	3.7016 (3)	3.7134 (1)	3.7059 (2)	0.2091 (3)	0.2097 (4)	0.2085 (2)	0.2072 (1)	0.2102 (5)	2.8449 (3)	2.8398 (2)	2.8326 (1)	2.8326 (1)	2.8491 (4)
Mean Rank	3.4	3.5	3.6	**1.2**	3.3	3.2	3.3	2.3	**1.4**	3.4	3.9	4.0	2.8	**1.1**	2.3

Table 4
Mean of optimal objectives for P_2,P_3,P_5 and P_{15} after 150 iterations and 20 runs.

Prob.	Chaotic map	CCSA I	CCSA II	CCSA III	CCSA IV
P_2	Chebyshev	0.337751	0.336038	0.345872	0.355508
	Circle	0.360120	**0.336037**	0.345757	0.3360384
	Gauss/Mouse	0.399927	0.337085	0.336509	0.353865
	Iterative	0.344515	0.3360439	0.341660	0.336090
	Logistic	0.337084	0.336040	0.347875	0.3360389
	Piecewise	0.346113	0.3360396	0.343044	0.336211
	Sine	0.338032	0.339195	0.344821	0.336203
	Singer	0.336113	0.336144	0.345096	0.336211
	Sinusoidal	0.336044	0.355037	0.348788	0.336211
	Tent	0.339139	0.336039	0.343477	0.336211
P_3	Chebyshev	0.333335	0.333337	0.333474	0.336103
	Circle	0.333343	**0.333333**	0.333545	0.333431
	Gauss/Mouse	0.347259	0.335614	0.336725	0.347955
	Iterative	0.333351	0.333338	0.333504	0.333369
	Logistic	0.333346	0.333341	0.333558	0.333978
	Piecewise	0.333351	0.333335	0.333541	0.333637
	Sine	0.333390	0.333335	0.333632	0.333647
	Singer	0.333342	0.333337	0.333484	0.333351
	Sinusoidal	0.333346	0.333342	0.333437	0.352791
	Tent	0.333347	0.333335	0.333468	0.333513
P_5	Chebyshev	6.499994	6.499895	6.498212	6.500020
	Circle	6.498114	**6.500019**	6.496386	6.498678
	Gauss/Mouse	6.497068	5.974094	5.747329	5.494576
	Iterative	6.495322	6.499972	6.495931	6.377566
	Logistic	6.499882	6.490142	6.497559	6.499569
	Piecewise	6.497763	6.499961	6.497684	6.480342
	Sine	6.485006	6.500002	6.496073	6.494814
	Singer	6.499278	6.499432	6.497747	6.499902
	Sinusoidal	5.494576	6.480272	6.498881	6.499805
	Tent	6.499339	6.499951	6.498007	6.499883
P_{15}	Chebyshev	−2.331449	−2.332154	−2.331841	−2.320648
	Circle	−2.331830	**−2.332218**	−2.332126	−2.332188
	Gauss/Mouse	−2.189800	−2.332188	−2.332057	−2.324015
	Iterative	−2.332152	−2.332131	−2.331989	−2.332141
	Logistic	−2.332215	−2.332118	−2.332121	−2.332216
	Piecewise	−2.331985	−2.332198	−2.331926	−2.332202
	Sine	−2.332188	−2.332118	−2.332141	−2.332188
	Singer	−2.332188	−2.332131	−2.332094	−2.332188
	Sinusoidal	−2.332188	−2.332206	−2.332069	−2.332215
	Tent	−2.332202	−2.332141	−2.332131	−2.332069

Table 5
Rank of chaotic variants for P_2,P_3,P_5 and P_{15}.

Algorithm	MAE	Rank	Algorithm	MAE	Rank
P_2			**P_3**		
CCSA I	1.1484E-2	4	CCSA I	1.4076E-3	3
CCSA II	**2.3698E-3**	1	CCSA II	**2.3146E-4**	1
CCSA III	8.2899E-3	3	CCSA III	5.0349E-4	2
CCSA IV	3.8587E-3	2	CCSA IV	3.8442E-3	4
P_5			**P_{15}**		
CCSA I	1.0336E-1	3	CCSA I	1.4393E-2	4
CCSA II	**5.5630E-2**	1	CCSA II	**1.8500E-05**	1
CCSA III	7.7619E-2	2	CCSA III	1.6900E-4	2
CCSA IV	1.1549E-1	4	CCSA IV	1.9900E-3	3

where some selective fractional problems are considered. This figure shows that the proposed CCSA exhibits a rapid convergence due to its powerful diversity resulting from the CM time series, which return values from the interval [0,1] of the double float type. Fig. 4 reveals that the CCSA outperforms the other algorithms and converges to the global solution very quickly.

5.1. Performance assessment

The performance of the proposed CCSA is assessed using the WSRT [32]. The WSRT is a nonparametric test that is adopted in a hypothesis testing situation to achieve better comparisons. The WSRT is a pair wise test that aims to assess the performances of two algorithms and detects the significant differences between them. It works as follows. First, the difference between the values achieved via two approaches regarding nth of N problems is calculated, and then the obtained differences are sorted according to their absolute values. Second, the sums of positive and negative ranks (i.e., R^+andR^-) are calculated. The significant results are determined using the confidence level that corresponds to the p-value, where $p > 0.05$ represents a failure to reject the null hypothesis, while $p < 0.05$ indicates a rejection of the null hypothesis. The diagram of the main procedures of this test is given in Fig. 5. Therefore, the results of the proposed CCSA are evaluated against the results of other comparative algorithms, which are given in Table 6, using the WSRT. The results obtained by applying the WSRT are reported in Table 8. Based on the WSRT, it can be concluded that the proposed CCSA is a promising algorithm that outperforms the other algorithms in most cases. Additionally, the CCSA can provide a significant balance between exploration and exploitation trends.

The results obtained using the WSRT prove the reliable performance of the proposed CCSA compared to the performances of the others. Additionally, the recorded results affirm statistically the robustness of the proposed CCSA, with these results not having occurred by chance. Therefore, the proposed CCSA is reliable when addressing FOPs.

5.2. Economic environmental power dispatch (EEPD)

This section is devoted to validate the proposed algorithm on the EEPD. The EEPD problem is formulated as an FP problem [13], where the FP aspect reflects the reality of this problem due to the competing and conflicting functions. Thus, the aim is to minimize the ratio of two conflicting and competing targets, i.e., [total emissions function]/[total fuel cost function], which is formulated as follows:

$$\mathrm{Min}Z = \frac{\sum_{i=1}^{NG} 10^{-2}(\alpha_i + \beta_i PG_i + \gamma_i PG_i^2) + \quad |\zeta_i \exp[\lambda_i PG_i]|}{\sum_{i=1}^{NG} (a_i + b_i PG_i + c_i PG_i^2)} \tag{10}$$

Subject to :

$$\sum_{i=1}^{NG} P_{Gi} = P_{Dj} + +P_L, \quad P_L = \sum_{i=1}^{NG}\sum_{j=1}^{NG} P_{Gi}B_{ij}P_{Gi} + \sum_{i=1}^{NG} P_{Gi}B_{0i}P_{Gi} + B_{00}$$
$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max} \ \ \forall i.$$

where α_i, β_i, γ_i, ζ_i, and λ_i are the coefficients of power generation emissions and a_i, b_i, and c_i are the coefficients of the power generation cost function; NG is the number of generation buses; P_{Gi} is the power generation at the bus i, P_D is the load demand andP_L is the total power losses in the system. $P_{Gi}^{\min}$ and $P_{Gi}^{\max}$ are the lower limit and upper limit of the generated power, respectively.

In this context, the proposed approach is tested using the standard IEEE 30-bus test system, the single-line diagram (SLD) of which is shown in Fig. 6. The SLD or the one-line diagram is a simplified diagram that can be used to represent the three-phase power system network as a single line. It contains the main components of a power system, denoted by their standardized schematic symbols, including the generators, branches, bus bars, and loads. The main target is to find the optimal output of the electrical generators to satisfy the system constraints and the load demand with minimum cost and minimum emission. Thus, IEEE-30 bus is one of the widely used test systems in a power system [14]. The test system comprises 41 lines, six generators, located at buses 25–30, and 283.4 MW for the load demand. Table 9 shows the cost and emission coefficients of the six generators of the studied system as well as the minimum and maximum limits of the generated power.

To evaluate the efficiency and feasibility of the proposed CCSA algorithm, it has been applied to a fractional EEPD application. The

Table 6
Comparison between CCSA and other compared algorithms on P_1-P_{20} in terms of mean and std.

Fun.	Meas.	PSO	FFA	GWA	DA	CSA	CCSA
P_1	Mean	−6.6571E + 07	3.35769E + 00	3.7137E + 00	3.7112E + 00	−5.7649E + 06	**3.7143E + 00**
	Std	1.4172E + 08	3.3722E−01	8.9809E−04	7.6511E−03	2.5781E + 07	**7.1712E−12**
P_2	Mean	0.4602E + 00	0.3608E + 00	0.3904E + 00	0.4042E + 00	4.5067E + 06	**0.3360E + 00**
	Std	5.8522E − 17	5.2456E − 02	6.1886E − 02	2.9076E − 02	9.8832E + 06	**0.0000E + 00**
P_3	Mean	0.3460E + 00	0.5836E + 00	0.3336E + 00	0.3371E + 00	0.3688E + 00	**0.3333E + 00**
	Std	2.6779E − 02	6.6559E − 02	9.6902E − 04	1.043E − 02	3.3769E − 02	**5.8510E − 17**
P_4	Mean	−6.9322E + 06	0.3121E + 00	**0.3383E + 00**	0.3079E + 00	**0.3383E + 00**	**0.3383E + 00**
	Std	3.0874E + 07	3.0726E − 02	1.1566E − 08	6.3117E − 02	6.5539E − 12	**2.9058E − 15**
P_5	Mean	−5.2833E + 4	5.2675E + 00	6.4991E + 00	6.4862E + 00	6.3563E + 00	**6.5000E + 00**
	Std	2.3631E + 05	4.1767E − 01	9.3457E − 04	2.6358E − 02	3.3512E − 01	**4.7927E − 13**
P_6	Mean	3.9399E + 00	**4.0608E + 00**	**4.0608E + 00**	4.0607E + 00	4.0376E + 00	**4.0608E + 00**
	Std	1.6135E − 01	2.1682E − 05	7.1606E − 08	1.2284E − 04	6.3192E − 02	**9.7703E − 16**
P_7	Mean	−1.4268E + 08	**5.5167E + 00**	5.4454E + 00	4.8298E + 00	−5.3563E + 07	**5.5167E + 00**
	Std	3.2933E + 08	**9.3623E − 16**	2.2015E − 01	9.1876E − 01	1.0767E + 08	**9.3623E − 16**
P_8	Mean	−1.9252E + 01	−5.0351E + 01	3.6390E + 00	4.8125E + 00	4.9606E + 00	**5.0000E + 00**
	Std	3.6274E + 01	5.9362E + 01	1.6201E + 00	3.9783E − 01	1.9407E − 02	**2.7210E-10**
P_9	Mean	1.1924E + 03	0.4871E + 00	0.4857E + 00	0.4901E + 00	0.5310E + 00	**0.4856E + 00**
	Std	3.7690E + 03	1.1563E − 03	6.6705E − 05	8.2798E − 03	4.9140E − 02	**1.3823E − 12**
P_{10}	Mean	−7.1458E + 05	3.2553E + 00	8.2767E + 00	8.2718E + 00	8.2124E + 00	**8.2799E + 00**
	Std	2.1179E + 06	3.1742E + 00	2.1206E − 03	7.8286E − 03	1.7234E − 01	**7.2014E − 13**
P_{11}	Mean	2.8523E + 08	0.4552E + 00	0.5241E + 00	0.2096E + 00	4.3785E + 08	**0.2062E + 00**
	Std	9.0198E + 08	3.3098E − 01	4.1690E − 01	8.7336E − 03	1.0039E + 09	**3.0812E − 05**
P_{12}	Mean	0.7521E + 00	0.9049E + 00	**0.7519E + 00**	0.8233E + 00	0.7528E + 00	**0.7519E + 00**
	Std	2.0187E − 04	1.6987E − 01	2.1526E − 05	2.2571E − 01	1.2091E − 03	**1.9325E − 15**
P_{13}	Mean	1.2321E + 00	0.9437E + 00	0.8859E + 00	0.8877E + 00	2.0855E + 04	**0.8838E + 00**
	Std	2.8313E − 01	1.1756E − 01	1.8231E − 03	4.6734E − 03	6.5947E + 04	**9.1386E − 14**
P_{14}	Mean	1.3539E + 00	**1.3472E + 00**	**1.3472E + 00**	1.3488E + 00	1.3933E + 00	**1.3472E + 00**
	Std	2.1269E − 02	**2.3401E − 16**	**2.3401E − 16**	2.7447E − 03	2.7947E − 02	**2.3401E − 16**
P_{15}	Mean	1.4152E + 01	−1.9438E + 00	−2.3217E + 00	−2.2602E + 00	1.4245E + 05	**− 2.3212E + 00**
	Std	5.0806E + 01	5.0739E − 01	6.6685E − 03	6.1881E − 02	4.5047E + 05	**1.5915E − 02**
P_{16}	Mean	6.8702E + 09	0.7749E + 00	0.6377E + 00	3.7767E + 09	0.6795E + 00	**0.6352E + 00**
	Std	1.3023E + 10	8.4458E − 02	2.3971E − 03	1.6686e + 10	4.7710E − 02	**1.4289E − 06**
P_{17}	Mean	2.0017E + 00	2.7639E + 00	2.0503E + 00	2.4787E + 00	1.8933E + 00	**1.8833E + 00**
	Std	1.2929E − 01	7.7267E − 01	6.8012E − 01	7.4324E − 01	1.6403E − 02	**2.5523E − 08**
P_{18}	Mean	4.0853E + 00	3.8529E + 00	5.5882E + 10	1.4624E + 12	3.7479E + 00	**3.7109E + 00**
	Std	3.0797E − 01	3.9118E − 02	2.2422E + 11	1.6334E + 12	2.4255E − 02	**1.2203E − 06**
P_{19}	Mean	1.1556E + 11	2.9034E + 00	6.7171E + 10	6.1411E + 11	2.8591E + 00	**2.8276E + 00**
	Std	5.0304E + 11	1.1878E − 02	2.2361E + 11	9.2212E + 11	2.0368E − 02	**5.3222E − 07**
P_{20}	Mean	1.0796E + 09	8.5871E + 08	0.5334E + 00	6.2516E + 07	0.5832E + 00	**0.5333E + 00**
	Std	1.5672E + 09	1.8680E + 09	8.5605E − 05	2.7958E + 08	4.9749E − 02	**3.8399E − 09**

Table 7
Statistical results of NFEs for all test instances.

Prob.	Mean of NFEs						Maximum of NFEs						Minimum of NFEs					
	PSO	FFA	GWO	DA	CSA	CCSA	PSO	FFA	GWO	DA	CSA	CCSA	PSO	FFA	GWO	DA	CSA	CCSA
P_1	3000	3000	2690	2986	2765	554	3000	3000	3000	3000	3000	1220	3000	3000	1140	2760	1360	200
P_2	3000	3000	1756	3000	3000	1105	3000	3000	3000	3000	3000	2020	3000	3000	1500	3000	3000	60
P_3	674	3000	484	3000	3000	118	3000	3000	3000	3000	3000	180	60	3000	100	3000	3000	60
P_4	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
P_5	3000	3000	2703	3000	2542	422	3000	3000	3000	3000	3000	820	3000	3000	820	3000	1080	220
P_6	3000	3000	2313	3000	2700	1072	3000	3000	3000	3000	3000	1860	3000	3000	860	3000	1480	220
P_7	3000	520	3000	3000	3000	40	682	216	852	3000	3000	40	60	60	80	3000	3000	40
P_8	3000	3000	3000	3000	3000	2640	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	2300
P_9	3000	3000	3000	3000	3000	2146	3000	3000	3000	3000	3000	2420	3000	3000	3000	3000	3000	1920
P_{10}	3000	3000	2993	3000	3000	1825	3000	3000	3000	3000	3000	2160	3000	3000	2860	3000	3000	860
P_{11}	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
P_{12}	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000
P_{13}	3000	3000	2861	3000	3000	1384	3000	3000	3000	3000	3000	1580	3000	3000	220	3000	3000	900
P_{14}	390	96	354	3000	3000	42	3000	220	640	3000	3000	60	80	60	100	3000	3000	40
P_{15}	3000	3000	3000	3000	3000	2620	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	2120
P_{16}	3000	3000	3000	2841	3000	2827	3000	3000	3000	3000	3000	3000	3000	3000	3000	2680	3000	840
P_{17}	3000	3000	2987	2755	3000	1566	3000	3000	3000	3000	3000	1800	3000	3000	2740	500	3000	1000
P_{18}	3000	3000	3000	3000	3000	2859	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	2440
P_{19}	3000	3000	3000	3000	3000	2905	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	3000	2640
P_{20}	3000	3000	3000	3000	3000	1720	3000	3000	3000	3000	3000	1720	3000	3000	1380	400	3000	60

proposed CCSA is compared with the standard CSA and PSO, with the statistical measures, such as the best, mean, median and worst objective values, and their standard deviations (Std) and average time (AT) in seconds, obtained over 20 independent runs, being reported in Table 10, where the best result is given in bold font. Additionally, the convergence characteristic and box plot for the ratio of total emission to total fuel cost obtained using the CCSA, standard CSA, and PSO are shown in Fig. 7. The box plot affirms the stability of the proposed CCSA throughout the algorithm runs.

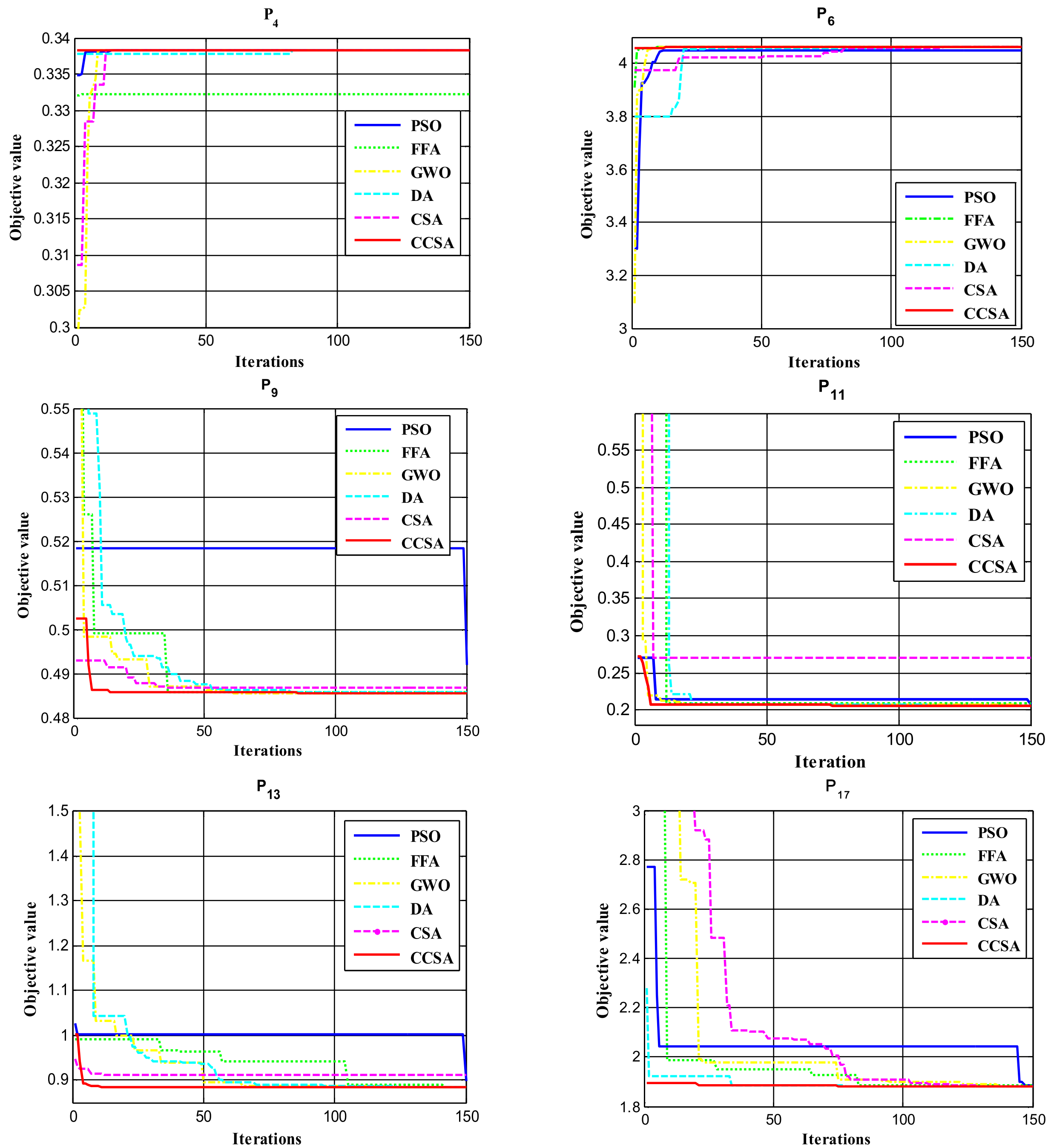

Fig. 4. Convergence behavior for selective problems.

Table 8
Evaluating the results of Table 5 using Wilcoxon test.

Compared methods		Solution evaluations			
The proposed	Compared algorithms	R^+	R^-	p-Value	Winner
CCSA	PSO	91	0	1.4740E-3	CCSA
CCSA	FFA	78	0	2.2180E-3	CCSA
CCSA	GWA	63	3	7.6460E-3	CCSA
CCSA	DA	91	0	1.4740E-3	CCSA
CCSA	CSA	91	0	1.4740E-3	CCSA

Furthermore, a comparison of the proposed CCSA and other reported algorithms [13] is shown in Table 11. Table 11 records the results of the proposed CCSA against other comparative algorithms, such as the genetic algorithm based on the goal programming

Table 9
Generator cost and emission coefficients with generators bounds.

	Coeff.	P_{G1}	P_{G2}	P_{G3}	P_{G4}	P_{G5}	P_{G6}
Cost	a	10	10	20	10	20	10
	b	200	150	180	100	180	150
	c	100	120	40	60	40	100
Emission	α	4.091	2.543	4.258	5.426	4.258	6.131
	β	−5.554	−6.047	−5.094	−3.550	−5.094	−5.555
	γ	6.490	4.638	4.586	3.380	4.586	5.151
	ζ	2.0E-4	5.0E-4	1.0E-6	2.0E-3	1.0E-6	1.0E-5
	λ	2.857	3.333	8.000	2.000	8.000	6.667
Bounds (in p.u).							
Lower bound($P_{Gi}^{\min}$)		0.05	0.05	0.05	0.05	0.05	0.05
Upper bound($P_{Gi}^{\max}$)		0.5	0.6	1	1.2	1	0.6

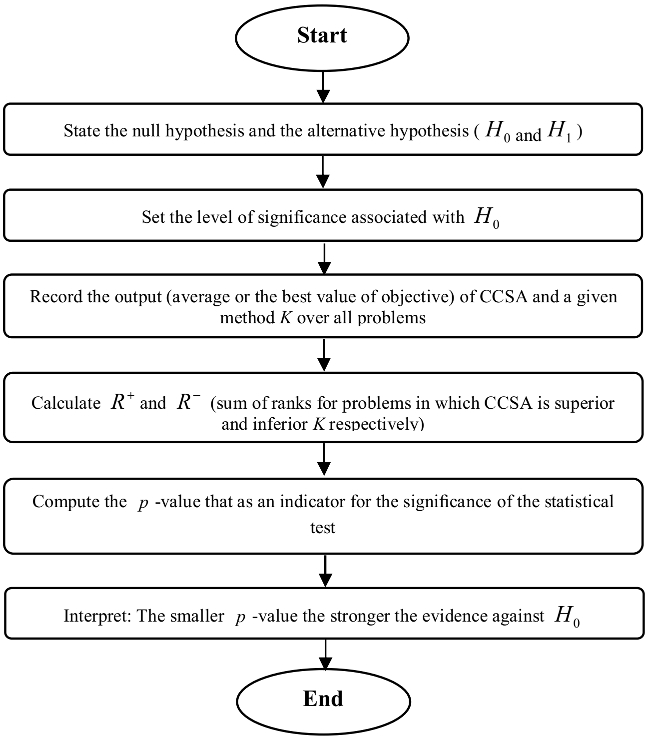

Fig. 5. Wilcoxon test procedures.

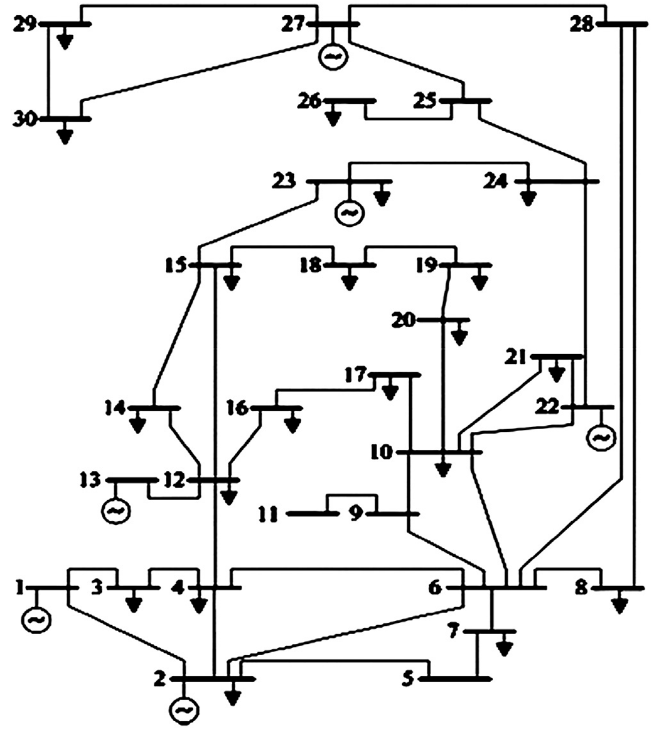

Fig. 6. The architecture for single-line diagram of IEEE 30-bus system.

Table 10
The statistical results of the proposed algorithm, standard CSA, and PSO.

Algorithm	PSO	CSA	Proposed CCSA
Control variables			
P_{G1}	0.4431	0.3154	0.4588
P_{G2}	0.4736	0.5425	0.5287
P_{G3}	0.4622	0.7035	0.6425
P_{G4}	0.2058	0.1331	0.0848
P_{G5}	0.8527	0.7028	0.5923
P_{G6}	0.4345	0.4676	0.5679
Statistical Measures			
Best	3.0289E-4	3.0006E-4	**2.9086E-4**
Mean	3.7121E-4	3.1737E-4	3.0413E-4
Median	3.7276E-4	3.1807E-4	3.0396E-4
Worst	5.2655E-4	3.3419E-4	3.2439E-4
Std	4.5426E-5	9.1946E-6	6.3979E-6
Time (s)	1.0290	1.1217	0.9080

Table 11
Comparison results of obtained solutions by different algorithms.

Method	[total emissions function]/[total fuel cost function]
GAGPM	3.6162E-4
MO-DE/PSO	3.6450E-4
MOPSO	3.6378E-4
SMOPSO	3.6418E-4
TV-MOPSO	3.6242E-4
DE	3.2724E-4
CSA	3.0006E-4
CSA-FL	**2.9086E-4**

model (GAGPM), the multiobjective algorithm based on PSO and differential evolution (MO-DE/PSO), multiobjective PSO (MOPSO), MOPSO with the sigma method (SMOPSO), time variant MOPSO and differential evolution (TV-MOPSO/DE). Additionally, the graph-

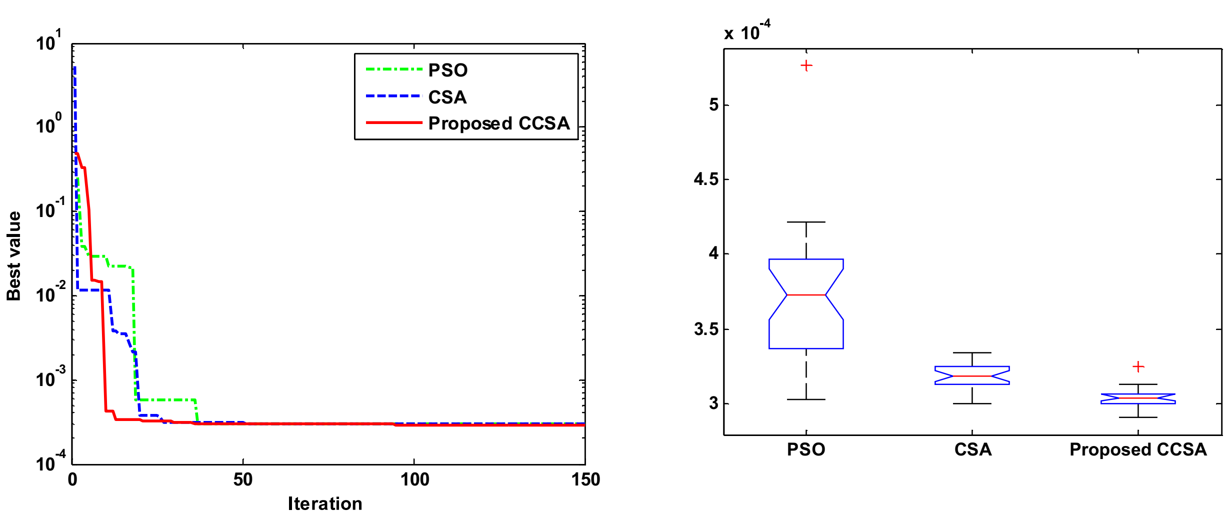

Fig. 7. Convergence curves (left) and Box plot (right) for the economic problem.

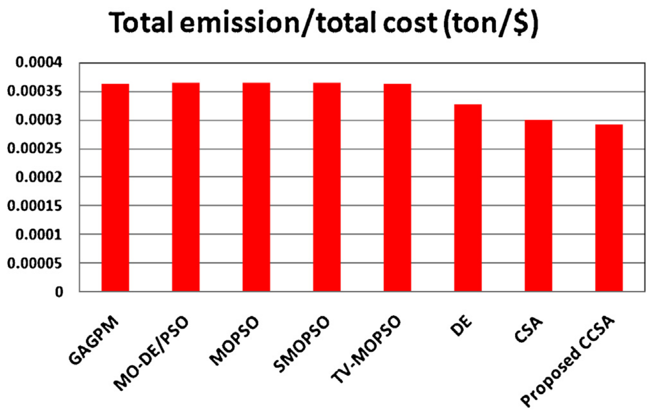

Fig. 8. Graphical representation of the ratio formulation for the EEPD.

ical representations of the [total emissions function]/[total fuel cost function] for the proposed CCSA and the other comparative algorithms are depicted in Fig. 8.

From the reported results (i.e., Table 11 and Fig. 8), the proposed CCSA is clearly superior to the reported algorithms. Thus, the proposed CCSA is a more environmentally friendly and cost-effective scheme in the power dispatch system in terms realization.

In this subsection, a comparative discussion was performed to evaluate the performance of the proposed CCSA regarding the chaotic-based methodology, quality of the obtained solution and computational time. Some pure techniques still struggle to reach an optimal solution in a suitable amount of time and can become stagnate and experience premature convergence, especially for high-dimensional problems. Consequently, the integrated algorithm with chaos offers two features: the ability to avoid premature convergence and the ability to improve the performance using the properties of CMs. On the other hand, chaotic behavior can reveal some algorithm parameter values that can enable the algorithm to reach the optimal solution quickly and therefore accelerate the performance. Thus, the coupling of the proposed CCSA with the chaotic behavior offers great potential to solve the fractional benchmark problems and their applications. The results corresponding to the statistical measures and the WSRT affirm the superior performance of the proposed CCSA among others. Additionally, the recorded results, which did not occur by chance, affirm statistically the robustness of the proposed CCSA.

6. Conclusions and future work

The combined CSA and CM strategy is shown to enhance the quality of solutions found and to accelerate the convergence of the proposed algorithm. The basic CSA with a randomization process is shown to yield unsatisfactory solutions in some cases. Therefore, integrating CSA with CMs can alleviate the premature convergence problem of the standard CSA. The experimental results show that the hybridization of the CSA with the chaotic scheme can accelerate the convergence performance and enhance the quality of solutions. Additionally, the efficiency of the proposed CCSA is justified using the nonparametric WSRT. Thus, it is concluded that the proposed CCSA is very suitable and reliable for solving FOPs.

Careful examination of the evidence will reveal the following benefits of the proposed CCSA.

1) It can efficiently enrich the exploratory capabilities of the CSA by introducing CMs.
2) It surpasses the other algorithms in terms of optimality.
3) It can find the optimal global solutions for fractional problems and electrical applications efficiently.
4) It can improve the convergence speed and boost the performance by reducing the computational time.

Future work will focus on three trends: (i) extending the proposed algorithm to address multiobjective FP problems; (ii) solving several complex engineering optimization problems; and (iii) solving bi-level and multilevel FOPs.

Acknowledgments

The authors are grateful to the anonymous reviewers and the editor for their suggestions in improving the quality of the paper.

Appendix A.

In this section, we give the mathematical description of the each fractional test function and its known optimal solution associated with its optimum value, as appended below (Table A1).

Table A1
Benchmark fractional functions with optimal solution and optimum value.

Mathematical description	Optimal solution	Optimum value
$P_1: \ \mathrm{Max}\, z(\mathbf{x}) = \frac{4x_1+2x_2+10}{x_1+2x_2+5}$ S.t.: $h_1(\mathbf{x}) = x_1+3x_2-30 \le 0,$ $h_2(\mathbf{x}) = -x_1+2x_2-5 \le 0,$ $x_1, x_2 \ge 0.$	(30, 0)	3.714286
$P_2: \ \mathrm{Min}\, z(\mathbf{x}) = \left(\frac{x_1+x_2+1}{x_1+x_2+2}\right)^{1.5} \times \left(\frac{x_1+x_2+3}{x_1+x_2+4}\right)^{2.1} \times$ $\left(\frac{x_1+x_2+5}{x_1+x_2+6}\right)^{1.2} \times \left(\frac{x_1+x_2+7}{x_1+x_2+8}\right)^{1.1}$ S.t.: $q_1(\mathbf{x}) = x_1 - x_2 = 0,$ $1 \le x_1 \le 2,$ $x_1, x_2 \ge 0.$	(1.00000068; 1.0000000458)	0.3360
$P_3: \ \mathrm{Min}\, z(\mathbf{x}) = \frac{x_1+x_2+1}{2x_1-x_2+3}$ S.t.: $0 \le x_1, x_2 \le 1.$	(0,0)	0.3333
$P_4: \ \mathrm{Max}\, z(\mathbf{x}) = \frac{8x_1+7x_2-2.33(9x_1^2+4x_2^2)^{0.5}}{20x_1+12x_2-2.33(3x_1^2+2x_1x_2+4x_2^2)^{0.5}}$ S.t.: $h_1(\mathbf{x}) = 2x_1+x_2-18 \le 0,$ $h_2(\mathbf{x}) = x_1+2x_2-16 \le 0,$ $x_1, x_2 \ge 0$	(0.4248,2.3798)	0.3383
$P_5: \ \mathrm{Max}\, z(\mathbf{x}) = \frac{2x_1+x_2}{x_1} + \frac{2}{x_2}$ S.t.: $h_1(\mathbf{x}) = 2x_1+x_2-6 \le 0,$ $h_2(\mathbf{x}) = 3x_1+x_2-8 \le 0,$ $h_3(\mathbf{x}) = x_1-x_2-1 \le 0,$ $x_1, x_2 \ge 1$	(1,4)	6.5
$P_6: \ \mathrm{Max}\, z(\mathbf{x}) = \frac{-x_1^2+3x_1-x_2^2+3x_2+3.5}{x_1+1} + \frac{x_2}{x_1^2-2x_1+x_2^2-8x_2+20}$ S.t.: $h_1(\mathbf{x}) = 2x_1+x_2-6 \le 0,$ $h_2(\mathbf{x}) = 3x_1+x_2-8 \le 0,$ $h_3(\mathbf{x}) = x_1-x_2-1 \le 0,$ $1 \le x_1 \le 2.25, \ 1 \le x_2 \le 4.$	(1, 1.75)	4.0608
$P_7: \ \mathrm{Max}\, z(\mathbf{x}) = \frac{-x_1^2x_2^{0.5}+2x_1x_2^{-1}-x_2^2+2.8x_1^{-1}x_2+7.5}{-x_1x_2^{1.5}+1} +$ $\frac{x_2+0.1}{-x_1^2x_2^{-1}-3x_1^{-1}+2x_1x_2^2+9x_2^{-1}+12}$ S.t.: $h_1(\mathbf{x}) = 2x_1^{-1}+x_1x_2-4 \le 0,$ $h_2(\mathbf{x}) = x_1+3x_1^{-1}x_2-5 \le 0,$ $h_3(\mathbf{x}) = x_1^2-3x_2^3-2 \le 0,$ $1 \le x_1, x_2 \le 3.$	(1,1)	5.5167
$P_8: \ \mathrm{Max}\, z(\mathbf{x}) = \frac{37x_1+73x_2+13}{13x_1+13x_2+13} + \frac{63x_1-18x_2+39}{13x_1+26x_2+13}$ S.t.: $q_1(\mathbf{x}) = 5x_1+3x_2-3 = 0,$ $1.5 \le x_1 \le 23,$ $x_1, x_2 \ge 0.$	(3,4)	5
$P_9: \ \mathrm{Min}\, z(\mathbf{x}) = \frac{2x_1+x_2}{x_1+10} + \frac{2}{x_2+10}$ S.t.: $h_1(\mathbf{x}) = -x_1^2-x_2^2+3 \le 0,$ $h_2(\mathbf{x}) = -x_1^2-x_2^2+8x_2-3 \le 0,$ $h_3(\mathbf{x}) = 2x_1+x_2-6 \le 0,$ $h_4(\mathbf{x}) = 3x_1+x_2-8 \le 0,$ $h_5(\mathbf{x}) = x_1-x_2-1 \le 0,$ $1 \le x_1 \le 3, \ 1 \le x_2 \le 4.$	(1, 1.4142)	0.48558
$P_{10}: \ \mathrm{Max}\, z(\mathbf{x}) = \left(\frac{13x_1+13x_2+13}{37x_1+73x_2+13}\right)^{-1.4} \times \left(\frac{64x_1-18x_2+39}{13x_1+26x_2+13}\right)^{1.2} -$ $\left(\frac{x_1+2x_2+5x_3+50}{x_1+5x_2+5x_3+50}\right)^{0.5} \times \left(\frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50}\right)^{1.1}$ S.t.: $h_1(\mathbf{x}) = 2x_1+x_2+5x_3-10 \le 0,$ $q_1(\mathbf{x}) = 5x_1-3x_2 = 3,$ $1.5 \le x_1 \le 3,$ $x_1, x_2, x_3 \ge 0.$	(1.5,1.5,0)	8.2799
$P_{11}: \ \mathrm{Min}\, z(\mathbf{x}) = \frac{4x_1^2+x_2^2+1}{2x_1^2+5x_2^2+1}$ S.t.: $h_1(\mathbf{x}) = -5x_1+3x_2-15 \le 0,$ $h_2(\mathbf{x}) = 4x_1+3x_2-45 \le 0,$ $x_1, x_2 \ge 0$	(0.1, 5.12)	0.21
$P_{12}: \ \mathrm{Min}\, z(\mathbf{x}) = \frac{3x_1^2+5x_2^2+1}{4x_1^2+3x_2^2+x_2}$ S.t.: $h_1(\mathbf{x}) = -5x_1+3x_2-15 \le 0,$ $h_2(\mathbf{x}) = 4x_1+3x_2-45 \le 0,$ $x_1, x_2 \ge 0$	(11.16, 0.11)	0.75

Table A1 (*Continued*)

Mathematical description	Optimal solution	Optimum value
$\begin{aligned} &P_{13}: \text{Min}\, z(\mathbf{x}) = \frac{0.25(-x_1^2+3x_1+2x_2^2+3x_2+3.5)}{x_1+1} - \frac{1.75x_2}{x_1^2-2x_1+x_2^2-8x_2+20} \\ &\text{S.t.}: \quad h_1(\mathbf{x}) = 3x_1+x_2-8 \le 0, \\ &h_2(\mathbf{x}) = x_1 - x_1^{-1}x_2 - 1 \le 0, \\ &h_3(\mathbf{x}) = 2x_1x_2^{-1} + x_2 - 6 \le 0, \\ &1 \le x_1, x_2 \le 3. \end{aligned}$	(1.618033989, 1.0)	0.883868686
$\begin{aligned} &P_{14}: \text{Min}\, z(\mathbf{x}) = \frac{x_1+x_2+11}{x_1+x_2+2} + \frac{x_1+x_2+3}{x_1+x_2+4} + \frac{x_1+x_2+5}{x_1+x_2+6} - \frac{x_1+x_2+8}{x_1+x_2+7} \\ &\text{S.t.}: \quad h_1(\mathbf{x}) = x_1x_2^2 + x_1^2x_2 - 10 \le 0, \\ &1 \le x_1, x_2 \le 2 \end{aligned}$	(1.0000000458, 1.0)	1.347219844
$\begin{aligned} &P_{15}: \text{Min}\, z(\mathbf{x}) = \frac{-1.35(x_1^2x_2^{0.5} - 2x_1x_2^{-1} + x_2^2 - 2.8x_1^{-1}x_2 + 7.5)}{x_1x_2^{1.5}+1} + \frac{12.99(x_2+0.1)}{x_1^2x_2^{-1} - 3x_1^{-1} + 2x_1x_2^2 - 9x_2^{-1} + 12} \\ &\text{S.t.}: \quad h_1(\mathbf{x}) = 2x_1^{-1} + x_1x_2 - 4 \le 0, \\ &h_2(\mathbf{x}) = x_1 + 3x_1^{-1}x_2 - 5 \le 0, \\ &h_3(\mathbf{x}) = x_1^2 - 3x_2^3 - 2 \le 0, \\ &1 \le x_1, x_2 \le 3. \end{aligned}$	(2.698690670, 1.20758556)	−2.3322
$\begin{aligned} &P_{16}: \text{Min}\, z(\mathbf{x}) = \frac{x_1^2+x_2^2+2x_1x_3}{x_3^2+5x_1x_3} + \frac{x_1+1}{x_1^2-2x_1+x_2^2-8x_2+20} \\ &\text{S.t.}: \quad g_1(\mathbf{x}) = x_1^2+x_2^2+x_3-5 \le 0, \\ &g_2(\mathbf{x}) = (x_1-2)^2 + x_2^2 + x_3^2 - 5 \le 0, \\ &1 \le x_1, x_2, x_3 \le 3 \end{aligned}$	(1,1, 1.732051)	0.6352758
$\begin{aligned} &P_{17}: \text{Min}\, z(\mathbf{x}) = \frac{x_2}{x_1^2-2x_1+x_2^2-8x_2+20} - \frac{x_1^2+2x_2-3x_1-3x_2-10}{x_1+1} \\ &\text{S.t.}: \quad g_1(\mathbf{x}) = 2x_1+x_2^2-6x_2 \le 0, \\ &g_2(\mathbf{x}) = 3x_1+x_2-8 \le 0, \\ &g_3(\mathbf{x}) = x_1^2-x_2-x_1 \le 0, \\ &1 \le x_1, x_2 \le 3 \end{aligned}$	(1.666667, 3)	1.883333
$\begin{aligned} &P_{18}: \text{Min}\, z(\mathbf{x}) = \frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} + \frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50} + \frac{x_1+2x_2+4x_3+50}{x_1+5x_2+5x_3+50} + \frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50} \\ &\text{S.t.}: \quad g_1(\mathbf{x}) = 2x_1+x_2+5x_3-10 \le 0, \\ &g_2(\mathbf{x}) = x_1+6x_2+2x_3-10 \le 0, \\ &g_3(\mathbf{x}) = 10-9x_1-7x_2-3x_3 \le 0, \\ &x_1, x_2, x_3 \ge 0 \end{aligned}$	(0, 1.666667, 0)	3.7109
$\begin{aligned} &P_{19}: \text{Min}\, z(\mathbf{x}) = \frac{3x_1+5x_2+3x_3+50}{3x_1+4x_2+3x_3+50} + \frac{3x_1+4x_3+50}{4x_1+3x_2+2x_3+50} + \frac{4x_1+2x_2+4x_3+50}{5x_1+4x_2+3x_3+50} + \frac{4x_1+2x_2+4x_3+50}{5x_2+4x_2+3x_3+50} \\ &\text{S.t.}: \quad g_1(\mathbf{x}) = 2x_1+x_2+5x_3-10 \le 0, \\ &g_2(\mathbf{x}) = x_1+6x_2+2x_3-10 \le 0, \\ &g_3(\mathbf{x}) = 10-9x_1-7x_2-3x_3 \le 0, \\ &x_1, x_2, x_3 \ge 0 \end{aligned}$	(4.545455,0.9090909,0)	2.827616
$\begin{aligned} &P_{20}: \text{Min}\, z(\mathbf{x}) = \left(\frac{-x_1+2x_2+2}{3x_1-4x_2+5}\right) \times \left(\frac{4x_1-3x_3+4}{-2x_1+x_2+3}\right) \\ &\text{S.t.}: \quad g_1(\mathbf{x}) = x_1+x_2-1.5 \le 0, \\ &g_2(\mathbf{x}) = x_1-x_2 \le 0, \\ &1 \le x_1, x_2 \le 2 \end{aligned}$	(0,0)	0.5333333

References

[1] E.B. Bajalinove, Linear Fractional Programming: Theory, Methods, Applications and Software, Kluwer Academic Publishers, 2003.

[2] J.R. Isbell, W.H. Marlow, Atrition games, Nav. Res. Logist. Q. 3 (1956) 71–94.

[3] A. Charnes, W.W. Cooper, Programming with linear functional, Nav. Res. Logist. Q. 9 (1962) 181–186.

[4] P. Pandey, A.P. Punnen, A simplex algorithm for piecewise-linear fractional programming problems, Eur. J. Oper. Res. 178 (2007) 343–358.

[5] B. Mojtaba, S.R. Azmin, S. Mansour, Solving linear fractional programming problems with interval coefficients in the objective function-A new approach, Appl. Math. Sci. 6 (2012) 3442–3452.

[6] Rizk M. Rizk-Allah, Aboul Ella Hassanien, New binary bat algorithm for solving 0–1 knapsack problem, Complex Intell. Syst. (2017), http://dx.doi.org/10.1007/s40747-017-0050-z.

[7] R.M. Rizk-Allah, M. Zaki Elsayed, A.A. El-Sawy, Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems, Appl. Math. Comput. 224 (2013) 473–483.

[8] A. Sameeullah, S.D. Devi, B. Palaniappan, Genetic algorithm based method to solve linear fractional programming problem, Asian J. Inf. Technol. 7 (2008) 83–86.

[9] H.I. Calvete, C. Gale, P.M. Mateo, A genetic algorithm for solving linear fractional bilevel problems, Ann. Oper. Res. 166 (2009) 39–56.

[10] M. Jaberipour, E. Khorram, Solving the sum-of-ratios problems by a harmony search algorithm, J. Comput. Appl. Math. 234 (2010) 733–742.

[11] M. Abdel-Baset, I.M. Hezam, An improved flower pollination algorithm for ratios optimization problems, Appl. Math. Inf. Sci. Lett. 3 (2015) 83–91.

[12] O. Abdel Raouf, I.M. Hezam, Solving fractional programming problems based on swarm intelligence, J. Ind. Eng. Int. (2014) 10–56.

[13] F. Chen, G.H. Huang, Y.R. Fan, R.F. Liao, A nonlinear fractional programming approach for environmental–economic power dispatch, Electr. Power Energy Syst. 78 (2016) 463–469.

[14] Rizk M. Rizk-Allah, Ragab A. El-Sehiemy, Gai-Ge Wang, A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution, Appl. Soft Comput. 63 (2018) 206–222.

[15] C.T. Chang, Multi-choice goal programming, Omega 35 (2007) 389–396.

[16] Abd Allah A. Mousa, Hybrid ant optimization system for multiobjective economic emission load dispatch problem under fuzziness, Swarm Evol. Comput. 18 (2014) 11–21.

[17] Shamik Chaudhuri, Kalyanmoy Deb, An interactive evolutionary multi-objective optimization and decision making procedure, Appl. Soft Comput. 10 (2010) 496–511.

[18] G. Chiandussi, M. Codegone, S. Ferrero, F.E. Varesio, Comparison of multi-objective optimization methodologies for engineering applications, Comput. Math. Appl. 63 (2012) 912–942.
[19] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[20] Xiaofang Yuan, Ting Zhang, Yongzhong Xiang, Xiangshan Dai, Parallel chaos optimization algorithm with migration and merging operation, Appl. Soft Comput. 35 (2015) 591–604.
[21] S. Saremi, S. Mirjalili, A. Lewis, Biogeography-based optimisation with chaos, Neural Comput. Appl. 25 (2014) 1077–1097.
[22] B. Alatas, E. Akin, A. Bedri Ozer, Chaos embedded particle swarm optimization algorithms, Chaos Soliton Fractals 40 (2009) 1715–1734.
[23] A.H. Gandomi, X.S. Yang, S. Talatahari, A.H. Alavi, Firefly algorithm with chaos, Commun. Nonlinear Sci. Numer Simul. 18 (2013) 89–98.
[24] Seyedali Mirjalili, Amir H. Gandomi, Chaotic gravitational constants for the gravitational search algorithm, Appl. Soft Comput. 53 (2017) 407–419.
[25] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Syst. Appl. 37 (2010) 5682–5687.
[26] Hui Lu, Xiaoteng Wang, Zongming Fei, Meikang Qiu, The effects of using chaotic map on improving the Performance of multiobjective evolutionary algorithms, Math. Prob. Eng. 2014 (2014) 1–16.
[27] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Inf. Process. Lett. 85 (2003) 317–325.
[28] S. Peiping, M. Yuan, C. Yongqiang, Global optimization for the generalized polynomial sum of ratios problem, J. Glob. Optim. 50 (2011) 439–455.
[29] H. Jiao, Z. Wang, Y. Chen, Global optimization algorithm for sum of generalized polynomial ratios problem, Appl. Math. Model. 37 (2013) 187–197.
[30] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Software 69 (2014) (2014) 46–61.
[31] S. Mirjalili, 'Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems', Neural Comput. Appl. 27 (2016) 1053–1073.
[32] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18.