Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi

Afyon Kocatepe University Journal of Science and Engineering

AKÜ FEMÜBİD **23** (2023) 055101 (1122-1141) AKU J. Sci. Eng. **23** (2023) 055101 (1122-1141)

DOI: 10.35414/akufemubid.1263731

Araştırma Makalesi / Research Article

Chaotic Snake Optimizer

Gülnur YILDIZDAN[1]

[1] *Kulu Vocational School, Selcuk University, Konya, Turkey*

Corresponding author e-posta: gavsar@selcuk.edu.tr ORCID ID: https://orcid.org/0000-0001-6252-9012

Geliş Tarihi: 11.03.2023 Kabul Tarihi: 18.09.2023

Keywords
Chaotic maps; Continuous optimization; Snake optimizer; Metaheuristic algorithms

Abstract

Metaheuristic algorithms provide approximate or optimal solutions for optimization problems in a reasonable time. With this feature, metaheuristic algorithms have become an impressive research area for solving difficult optimization problems. Snake Optimizer is a population-based metaheuristic algorithm inspired by the mating behavior of snakes. In this study, different chaotic maps were integrated into the parameters of the algorithm instead of random number sequences to improve the performance of Snake Optimizer, and Snake Optimizer variants using four different chaotic mappings were proposed. The performances of these proposed variants for eight different chaotic maps were examined on classical and CEC2019 test functions. The results revealed that the proposed algorithms contribute to the improvement of Snake Optimizer performance. In the comparison with the literature, the proposed Chaotic Snake Optimizer algorithm found the best mean values in many functions and took second place among the algorithms. As a result of the tests, Chaotic Snake Optimizer has been shown to be a promising, successful, and preferable algorithm.

Kaotik Yılan Optimize Edici

Anahtar kelimeler
Kaotik haritalar; Sürekli optimizasyon; Yılan optimize edici; Metasezgisel algoritmalar

Öz

Metasezgisel algoritmalar, optimizasyon problemlerine makul bir sürede yaklaşık veya optimal çözümler sunar. Bu özelliği ile metasezgisel algoritmalar zor optimizasyon problemlerini çözmek için etkileyici bir araştırma alanı haline gelmiştir. Yılan Optimize Edici, yılanların çiftleşme davranışlarından esinlenen popülasyon tabanlı bir metasezgisel algoritmadır. Bu çalışmada, Yılan Optimize Edicinin performansını iyileştirmek için rastgele sayı dizileri yerine algoritmanın parametrelerine farklı kaotik haritalar entegre edilmiş ve dört farklı kaotik haritalama kullanılarak Yılan Optimize Edici varyantları önerilmiştir. Önerilen bu varyantların sekiz farklı kaotik harita için performansları klasik ve CEC2019 test fonksiyonları üzerinde incelenmiştir. Sonuçlar, önerilen algoritmaların Yılan Optimize Edici performansının iyileştirilmesine katkıda bulunduğunu ortaya koydu. Literatür ile karşılaştırıldığında önerilen Kaotik Optimize Edici algoritması birçok fonksiyonda en iyi ortalama değerleri bulmuş ve algoritmalar arasında ikinci sırada yer almıştır. Yapılan testler sonucunda, Kaotik Yılan Optimize Edicinin gelecek vadeden, başarılı ve tercih edilebilir bir algoritma olduğu görülmüştür.

1. Introduction

The term metaheuristic algorithm refers to higher-level heuristics that can be used to solve many different types of optimization problems. Optimization problems solved by metaheuristic algorithms have a wide variety, from single to multi-objective, from continuous to discrete, and from constrained to unconstrained (Dokeroglu *et al*. 2019). Most of these problems are NP-hard problems, which are a group of optimization problems that cannot be solved in polynomial time (Daliri *et al*. 2022). Solving these problems is often complex and not easy. Metaheuristic algorithms provide approximate or optimal solutions with reasonable execution times for these problems. With this feature, metaheuristics have become an impressive field of research that is improving day by day in solving NP-hard problems. Since the first metaheuristic algorithm was proposed, great progress has been made, and many new algorithms continue to be proposed every day.

There are five main categories of meta-heuristic algorithms that are derived from natural sources. These categories include evolution-based, swarm-based, physics/chemistry-based, human-based, and others. Swarm-based algorithms have modeled the self-organization observed in swarm behavior among social creatures in nature (Wang *et al*. 2022). The Snake Optimizer algorithm, one of the recently proposed metaheuristic algorithms, is a swarm-based algorithm inspired by the mating behavior of snakes. The algorithm has attracted attention since the day it was proposed and has been used on different problems. Klimov *et al*. (2020) used the Snake Optimizer to optimize the frequencies at which quantum logic gates are applied in superconducting qubits. (Li *et al*.2022) proposed a new method with optimized variable mode decomposition with snake optimization and a double-threshold correlation coefficient to eliminate ship-radiated noise. (Rawa 2022) used a hybridization of snake optimizer and sine-cosine algorithms for the transmission expansion planning problem. (El-Saleh *et al*. 2023) introduced a Binary Snake Optimizer-based feature selection approach to improve the performance of intrusion detection systems. (Dai *et al*. 2022) developed a model based on snake optimization to improve the accuracy of the thermal error estimation of a motorized spindle. (Liu *et al*. 2023) proposed a chaotic gaussian snake optimization algorithm for sensor node optimization in soil monitoring wireless sensor networks. (Fu *et al*. 2022) proposed a gas explosion prediction model in which the improved snake optimization algorithm is integrated. Sine chaos mapping, spiral search strategy, and snake dynamic adaptive weight were used in the snake optimizer to increase the search capability. (Cheng *et al*.2022) presented a neural network-based prediction model for fingerprint indoor localization technology whose weights and thresholds were adjusted using the snake optimization technique. (Omran *et al*.2022) used the snake optimizer for optimum sizing of a complete green photovoltaic battery fast charging station for electric vehicles. Yao *et al*. (2023) proposed an improved SO with a new opposite learning strategy and four new dynamic update mechanisms, including tent-chaos logic, to improve SO performance. (Vellingiri *et al*. 2023) proposed the chaotic SO algorithm, a hybrid algorithm combining chaotic maps with SO for the single diode model. (Gong *et al*. 2023) proposed a multi-objective clustering model for an industrial wireless sensor network. A novel chaotic multilevel elite clone snake optimization method is designed to improve the optimal clustering mechanism in this model.

In this study, different chaotic maps instead of random number sequences are integrated into the algorithm's parameters to improve the performance of standard SO, and SO variants using four different chaotic mappings (CSO) are proposed. The performances of these proposed algorithms for eight different chaotic maps were examined on the classical and CEC2019 test functions, and the results were compared with those of the standard SO and each other. There are not many studies in the literature about chaotic map-based SO. This study was conducted with the motivation to reveal the chaotic map-based performance of the algorithm and contribute to the literature on chaotic map-based versions of SO and their performance. The rest of the study is structured in the following manner: The snake optimizer and chaotic maps are explained in the material and method section in the second section. In the third section, the proposed chaotic snake optimizer is explained. In the fourth section, the findings of the tests are discussed. Finally, the fifth section includes the conclusion and future work.

2. Materials and Methods

2.1. Snake Optimizer (SO)

Hashim and Hussien (2022) proposed the Snake Optimizer (SO), a population-based metaheuristic algorithm, in 2022 to imitate snakes' mating behavior. Snakes engage in their mating behavior when it is cold outside and they can find food. SO is initialized by generating a random population according to Equation 1. The population is then divided equally into two groups, male and female (Equation 2).

$$x_{i,j} = Lb_j + r * (Ub_j - Lb_j), \quad i = 1,2,\dots,N \quad j = 1,2,\dots,m \tag{1}$$

$$N_{female} \cong \frac{N}{2}, \quad N_{male} = N - N_{female} \tag{2}$$

where $x_{i,j}$ is the jth dimension of the ith snake, m is the number of dimensions, N is the population size, r is a random number in the range (0,1), and Ub and Lb are the upper and lower bounds of the jth dimension, respectively. In addition, N_{female} indicates the number of female snakes, while N_{male} indicates the number of male snakes. The best individual from each group (i.e. $F_{best,female}$ and $F_{best,male}$) is found in each iteration.

In the algorithm, temperature (T) and food quality (FQ) are calculated according to Equations 3 and 4. In these equations, t is the current iteration number and t_{max} is the total number of iterations. c_1 is a constant ($c_1 = 0.5$).

$$T = \exp(\frac{-t}{t_{max}}) \tag{3}$$

$$FQ = c_1 \times \exp(\frac{t - t_{max}}{t_{max}}) \tag{4}$$

The snakes select a random location to search for food when FQ < Th (Th=Threshold= 0.25). Then they update their position. The exploration behavior of male and female snakes is expressed mathematically in Equations 5 and 6, respectively.

$$x_{i,j}(t+1) = x_{k,j}(t) \mp c_2 \times A_{i,male}\big((Ub - Lb) \times r_1 + Lb\big), \quad where\ A_{i,male} = \exp(\frac{-F_{r,male}}{F_{i,male}}) \tag{5}$$

$$x_{i,j}(t+1) = x_{k,j}(t+1) \mp c_2 \times A_{i,female}\big((Ub - Lb) \times r_2 + Lb\big), \quad where\ A_{i,female} = \exp(\frac{-F_{r,female}}{F_{i,female}}) \tag{6}$$

In these equations, k is a random integer in the range $(1, \frac{N}{2})$, $x_{k,j}$ is a randomly selected male/female snake from the male/female snake population, and $r_1$and r_2 are random numbers in the range (0,1). $A_{i,male}$ and $A_{i,female}$ are the food-finding abilities of male and female snakes, respectively. $F_{r,male}$ represents the fitness of a previously chosen random male snake, while $F_{r,female}$ represents the fitness of a previously chosen random female snake. $F_{i,male}$ and $F_{i,female}$ are the ith male and female snake fitness, respectively. The flag direction operator ($\mp$) scans all possible directions randomly in the given search space.

In the exploitation phase, the algorithm looks for the best solutions under the following two conditions:

If FQ>Th

- If the Temperature > Th (0.6) (hot), the snakes will only move to the food according to Equation 7.

$$x_{i,j}(t+1) = x_f \mp c_3 \times T \times r_3 \times (x_f - x_{i,j}(t)) \tag{7}$$

where $x_{i,j}$ indicates where male and female snakes are positioned, x_f denotes the best snakes, c_3 is a constant equal to 2, and r_3 is a random number in the range (0,1).

If FQ < Th (Th < 0.6) (cold), the snakes either fight or mate.

- Figthing

The fighting ability of the male snake F_{male} and female snake F_{female} can be expressed as in Equations 8 and 9.

$$x_{i,j}(t+1) = x_{i,j}(t) \pm c_4 \times F_{i,male} \times r_4 \times \left(x_{best,female} - x_{i,male}(t)\right), \quad where\ F_{i,male} = \exp\left(\frac{-F_{best,f}}{F_i}\right) \tag{8}$$

$$x_{i,j}(t+1) = x_{i,j}(t) \pm c_4 \times F_{i,female} \times r_5 \times \left(x_{best,male} - x_{i,female}(t+1)\right), where\ F_{i,female} = \exp(\frac{-F_{best,male}}{F_i}) \tag{9}$$

where $x_{i,j}$ indicates where male and female snakes are positioned, $x_{best,female}$ and $x_{best,male}$ denotes the positions of the best snakes in the female and male groups, respectively. $F_{i,male}$ indicates male snake-fighting ability, while $F_{i,female}$ indicates female snake-fighting ability. In addition, c_4 is a constant equal to 2, and r_4 and r_5 are random numbers in the range (0,1).

- Mating

In mating, male and female snakes update their positions as in Equation 10 and 11.

$$x_{i,male}(t+1) = x_{i,m}(t) \pm c_5 \times M_{i,male} \times r_6 \times \left(FQ \times x_{i,female} - x_{i,male}(t)\right), \quad where\ M_{i,male} = \exp(\frac{-f_{i,female}}{f_{i,male}}) \tag{10}$$

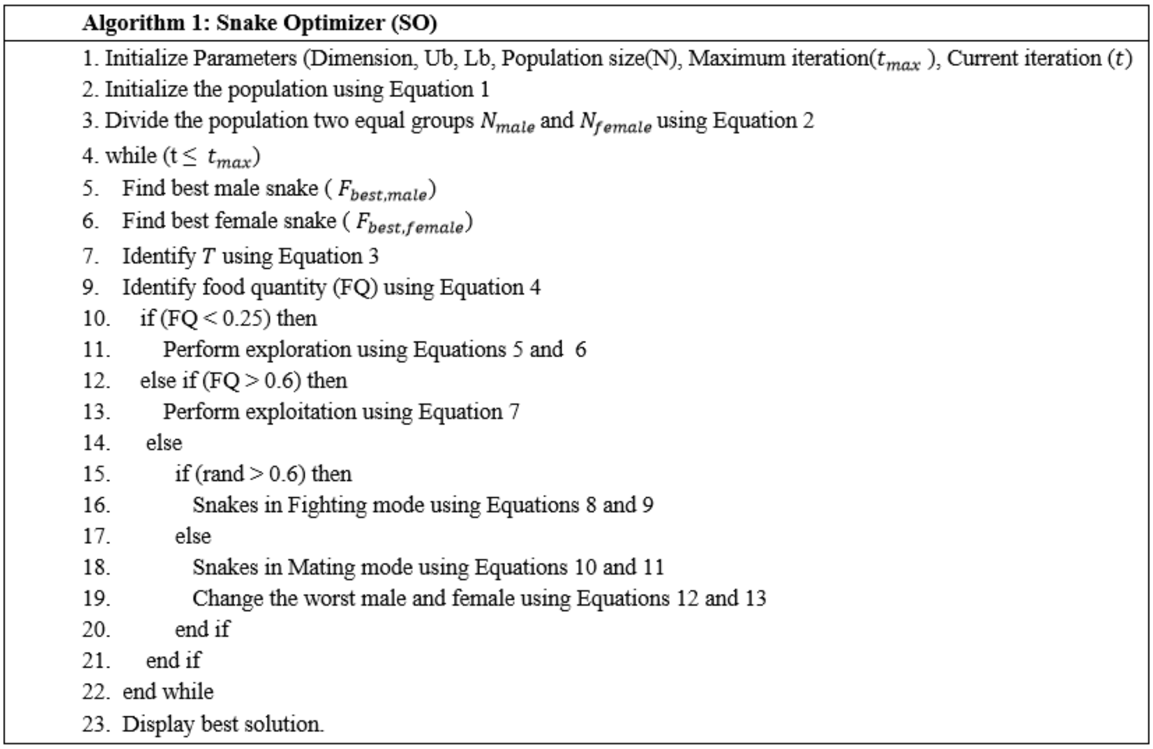
$$x_{i,female}(t+1) = x_{i,f}(t) \pm c_5 \times M_{i,female} \times r_7 \times \left(FQ \times x_{i,male} - x_{i,female}(t+1)\right), \ where\ M_{i,female} = \exp\left(\frac{-f_{i,male}}{f_{i,female}}\right) \tag{11}$$

where $x_{i,m}$ and $x_{i,f}$ are the ith positions of male and female snakes, and $M_{i,male}$ and $M_{i,female}$ refer to male and female mating ability. c_5 is a constant equal to 2, and r_6 and r_7 are random numbers in the range (0,1). If the egg hatches, choose the worst male and female and replace them.

$$x_{w,male} = Lb + r_8 \times (Ub - Lb) \tag{12}$$

$$x_{w,female} = Lb + r_8 \times (Ub - Lb) \tag{13}$$

where $x_{w,male}$ is the worst male snake while $x_{w,female}$ is the worst female snake. r_8 is a random number in the range (0,1). The pseudo-code of SO is given in Figure 1.



Algorithm 1: Snake Optimizer (SO)

```
Initialize Parameters (Dimension, Ub, Lb, Population size(N), Maximum iteration($t_{max}$ ), Current iteration ($t$)
Initialize the population using Equation 1
Divide the population two equal groups $N_{male}$ and $N_{female}$ using Equation 2
while (t ≤ $t_{max}$)
  Find best male snake ( $F_{best,male}$)
  Find best female snake ( $F_{best,female}$)
  Identify $T$ using Equation 3
  Identify food quantity (FQ) using Equation 4
    if (FQ < 0.25) then
      Perform exploration using Equations 5 and  6
    else if (FQ > 0.6) then
      Perform exploitation using Equation 7
    else
        if (rand > 0.6) then
          Snakes in Fighting mode using Equations 8 and 9
        else
          Snakes in Mating mode using Equations 10 and 11
          Change the worst male and female using Equations 12 and 13
        end if
    end if
  end while
  Display best solution.
```

Figure 1. The pseudo-code of SO

2.2. Chaotic maps

The randomness of a mathematically simple deterministic dynamic system is represented by chaotic maps, and the chaotic system can be regarded as a source of randomness (Alataş *et al*. 2007). The convergence ability of SO may depend on random sequences of numbers applied to various parameters during the run of the algorithm. There are studies in the literature showing that the results are very close but not equal when different random sequences are used in metaheuristic algorithms (Bingol and Alatas 2020, Varol Altay and Alatas 2020). Chaotic maps are used to generate chaotic sequences in the process of metaheuristics. The main principle is to apply small chaotic perturbations to the candidate solutions in order to take advantage of the ergodic (i.e., a dynamic system that behaves like the mean) property of chaotic maps to enhance the performance of the solution (Alatas *et al*. 2009, Bingol and Alatas 2020, Wei *et al*. 2019). Equation 14 represents a chaotic map as a dynamical system.

$$x_{r+1} = F(x_r), \quad 0 < x_r < 1, \quad r = 0,1,2,\ldots \tag{14}$$

In this study, it has been investigated whether more efficient results can be obtained from the SO algorithm by using chaotic maps. Table 1 presents the maps generating the chaotic numbers to be used for the SO parameters and their demonstrations.

3. Chaotic Snake Optimizer (CSO)

As mentioned above, the numbers obtained from chaotic maps have been used in many applications, and their effect on performance has been investigated. In this study, we aim to improve the global convergence performance of the algorithm by integrating chaotic maps into the formulas of the search strategies of the standard SO. Chaotic maps can be applied to all random values in the algorithm. However, in this study, the formulas of the search strategies, which are considered to contribute more to performance, are preferred. In algorithms where chaotic maps are used, random numbers are generated by pushing the selected chaotic map one step further. That is, when random number generation is needed from the first iteration on, the selected chaotic map is incrementally advanced starting from the selected starting point.The new CSOs proposed in this study are classified and explained as follows:

- **CSO1:**

CSO1 is obtained by taking the random values (r_1, r_2) in Equations 5 and 6 from the selected chaotic map according to iterations. Accordingly, in the proposed CSO1 algorithm, these equations are replaced by Equations 15 and 16, respectively. Ch_k is the chaotic sequence obtained from the selected chaotic map. The value of k indicates the type of chaotic map, which can be Gauss, Tend, Logistic, Sinusoidal, Circle, Iterative, Sine, or Piecewise.

$$x_{i,j}(t+1) = x_{k,j}(t) \mp c_2 \times A_{i,male}\big((Ub - Lb) \times \boldsymbol{Ch_k(t+1)} + Lb\big) \tag{15}$$

$$x_{i,j}(t+1) = x_{k,j}(t+1) \mp c_2 \times A_{i,female}\big((Ub - Lb) \times \boldsymbol{Ch_k(t+1)} + Lb\big) \tag{16}$$

- **CSO2:**

CSO2 is obtained by taking the random value (r_3) in Equation 7 from the selected chaotic map according to iterations. Accordingly, in the proposed CSO2 algorithm, this equation is replaced by Equation 17.

$$x_{i,j}(t+1) = x_f \mp c_3 \times T \times \boldsymbol{Ch_k(t+1)} \times (x_f - x_{i,j}(t)) \tag{17}$$

- **CSO3:**

CSO3 is obtained by taking the random values (r_4, r_5) in Equations 8 and 9 in fighting mode from the selected chaotic map according to iterations. Accordingly, in the proposed CSO3 algorithm, these equations are replaced by Equations 18 and 19, respectively.

$$x_{i,j}(t+1) = x_{i,j}(t) \pm c_4 \times F_{i,male} \times \boldsymbol{Ch_k(t+1)} \times \big(x_{best,female} - x_{i,male}(t)\big) \tag{18}$$

$$x_{i,j}(t+1) = x_{i,j}(t) \pm c_4 \times F_{i,female} \times \boldsymbol{Ch_k(t+1)} \times \big(x_{best,male} - x_{i,female}(t+1)\big) \tag{19}$$

- **CSO4 :**

CSO4 is obtained by taking the random values (r_6, r_7) in Equations 10 and 11 in mating mode from the selected chaotic map according to iterations. Accordingly, in the proposed CSO4 algorithm, these equations are replaced by Equations 20 and 21, respectively.

$$x_{i,male}(t+1) = x_{i,m}(t) \pm c_5 \times M_{i,male} \times \boldsymbol{Ch_k(t+1)} \times \left(FQ \times x_{i,female} - x_{i,male}(t)\right) \quad (20)$$

$$x_{i,female}(t+1) = x_{i,f}(t) \pm c_5 \times M_{i,female} \times \boldsymbol{Ch_k(t+1)} \times \left(FQ \times x_{i,male} - x_{i,female}(t+1)\right) \quad (21)$$

4. Computational Experiments

In this section, the performance of the proposed chaotic mapping SO variants (CSO1, CSO2, CSO3, and CSO4) is examined. Two test function suites have been chosen for this: the classic test functions and the CEC2019 test functions. The classical test functions used in this study are given in Table 2. These functions consist of a total of twelve functions, eight of which are unimodal and four of which are multimodal. All of these functions are minimization problems, and their optimal values are given in the f_{min} column of the table. The proposed CSO1, CSO2, CSO3, and CSO4 algorithms were tested on these test functions using eight different chaotic maps (Gaussian, Tend, Logistic, Sinusoidal, Circle, Iterative, Sine, and Piecewise). In this test process, the problem dimension is 30, the number of runs is 30, the maximum number of iterations is 1000, and the snake population size is 50. Comparisons in test operations on classical test functions were made by taking into account the mean of the values obtained from 30 independent runs.

Accordingly, in Table 3, the results obtained by using eight different chaotic maps in the CSO1 algorithm are compared with the standard algorithm. In addition to the mean values, the row named "R" gives the order of the results obtained for each function in the table. According to the results in the table, although the best mean values were obtained from different CSO1 variants, $CSO1_{Piecewise}$ found the best mean value in six of the twelve functions. When the MR line showing the mean of the R values obtained for each function was examined, it was seen that the best rank mean was again obtained from $CSO1_{Piecewise}$ and that it was the most successful variant. In addition, all CSO1 variants outperformed the SO algorithm. The standard algorithm, namely SO, took the last place in the MR value ranking. According to the result, it can be said that the performance of the SO algorithm has been improved with the CSO1 algorithm.

Similarly, in Table 4, the results obtained by using eight different chaotic maps in the CSO2 algorithm are compared with the standard algorithm. According to the results in the table, although the best mean values were obtained from different algorithms, SO found the best mean value in six of the twelve functions. When the MR values were examined, it was seen that the best mean rank was again obtained from SO. So, it can be said that the performance of the SO algorithm has not improved with the CSO2 variants. CSO2 variants were ineffective in improving SO performance.

In Table 5, the results obtained by using eight different chaotic maps in the CSO3 algorithm are compared with the standard algorithm. When the results were examined, it was seen that $CSO3_{Piecewise}$ found the best mean value in five of the twelve functions. According to the MR values, $CSO3_{Piecewise}$ took the first place with a mean rank value of 2.58. The worst MR value was obtained from SO. All CSO3 variants are more successful than the standard algorithm. Therefore, the CSO3 approach has improved SO performance.

Table 1. Chaotic maps and their demonstration

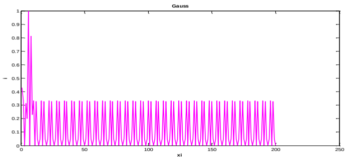
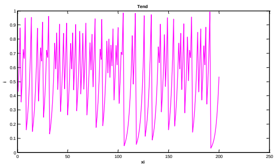
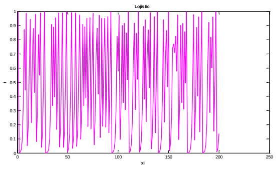
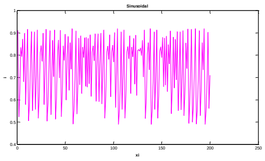
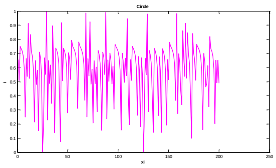
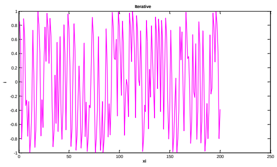
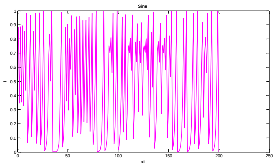
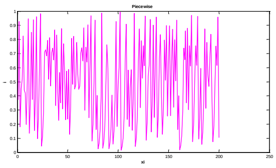
Gauss map $X_{r+1} = \begin{cases} 0 & X_r = 0 \\ \dfrac{1}{X_r mod(1)} & X_r \in (0,1) \end{cases} \quad , \quad \dfrac{1}{X_r mod(1)} = \dfrac{1}{X_r} - \left\lfloor \dfrac{1}{X_r} \right\rfloor$	
Tend map $X_{r+1} = \begin{cases} \dfrac{X_r}{0.7} & X_r < 0.7 \\ \dfrac{10}{3X_r(1-X_r)} & else \end{cases}$	
Lojistic map $X_{r+1} = aX_r(1 - X_r)$	
Sinusoidal map $X_{r+1} = ax_r^2 \sin(\pi x_r)\,, for\ a = 2.3\ and\ X_0 = 0.7 \quad X_{r+1} = \sin(\pi x_r)$	
Circle map $X_{r+1} = X_r + b - \left(\dfrac{a}{2\pi}\right)\sin(2\pi X_r)\, mod(1), \qquad a = 0.5\ \ and\ b = 0.2$	
Iterative map $X_{r+1} = \sin\left(\dfrac{a\pi}{X_r}\right), \quad a = 0.7$	
Sine map $X_{r+1} = \dfrac{a}{4}\sin(\pi x_r)\,, \qquad a = 4$	
Piecewise map $X_{r+1} = \begin{cases} \dfrac{X_r}{Z} & 0 \le X_r < Z \\ \dfrac{X_r - Z}{0.5 - Z} & Z \le X_r < 0.5 \\ \dfrac{1 - Z - X_r}{0.5 - Z} & 0.5 \le X_r < 1 - Z \\ \dfrac{1 - X_r}{Z} & 1 - Z \le X_r < 1 \end{cases} \quad , \qquad Z = 0.4$	

Table 2. Classical test functions

Unimodal Functions	**Range**	$\boldsymbol{f_{min}}$	**Dim**
$f1(x) = \sum_{i=1}^{n} x_i^2$	$[-100{,}100]$	0	30
$f2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10{,}10]$	0	30
$f3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	$[-100{,}100]$	0	30
$f4(x) = max_i\{\lvert x_i\ \rvert, 1 \le i \le n\}$	$[-100{,}100]$	0	30
$f5(x) = \sum_{i=1}^{n-1} [100\ (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30{,}30]$	0	30
$f6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	$[-100{,}100]$	0	30
$f7(x) = \sum_{i=1}^{n} i x_i^4 + rand[0{,}1)$	$[-1.28{,}1.28]$	0	30
$f8 = \sum_{i=1}^{d} -x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500{,}500]$	−418.982 ×dim	30
Multimodal Functions			
$f9 = \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12{,}5.12]$	0	30
$f10(x) = \sum_{i=1}^{n} -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32{,}32]$	0	30
$f11(x) = \frac{1}{4} \times 10^{-3} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}} + 1)$	$[-600{,}600]$	0	30
$f12(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10{,}100{,}4)$ $y_i = \frac{x_i + 5}{4} \quad u(x_i, a, k, m) = \left\{\begin{matrix} k(x_i - a)^m & Eğer\ x_i > a \\ k(-x_i - a)^m & Eğer\ x_i < -a \\ 0 & diğer \end{matrix}\right\}$	$[-50{,}50]$	0	30

In Table 6, the results obtained using eight different chaotic maps in the CSO4 algorithm are compared with the standard algorithm. When the results were examined, it was seen that $\mathrm{CSO4}_{\mathrm{Sinusoidal}}$ found the best mean value in six of the twelve functions. According to the MR values, $\mathrm{CSO4}_{\mathrm{Sinusoidal}}$ took first place with a mean rank value of 2.92. The worst MR value was obtained from SO. All CSO4 variants are more successful than the standard algorithm. Therefore, the CSO4 approach has improved SO performance. Finally, in order to make a general evaluation, the most successful CSO variants were selected and compared. (i.e., $CSO1_{Piecewise}$ from Table 3, SO from Table 4, $CSO3_{Piecewise}$ from Table 5, and $\mathrm{CSO4}_{\mathrm{Sinusoidal}}$ from Table 6). In addition to these, the most successful $CSO3_{Piecewise}$ and $\mathrm{CSO4}_{\mathrm{Sinusoidal}}$ variants in the previous comparisons were run together in the algorithm, and a new $CSO3_{Piecewise} + \mathrm{CSO4}_{\mathrm{Sinusoidal}}$ variant was created. The comparison results for all these variants are given in Table 7.

According to the comparison results given in Table 7, the $CSO3_{Piecewise} + \mathrm{CSO4}_{\mathrm{Sinusoidal}}$ variant created by combining the $CSO3_{Piecewise}$ and $\mathrm{CSO4}_{\mathrm{Sinusoidal}}$ variants with similar performance found the best mean value in seven of the functions. It also ranked first among the algorithms with an MR value of 2.08. The standard algorithm took last place with an MR value of 3.92. According to these results, it was determined that all compared variants improved SO performance, and the most successful variant was $CSO3_{Piecewise} + \mathrm{CSO4}_{\mathrm{Sinusoidal}}$. In addition to these evaluations, the nonparametric Wilcoxon signed rank test (García *et al*. 2009) was applied at the 0.005 significance level to determine whether there was a significant difference between the SO and CSO variants. The results are presented in Table 8. In the table, 'Better', 'Worse', and 'Equal' denote the number of functions for which the CSO variants found better, worse, and equal mean values, respectively, compared to SO. The 'p-value' denotes the level of statistical significance. A p-value less than 0.05 indicates that there is a significant difference between the algorithms; otherwise, there is no significant difference. Accordingly, when the results in Table 8 are examined, it is found that there is a significant difference between the $CSO3_{Piecewise}$ variant and SO, while there is no significant difference between the other variants and SO.

Figure 2 shows the convergence graphics according to the best value obtained by the CSO variants for four randomly selected classical test functions (F1, F4, F9, and F10). When the graphs are analyzed, it is found that the $CSO3_{Piecewise} + \mathrm{CSO4}_{\mathrm{Sinusoidal}}$ variant converges faster in all functions except F9. In the F9 function, the fastest converging variant is $CSO3_{Piecewise}$. The slowest converging variant in all functions is $CSO1_{Piecewise}$. Accordingly, it can be said that the $CSO3_{Piecewise} + \mathrm{CSO4}_{\mathrm{Sinusoidal}}$ variant is generally capable of converging faster to a better or similar value.

Secondly, the performance of the most successful variant was examined on the CEC2019 test functions (Price *et al*. 2018). CEC2019 test functions and features are given in Table 9. The mean and standard deviation values obtained as a result of this test process were compared with the algorithms in the literature (Xu *et al*. 2022). These algorithms are CSA (Hussien *et al*. 2020), BOA (Arora and Singh 2019), MFO (Mirjalili 2015), BA (Yang and He 2013), WOA (Mirjalili and Lewis 2016), SCA (Mirjalili 2016), PSOGSA (Mirjalili and Hashim 2010), AGWO (Qais *et al*. 2018), OBSCA (Abd Elaziz *et al*. 2017), and EGWO (Joshi and Arora 2017) algorithm. The comparison results are given in Table 10. In Table 10, CSO refers to the best-performing $CSO3_{Piecewise} + \mathrm{CSO4}_{\mathrm{Sinusoidal}}$ in previous comparisons. Results for other algorithms are taken directly from the study of Xu *et al*. (Xu et al. 2022). For a fair comparison, the population size is 50 and the maximum iteration is 10,000 in all algorithms. Each algorithm was run independently 30 times, and the mean and standard deviation values were found accordingly.

Table 3. Comparison results for SO and CSO1 variants on classical test functions

	SO	$CSO1_{Gauss}$	$CSO1_{Tend}$	$CSO1_{\text{Logistic}}$	$CSO1_{\text{Sinusoidal}}$	$CSO1_{\text{Circle}}$	$CSO1_{\text{Iterative}}$	$CSO1_{\text{Sine}}$	$CSO1_{Piecewise}$
F1	3.56E-195	2.40E-195	1.10E-195	2.73E-196	5.78E-194	**2.16E-196**	1.40E-195	3.67E-196	2.42E-196
R	8	7	5	3	9	1	6	4	2
F2	2.57E-97	2.45E-98	2.07E-98	9.17E-98	1.26E-98	1.37E-98	3.45E-98	2.28E-98	**1.01E-98**
R	9	6	4	8	2	3	7	5	1
F3	2.54E-128	1.14E-127	1.63E-123	2.93E-127	**2.22E-128**	1.05E-124	5.98E-125	1.06E-125	1.00E-127
R	2	4	9	5	1	8	7	6	3
F4	7.00E-87	7.01E-87	4.24E-87	8.38E-87	1.23E-86	5.58E-87	**2.23E-87**	6.91E-87	6.31E-87
R	6	7	2	8	9	3	1	5	4
F5	1.57E+01	1.52E+01	1.50E+01	1.57E+01	1.52E+01	1.43E+01	1.69E+01	1.46E+01	**1.30E+01**
R	7	5	4	7	5	2	9	3	1
F6	2.61E-02	1.31E-03	1.45E-03	5.25E-03	2.00E-02	1.91E-03	5.65E-03	2.22E-03	**1.10E-03**
R	9	2	3	6	8	4	7	5	1
F7	1.03E-04	9.17E-05	1.09E-04	1.04E-04	9.53E-05	1.21E-04	1.14E-04	1.04E-04	**8.15E-05**
R	4	2	7	5	3	9	8	5	1
F8	-1.25E+04	**-1.26E+04**	**-1.26E+04**	-1.25E+04	**-1.26E+04**	**-1.26E+04**	-1.25E+04	-1.25E+04	**-1.26E+04**
R	2	1	1	2	1	1	2	2	1
F9	4.73E-01	1.28E-01	**9.09E-02**	5.62E-01	9.05E-01	5.98E-01	1.15E+00	4.62E-01	6.70E-01
R	4	2	1	5	8	6	9	3	7
F10	4.09E-15	3.97E-15	3.97E-15	3.97E-15	4.09E-15	3.85E-15	3.85E-15	3.97E-15	**3.61E-15**
R	4	3	3	3	4	2	2	3	1
F11	8.10E-03	3.42E-03	**1.21E-03**	4.73E-03	1.13E-02	1.34E-03	4.22E-03	3.71E-03	1.50E-03
R	8	4	1	7	9	2	6	5	3
F12	2.09E-02	3.56E-02	2.15E-02	2.40E-02	**3.67E-04**	9.56E-03	4.30E-03	9.35E-03	3.22E-02
R	5	9	6	7	1	4	2	3	8
MR	5.67	4.33	3.83	5.50	5.00	3.75	5.50	4.08	**2.75**

Table 4. Comparison results for SO and CSO2 variants on classical test functions

	SO	$CSO2_{Gauss}$	$CSO2_{Tend}$	$CSO2_{\text{Logistic}}$	$CSO2_{\text{Sinusoidal}}$	$CSO2_{\text{Circle}}$	$CSO2_{\text{Iterative}}$	$CSO2_{\text{Sine}}$	$CSO2_{Piecewise}$
F1	**3.56E-195**	2.63E-190	8.34E-191	1.47E-189	9.95E-190	4.77E-189	5.51E-191	1.01E-189	2.12E-190
R	**1**	5	3	8	6	9	2	7	4
F2	**2.57E-97**	4.22E-92	1.96E-92	2.83E-92	2.63E-92	1.45E-92	2.30E-92	6.20E-92	3.53E-92
R	**1**	8	3	6	5	2	4	9	7
F3	**2.54E-128**	3.96E-126	9.10E-127	1.60E-123	1.43E-126	1.66E-126	2.48E-124	7.93E-123	1.70E-126
R	**1**	6	2	8	3	4	7	9	5
F4	7.00E-87	5.60E-87	6.39E-87	3.07E-86	8.94E-87	**3.59E-87**	8.02E-87	4.96E-87	4.35E-86
R	5	3	4	8	7	**1**	6	2	9
F5	1.57E+01	1.41E+01	**1.05E+01**	1.53E+01	1.80E+01	1.58E+01	1.36E+01	1.31E+01	1.47E+01
R	7	4	**1**	6	9	8	3	2	5
F6	**2.61E-02**	7.87E-01	7.67E-01	1.08E+00	6.77E-01	7.45E-01	6.80E-01	9.87E-01	5.63E-01
R	**1**	7	6	9	3	5	4	8	2
F7	1.03E-04	1.11E-04	8.46E-05	1.09E-04	**8.16E-05**	9.93E-05	9.37E-05	9.40E-05	1.06E-04
R	6	9	2	8	**1**	5	3	4	7
F8	-1.25E+04	**-1.26E+04**	**-1.26E+04**	-1.25E+04	-1.25E+04	-1.25E+04	-1.25E+04	**-1.26E+04**	**-1.26E+04**
R	2	**1**	**1**	2	2	2	2	**1**	**1**
F9	**4.73E-01**	2.51E+00	2.25E+00	1.71E+00	9.02E-01	2.62E+00	2.62E+00	8.39E-01	6.18E-01
R	**1**	7	6	5	4	8	8	3	2
F10	**4.09E-15**	4.44E-15	4.44E-15	4.32E-15	**4.09E-15**	4.44E-15	4.32E-15	4.32E-15	**4.09E-15**
R	**1**	3	3	2	**1**	3	2	2	**1**
F11	8.10E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	2.24E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
R	2	**1**	**1**	**1**	**1**	3	**1**	**1**	**1**
F12	2.09E-02	1.89E-02	1.43E-02	2.28E-02	1.18E-02	1.68E-02	**1.00E-02**	1.05E-01	1.07E-02
R	7	6	4	8	3	5	**1**	9	2
MR	**2.92**	5.00	3.00	5.92	3.75	4.58	3.58	4.75	3.83

Table 5. Comparison results for SO and CSO3 variants on classical test functions

	SO	$CSO3_{Gauss}$	$CSO3_{Tend}$	$CSO3_{\mathrm{Logistic}}$	$CSO3_{\mathrm{Sinusoidal}}$	$CSO3_{\mathrm{Circle}}$	$CSO3_{\mathrm{Iterative}}$	$CSO3_{\mathrm{Sine}}$	$CSO3_{Piecewise}$
F1	3.56E-195	3.94E-233	1.69E-243	2.71E-208	**7.39E-278**	1.42E-248	3.31E-174	2.25E-207	3.06E-240
R	8	5	3	6	**1**	2	9	7	4
F2	2.57E-97	8.55E-121	9.97E-126	1.76E-107	**8.28E-139**	1.15E-127	3.83E-92	5.09E-108	7.60E-122
R	8	5	3	7	**1**	2	9	6	4
F3	2.54E-128	9.33E-174	2.59E-186	1.18E-149	**1.46E-224**	4.51E-194	1.96E-120	3.60E-148	1.54E-182
R	8	5	3	6	**1**	2	9	7	4
F4	7.00E-87	8.62E-111	2.91E-111	8.78E-99	**2.84E-127**	2.62E-115	4.75E-83	2.37E-98	5.99E-112
R	8	5	4	6	**1**	2	9	7	3
F5	1.57E+01	1.27E+01	1.87E+01	1.38E+01	1.68E+01	1.92E+01	1.27E+01	1.27E+01	**9.56E+00**
R	6	2	8	5	7	9	2	2	**1**
F6	2.61E-02	2.26E-02	1.34E-02	2.60E-02	**1.22E-02**	1.88E-02	1.81E-02	1.84E-02	1.28E-02
R	9	7	3	8	**1**	6	4	5	2
F7	1.03E-04	8.57E-05	6.45E-05	7.01E-05	7.56E-05	5.78E-05	1.06E-04	9.51E-05	**5.70E-05**
R	8	6	3	4	5	2	9	7	**1**
F8	-1.25E+04	-**1.26E+04**	-**1.26E+04**	**-1.26E+04**	-1.25E+04	-**1.26E+04**	**-1.26E+04**	-**1.26E+04**	**-1.26E+04**
R	2	**1**	**1**	**1**	2	**1**	**1**	**1**	**1**
F9	4.73E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.20E-07	**0.00E+00**	7.66E-01	**0.00E+00**	**0.00E+00**
R	3	**1**	**1**	**1**	2	**1**	4	**1**	**1**
F10	4.09E-15	**8.88E-16**	**8.88E-16**	**8.88E-16**	1.13E-15	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**
R	3	**1**	**1**	**1**	2	**1**	**1**	**1**	**1**
F11	8.10E-03	7.25E-03	1.06E-02	1.13E-02	**5.02E-03**	8.62E-03	1.01E-02	1.25E-02	9.64E-03
R	3	2	7	8	**1**	4	6	9	5
F12	2.09E-02	5.28E-03	4.70E-03	8.36E-03	9.72E-03	8.98E-03	9.22E-03	**2.90E-03**	7.56E-03

R	9	3	2	5	8	6	7	**1**	4
MR	6.25	3.58	3.25	4.83	2.67	3.17	5.83	4.50	**2.58**

Table 6. Comparison results for SO and CSO4 variants on classical test functions

	SO	$CSO4_{Gauss}$	$CSO4_{Tend}$	$CSO4_{\text{Logistic}}$	$CSO4_{\text{Sinusoidal}}$	$CSO4_{\text{Circle}}$	$CSO4_{\text{Iterative}}$	$CSO4_{\text{Sine}}$	$CSO4_{Piecewise}$
F1	3.56E-195	9.33E-246	5.28E-252	7.21E-213	**3.91E-304**	2.22E-259	1.26E-171	5.90E-211	4.50E-249
R	8	5	3	6	**1**	2	9	7	4
F2	2.57E-97	5.21E-124	5.73E-128	1.62E-110	**1.81E-150**	1.32E-130	7.57E-90	1.54E-109	2.41E-125
R	8	5	3	6	**1**	2	9	7	4
F3	2.54E-128	5.17E-196	1.10E-201	1.10E-161	**3.31E-229**	5.52E-214	2.64E-126	3.36E-162	3.92E-197
R	8	5	3	7	**1**	2	9	6	4
F4	7.00E-87	2.38E-115	5.29E-120	4.73E-103	**5.97E-139**	1.16E-123	3.67E-82	4.74E-102	4.67E-118
R	8	5	3	6	**1**	2	9	7	4
F5	1.57E+01	1.46E+01	1.39E+01	1.33E+01	1.37E+01	1.34E+01	1.12E+01	**1.06E+01**	1.19E+01
R	9	8	7	4	6	5	2	**1**	3
F6	2.61E-02	**9.92E-03**	1.43E-02	1.15E-02	1.49E-02	2.11E-02	1.96E-02	2.26E-02	1.82E-02
R	9	**1**	3	2	4	7	6	8	5
F7	1.03E-04	7.94E-05	1.01E-04	8.71E-05	**6.89E-05**	8.27E-05	8.39E-05	1.06E-04	7.14E-05
R	8	3	7	6	**1**	4	5	9	2
F8	-1.25E+04	**-1.26E+04**	**-1.26E+04**	**-1.26E+04**	-1.25E+04	**-1.26E+04**	**-1.26E+04**	**-1.26E+04**	**-1.26E+04**
R	2	**1**	**1**	**1**	2	**1**	**1**	**1**	**1**
F9	4.73E-01	9.97E-01	1.00E+00	2.05E+00	2.37E+00	8.98E-01	7.69E-01	**0.00E+00**	5.23E-06
R	3	6	7	8	9	5	4	**1**	2
F10	4.09E-15	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**	**8.88E-16**
R	2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

F11	8.10E-03	6.78E-03	1.12E-02	1.10E-02	9.63E-03	1.24E-02	**6.71E-03**	8.00E-03	7.83E-03
R	5	2	8	7	6	9	**1**	4	3
F12	2.09E-02	1.22E-02	1.01E-02	2.12E-02	3.93E-03	1.38E-02	**2.82E-03**	2.21E-02	1.42E-02
R	7	4	3	8	2	5	**1**	9	6
MR	6.42	3.83	4.08	5.17	**2.92**	3.75	4.75	5.08	3.25

Table 7. Comparison of successful CSO variants

	SO	$CSO1_{Piecewise}$	$CSO3_{Piecewise}$	$CSO4_{\text{Sinusoidal}}$	$CSO3_{Piecewise}$ $+CSO4_{\text{Sinusoidal}}$
F1	3.56E-195	2.42E-196	3.06E-240	3.91E-304	**0.00E+00**
R	5	4	3	2	**1**
F2	2.57E-97	1.01E-98	7.60E-122	1.81E-150	**6.65E-196**
R	5	4	3	2	**1**
F3	2.54E-128	1.00E-127	1.54E-182	3.31E-229	**0.00E+00**
R	4	5	3	2	**1**
F4	7.00E-87	6.31E-87	5.99E-112	5.97E-139	**3.15E-192**
R	5	4	3	2	**1**
F5	1.57E+01	1.30E+01	**9.56E+00**	1.37E+01	1.71E+01
R	4	2	1	3	5
F6	2.61E-02	**1.10E-03**	1.28E-02	1.49E-02	1.99E-02
R	5	**1**	2	3	4
F7	1.03E-04	8.15E-05	**5.70E-05**	6.89E-05	5.83E-05
R	5	4	**1**	3	2
F8	-1.25E+04	**-1.26E+04**	**-1.26E+04**	-1.25E+04	**-1.26E+04**
R	2	**1**	**1**	2	**1**
F9	4.73E-01	6.70E-01	**0.00E+00**	2.37E+00	**0.00E+00**
R	3	4	**1**	2	**1**
F10	4.09E-15	3.61E-15	**8.88E-16**	**8.88E-16**	**8.88E-16**
R	3	2	**1**	**1**	**1**

F11	8.10E-03	**1.50E-03**	9.64E-03	9.63E-03	2.45E-02
R	2	**1**	4	3	5
F12	2.09E-02	3.22E-02	7.56E-03	**3.93E-03**	6.52E-03
R	4	5	3	**1**	2
MR	3.92	3.08	2.17	2.17	**2.08**

Table 8. Wilcoxon signed-rank test results

Algorithms	Better	Worse	Equal	p-value
SO - $\boldsymbol{CSO1_{Piecewise}}$	9	3	0	0,136097
SO - $\boldsymbol{CSO3_{Piecewise}}$	11	1	0	**0,012063**
SO - $\boldsymbol{CSO4_{\text{Sinusoidal}}}$	9	2	1	0,154860
SO - $\boldsymbol{CSO3_{Piecewise} + CSO4_{\text{Sinusoidal}}}$	10	2	0	0,136097

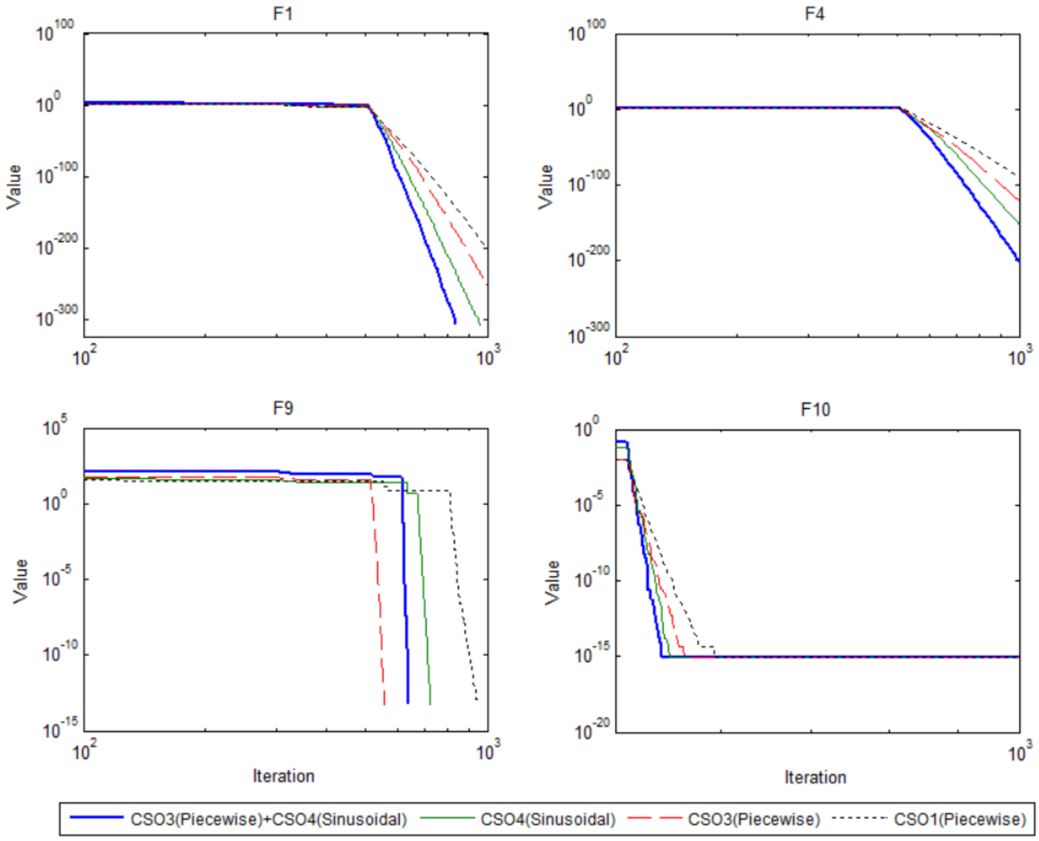


Figure 2. Convergence graphics of CSO variants

Table 9. CEC2019 test function

Functions		$\mathrm{f_{min}}$	Dimension	Range
C1	Storn's Chebyshev Polynomial Fitting Problem	1	9	[-8192, 8192]
C2	Inverse Hilbert Matrix Problem	1	16	[-16384, 16384]
C3	Lennard-Jones Minimum Energy Cluster	1	18	[-4,4]
C4	Rastrigin's Function	1	10	[-100,100]
C5	Griewangk's Function	1	10	[-100,100]
C6	Weierstrass Function	1	10	[-100,100]
C7	Modified Schwefel's Function	1	10	[-100,100]
C8	Expanded Schaffer's F6 Function	1	10	[-100,100]
C9	Happy Cat Function	1	10	[-100,100]
C10	Ackley Function	1	10	[-100,100]

According to the results in Table 10, the proposed CSO in six of the compared functions, OFLCSA in three, and OBSCA in one found the best mean. When the ranking values given in Table 11 are examined, it can be seen that the smallest mean rank (MR) value was obtained by the OFLCSA algorithm as 2.1. The CSO algorithm took second place with an MR value of 2.9. BOA, on the other hand, took last place with a mean rank of 11.8. According to these results, it has been proved that the CSO algorithm has a competitive and successful performance when compared to the algorithms in the literature.

In this study, the effect of using chaotic maps on SO performance is investigated. Successful results are obtained in the tests performed on classical and CEC2019 test functions. In addition, the results obtained have been a guide for researchers as they reveal which chaotic map is used in which search strategy of the algorithm contributes more to the performance. As encountered in other metaheuristic algorithms, a proposed algorithm cannot be expected to excel in all optimization problems. Therefore, some new modifications may be needed if CSO is applied to different test suites or real-world problems.

5. Conclusion and future works

In this study, SO variants using four different chaotic mappings (CSO1, CSO1, CSO3, and CSO4) are proposed by integrating different chaotic maps into the algorithm's search strategy parameters instead of random number sequences to improve SO performance. The performances of these proposed algorithms for eight different chaotic maps were evaluated for classical test functions. According to the test results, the most successful algorithm variants were $CSO1_{Piecewise}$, $CSO3_{Piecewise}$, and $\mathrm{CSO4_{Sinusoidal}}$. The CSO2 algorithms could not contribute to the performance of SO. The most successful $CSO3_{Piecewise} + \mathrm{CSO4_{Sinusoidal}}$ variant was compared with the results of eleven different algorithms selected from the literature for the CEC2019 test functions. In this comparison, the proposed algorithm took second place and proved to be a successful algorithm.

The algorithm proposed as a future work can be applied to different optimization problems, such as engineering problems, large-scale optimization problems, and multiobjective optimization problems.

Table 10. Mean and standard deviation value comparison of CSO with algorithms in the literature on CEC2019

Function		**Algorithms**											
		CSO	OFLCSA	CSA	AGWO	BOA	MFO	BA	WOA	SCA	PSOGSA	OBSCA	EGWO
C1	Mean	2.743E+03	1.621E+04	2.174E+05	1.159E+08	1.961E+11	5.464E+10	6.110E+12	3.286E+05	8.056E+07	1.341E+11	**4.656E−02**	3.902E+06
	Std	1.159E+04	3.002E+04	3.340E+05	4.335E+08	1.566E+11	8.713E+10	3.375E+12	2.955E+09	2.514E+09	2.834E+11	7.678E−02	1.374E+07
C2	Mean	**3.640E+01**	1.028E+06	9.665E+05	4.072E+06	1.535E+08	1.331E+07	2.143E+08	7.471E+07	1.243E+07	1.724E+07	8.257E+04	6.801E+06
	Std	1.214E+02	7.849E+05	4.234E+05	3.518E+06	6.924E+07	1.972E+07	5.880E+07	2.767E+07	8.182E+06	2.254E+07	2.661E+06	3.386E+06
C3	Mean	4.737E+00	**3.819E−01**	3.961E−01	2.052E+04	1.103E+05	4.613E+04	1.100E+05	9.987E−01	6.165E+04	2.342E+04	4.028E−01	5.185E+04
	Std	2.402E+00	1.030E−01	7.474E−02	1.522E+04	4.542E−01	2.254E+04	5.345E−01	1.162E+04	1.546E+04	2.600E+04	3.320E−01	3.140E+04
C4	Mean	**2.104E+01**	1.875E+05	2.500E+05	1.927E+05	1.375E+06	2.549E+05	1.170E+06	4.182E+05	3.213E+05	3.879E+05	3.187E+05	3.861E+05
	Std	7.400E+00	9.515E+04	9.690E+04	3.852E+04	2.091E+05	8.455E+04	2.075E+05	1.577E+05	4.030E+04	1.431E+05	7.141E+04	1.203E+05
C5	Mean	1.581E+00	**1.575E−01**	1.801E−01	1.572E+04	1.572E+06	1.841E−01	1.113E+06	8.418E−01	3.747E+04	3.575E+04	3.781E+04	1.051E+05
	Std	8.257E-02	9.020E−02	1.142E−01	1.558E−01	4.026E+05	2.833E+04	3.521E+05	3.970E−01	1.128E+04	5.020E+04	1.099E+04	1.348E+05
C6	Mean	**6.251E+00**	2.442E+04	2.980E+04	2.720E+04	1.415E+05	3.524E+04	1.241E+05	6.188E+04	4.756E+04	4.951E+04	4.753E+04	5.862E+04
	Std	1.534E+00	1.517E+04	9.540E−01	9.112E−01	8.705E−01	1.526E+04	9.978E−01	1.780E+04	9.904E−01	2.094E+04	9.183E−01	1.605E+04
C7	Mean	**6.354E+02**	6.380E+06	1.024E+07	6.758E+06	2.487E+07	9.483E+06	2.382E+07	1.035E+07	1.083E+07	1.111E+07	9.305E+06	1.054E+07
	Std	3.032E+02	3.088E+06	2.700E+06	2.022E+06	2.501E+06	3.385E+06	2.150E+06	2.641E+06	1.490E+06	3.578E+06	1.981E+06	2.946E+06
C8	Mean	**3.709E+00**	2.353E+04	2.464E+04	2.509E+04	4.359E+04	3.413E+04	4.275E+04	3.276E+04	2.902E+04	3.683E+04	3.284E+04	3.056E+04
	Std	4.842E-01	5.070E−01	2.531E−01	4.513E−01	1.606E−01	3.721E−01	1.816E−01	4.300E−01	2.132E−01	3.660E−01	1.841E−01	4.531E−01
C9	Mean	1.277E+00	**1.401E−01**	1.687E−01	1.705E−01	4.328E+04	3.400E−01	3.604E+04	3.543E−01	3.650E−01	5.224E−01	3.258E−01	2.677E−01
	Std	1.069E-01	5.726E−02	9.405E−02	6.300E−02	7.373E−01	1.542E−01	8.910E−01	1.941E−01	9.542E−02	2.990E−01	7.004E−02	1.205E−01
C10	Mean	**2.131E+01**	5.009E+04	1.100E+05	1.922E+05	2.102E+05	2.014E+05	2.089E+05	2.003E+05	1.965E+05	2.002E+05	1.519E+05	2.005E+05
	Std	4.736E-02	7.002E+04	9.165E+04	2.432E+04	1.361E−01	1.174E−01	1.150E−01	7.832E−02	1.967E+04	8.225E−02	4.194E+04	5.806E−02

Table 11. R-value comparison of CSO with algorithms in the literature on CEC2019

	CSO	OFLCSA	CSA	AGWO	BOA	MFO	BA	WOA	SCA	PSOGSA	OBSCA	EGWO
C1	3	4	5	8	11	9	12	2	7	10	1	6
C2	1	4	3	5	11	8	12	10	7	9	2	6
C3	5	1	2	6	12	8	11	4	10	7	3	9
C4	1	2	4	3	12	5	11	10	7	9	6	8
C5	5	1	2	6	12	3	11	4	8	7	9	10
C6	1	2	4	3	12	5	11	10	7	8	6	9
C7	1	2	6	3	12	5	11	7	9	10	4	8
C8	1	2	3	4	12	9	11	7	5	10	8	6
C9	10	1	2	3	12	6	11	7	8	9	5	4
C10	1	2	3	5	12	10	11	8	6	7	4	9
MR	2.9	**2.1**	3.4	4.6	11.8	6.8	11.2	6.9	7.4	8.6	4.8	7.5

6. References

Abd Elaziz, M., Oliva, D., and Xiong, S., 2017. An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications,* **90**, 484-500.

Alatas, B., Akin, E., and Ozer, A.B., 2009. Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals,* **40**, 1715-1734.

Alataş, B., Akın, E., and Özer, A.B., 2007. Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları. In ELECO 2007 5th International Conference on Electrical and Electronics Engineering, Bursa.

Arora, S., and Singh, S., 2019. Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing,* **23**, 715-734.

Bingol, H., and Alatas, B., 2020. Chaos based optics inspired optimization algorithms as global solution search approach. *Chaos, Solitons & Fractals,* **141**, 110434.

Cheng, K., Zhang, J., Tian, S., Liu, H., Gong, J., and Xie, Y., 2022. WiFi Localization Algorithm Based on Snake Optimization Algorithm to Optimize BP Neural Network. In 2022 International Conference on Image Processing, Computer Vision and Machine Learning , IEEE, 615-618.

Dai, Y., Pang, J., Li, Z., Li, W., Wang, Q., and Li, S., 2022. Modeling of thermal error electric spindle based on KELM ameliorated by snake optimization. *Case Studies in Thermal Engineering,* **40**, 102504.

Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., and Cosar, A., 2019. A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering,* **137**, 106040.

Daliri, A., Asghari, A., Azgomi, H., and Alimoradi, M., 2022. The water optimization algorithm: a novel metaheuristic for solving optimization problems. *Applied Intelligence*, **52**, 17990-18029.

El-Saleh, A. A., Thaher, T., Chantar, H., and Mafarja, M., 2023. Enhanced IoT Based IDS Driven by Binary Snake Optimizer for Feature Selection. In *Advances in Model and Data Engineering in the Digitalization Era: MEDI 2022 Short Papers and DETECT 2022 Workshop Papers, Cairo, Egypt, November 21–24, 2022, Proceedings*,Springer,29-43.

Fu, H., Shi, H., Xu, Y., and Shao, J., 2022. Research on Gas Outburst Prediction Model Based on Multiple Strategy Fusion Improved Snake Optimization Algorithm With Temporal Convolutional Network. *IEEE Access,* **10**, 117973-117984.

García, S., Molina, D., Lozano, M., and Herrera, F. , 2009. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics*, **15**, 617-644.

Gong, Y., Li, C., Wang, F., and Fang, X., 2023. MHCF-CECSO: A Novel High-Performance Clustering Framework for Industrial IoT. *IEEE Internet of Things Journal*, 1-1.

Hashim, F.A., and Hussien, A.G., 2022. Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowledge-Based Systems,* **242**, 108320.

Hussien, A. G., Amin, M., Wang, M., Liang, G., Alsanad, A., Gumaei, A., and Chen, H., 2020. Crow search algorithm: theory, recent advances, and applications. *IEEE Access,* **8**, 173548-173565.

Joshi, H., and Arora, S., 2017. Enhanced grey wolf optimization algorithm for global optimization. *Fundamenta Informaticae,* **153**, 235-264.

Klimov, P.V., Kelly, J., Martinis, J.M., and Neven, H. , 2020. The snake optimizer for learning quantum processor control parameters. *arXiv preprint arXiv,* 2006.04594.

Li, Y., Xiao, L., Tang, B., Liang, L., Lou, Y., Guo, X., and Xue, X., 2022. A denoising method for ship-radiated noise based on optimized variational mode decomposition with snake optimization and dual-threshold criteria of correlation coefficient. *Mathematical Problems in Engineering,* 2022**,** 8024753.

Liu, X., Tian, M., Zhou, J., and Liang, J., 2023. An efficient coverage method for SEMWSNs based on adaptive chaotic Gaussian variant snake optimization algorithm. *Mathematical Biosciences and Engineering,* **20**, 3191-3215.

Mirjalili, S., 2015. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems,* **89**, 228-249.

Mirjalili, S., 2016. SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems,* **96**, 120-133.

Mirjalili, S., and Hashim, S.Z.M., 2010. A new hybrid PSOGSA algorithm for function optimization. In 2010 international conference on computer and information application,Tianjin, China, 374-377.

Mirjalili, S., and Lewis, A., 2016. The whale optimization algorithm. *Advances in engineering software,* **95**, 51-67.

Omran, A. E.-F. A., Nafeh, A. E.-S. A., and Yousef, H. K., 2022. Optimal Sizing of a PV-Battery Stand-Alone Fast Charging Station for Electric Vehicles Using SO. *International Journal of Renewable Energy Research,* **12**, 1769-1778.

Price, K., Awad, N., Ali, M., and Suganthan, P., 2018. Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization. In *Technical Report*: Nanyang Technological University Singapore.

Qais, M.H., Hasanien, H.M., and Alghuwainem, S., 2018. Augmented grey wolf optimizer for grid-connected PMSG-based wind energy conversion systems. *Applied Soft Computing,* **69**, 504-515.

Rawa, M., 2022. Towards Avoiding cascading failures in transmission expansion planning of modern active power systems using hybrid Snake-Sine Cosine optimization algorithm. *Mathematics,* **10**, 1323.

Varol Altay, E., and Alatas, B., 2020. Bird swarm algorithms with chaotic mapping. *Artificial Intelligence Review,* **53**, 1373-1414.

Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., and Zhao, W., 2022. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence,* **114**, 105082.

Wei, X., Yuan, S., and Ye, Y., 2019. Optimizing facility layout planning for reconfigurable manufacturing system based on chaos genetic algorithm. *Production & Manufacturing Research,* **7**, 109-124.

Xu, W., Zhang, R., and Chen, L., 2022. An improved crow search algorithm based on oppositional forgetting learning. *Applied Intelligence*, 1-17.

Vellingiri, M., Rawa, M., Alghamdi, S., Alhussainy, A.A., Althobiti, A. S., Calasan, M., Micev, M., Ali, Z.M., and Abdel Aleem, S.H.E., 2023. Non-Linear Analysis of Novel Equivalent Circuits of Single-Diode Solar Cell

Models with Voltage-Dependent Resistance. *Fractal and Fractional*, **7**, 95.

Yang, X.-S., and He, X., 2013. Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation,* **5**, 141-149.

Yao, L., Yuan, P., Tsai, C.-Y., Zhang, T., Lu, Y., and Ding, S., 2023. ESO: An enhanced snake optimizer for real-world engineering problems. *Expert Systems with Applications*, **230**, 120594.