mathematics

Article

Chaotic Sand Cat Swarm Optimization

Farzad Kiani [1,*], Sajjad Nematzadeh [2], Fateme Aysin Anka [3] and Mine Afacan Findikli [4]

[1] Computer Engineering Department, Faculty of Engineering, Fatih Sultan Mehmet Vakif University, 34445 Istanbul, Turkey
[2] Computer Engineering Department, Faculty of Engineering, Istanbul Topkapi University, 34087 Istanbul, Turkey
[3] Political Sciences and Public Administration Department, Faculty of Economics, Administrative and Social Sciences, Istinye University, 34396 Istanbul, Turkey
[4] Business Administration Department, Faculty of Economics, Administrative and Social Sciences, Istinye University, 34396 Istanbul, Turkey
* Correspondence: farzad.kiyani@gmail.com

Abstract: In this study, a new hybrid metaheuristic algorithm named Chaotic Sand Cat Swarm Optimization (CSCSO) is proposed for constrained and complex optimization problems. This algorithm combines the features of the recently introduced SCSO with the concept of chaos. The basic aim of the proposed algorithm is to integrate the chaos feature of non-recurring locations into SCSO's core search process to improve global search performance and convergence behavior. Thus, randomness in SCSO can be replaced by a chaotic map due to similar randomness features with better statistical and dynamic properties. In addition to these advantages, low search consistency, local optimum trap, inefficiency search, and low population diversity issues are also provided. In the proposed CSCSO, several chaotic maps are implemented for more efficient behavior in the exploration and exploitation phases. Experiments are conducted on a wide variety of well-known test functions to increase the reliability of the results, as well as real-world problems. In this study, the proposed algorithm was applied to a total of 39 functions and multidisciplinary problems. It found 76.3% better responses compared to a best-developed SCSO variant and other chaotic-based metaheuristics tested. This extensive experiment indicates that the CSCSO algorithm excels in providing acceptable results.

Keywords: Chaotic Sand Cat Swarm Optimization; chaotic maps; constrained problems; multidisciplinary problems; hybrid metaheuristics

MSC: 68R12

check for updates

Citation: Kiani, F.; Nematzadeh, S.; Anka, F.A.; Findikli, M.A. Chaotic Sand Cat Swarm Optimization. *Mathematics* **2023**, *11*, 2340. https://doi.org/10.3390/math11102340

Academic Editor: Petr Stodola

Received: 8 April 2023
Revised: 12 May 2023
Accepted: 15 May 2023
Published: 17 May 2023

1. Introduction

In general, the most common and economical process for finding the best value (minimum or maximum) in systems and problems with challenging design is optimization [1,2]. As the size of the problem increases, so does its complexity, and therefore it becomes more difficult to solve [3]. Similar problems are called Nondeterministic Polynomial time (NP-hard) problems [4]. Such problems are common in real-world problems that have various objectives and constraints. Metaheuristic algorithms are the most popular and efficient of the different approaches to solving such problems. These algorithms can be efficient in solving nonlinear and non−differentiable design problems. These algorithms are stochastic-based optimization methods that prove their adequacy to solve many design problems in different fields [5]. Therefore, it is possible to develop different algorithms for various problems. Moreover, according to the No Free Lunch (NFL) [6] theorem, not every algorithm can best solve all problems, so it is important to build up new algorithms. The methods based on this approach do not guarantee that we will find the best solution but that we can try to find the near-best answer and do so with less complexity and a shorter execution time. These algorithms applied in the space area can both avoid local pitfalls and have fast convergence by scanning global and local regions on the basis of randomness. Therefore, recently, many researchers have focused on developing state-of-the-art algorithms to take advantage of this approach [7–9]. In addition to the advantages of the metaheuristic approach described earlier, these algorithms can be easily applied in various fields of science and real-life complex problems [10], as they have relatively simple concepts and do not need derivative function knowledge. Some of the studies can be accessed from References [11–14].

It will be helpful to briefly examine the properties of metaheuristic algorithms for the motivation of the study and the explanation of the main issue. The metaheuristic algorithms consist of two important phases: exploration and exploitation [15]. In the exploration phase, it provides numerous population-based parameters to explore the search space. In the second stage, it is tried to obtain the optimum solution from the existing search space, which can be global or local. Slow convergence and high computation time are unacceptable, although it is by nature not to reach a one-step solution. In these phases, search agents try to seek solutions and catch what they find. In this behavior circulation, the most critical issue is that the processes in these two phases and the transitions between phases are balanced. However, it should be noted that some algorithms can be unstable, converge slowly, and fail to go outside the local sometimes or in some problems. In this case, new strategies can be proposed to solve these limitations and/or improve the performance of current algorithms. Examples of these strategies are parameter tuning, elitism, chaos, and hybrid strategies [5,16]. The concept of chaos, which is one of the most effective approaches, is discussed in this study, and this strategy is planned to be used in the Sand Cat Swarm Optimization (SCSO) algorithm [17]. The performance of this algorithm is degraded by some complex and constrained multidisciplinary problems. Moreover, transitions between exploration and exploitation in the SCSO are sometimes slow; based on this, there may be slow convergence. Briefly, the main gaps of the SCSO algorithm are sometimes the problems of low search consistency, local optimum trap, inefficiency search, and low population diversity. Accordingly, it is planned to eliminate these problems with a chaos strategy. The concept of chaos has been applied extensively in various applications with the growth of nonlinear dynamical systems that are highly sensitive to the initial state. Chaos-based algorithms can generate a large number of different search points in a short time, which can help explore the optimization area more efficiently and quickly than traditional optimization algorithms.

Metaheuristic algorithms try to be effective in various engineering optimization processes by using chaotic maps based on the concept of chaos with random and regular features. In this study, a new hybrid algorithm is proposed based on the chaos strategy. According to [18,19], as the initial population diversity increases, it becomes possible for the algorithm to escape from the local optimum trap and prevent premature convergence. On the other hand, in another study [20], it was found that the application of the chaotic component in optimization is a performance-enhancing factor in many algorithms.

This paper proposes a novel Chaotic Sand Cat Swarm Optimization (CSCSO) algorithm. This chaotic-based algorithm seeks to follow global optimum solutions with better convergence by using various chaotic maps to improve SCSO performance. In addition to overcoming some of the limitations of SCSO, the proposed algorithm aims to determine the most suitable chaotic plots for the SCSO algorithm. In this regard, 12 maps are integrated to develop exploratory and exploitative mechanisms. Some benchmark test functions and constrained problems are investigated to evaluate the performance of the CSCSO algorithm. It is analyzed for all maps one by one. Finally, the outcomes of the proposed algorithm are compared with several well-known algorithms in the literature. Besides these, the main contributions of this paper are summarized below:

(1) It is to integrate the chaos feature of non-recurring locations into SCSO's core search process to improve global search performance.
(2) The problem of slow transitions between phases and early or late convergence is decreased.
(3) It focuses on exploration as early as possible to avoid falling into local optimization.

(4) It tries to find the best or near−optimum solutions by increasing the probability of the population spread.
(5) By using 12 different maps, the most suitable map for SCSO is determined. Therefore, it is a chaos-based inclusive algorithm for SCSO.
(6) The method is flexible and can be easily adapted to different types of optimization problems.

The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 overviews the standard sand cat swarm algorithm. Section 4 introduces a novel hybrid chaos-based algorithm (CSCSO). Sections 5–7 include the outcomes and analysis of the proposed algorithm in various cases and problems. The conclusion takes place in the Section 8.

2. Related Works

Since the study mainly focuses on metaheuristic algorithms of the concept of chaos, this section examines chaotic and hybrid methods with other metaheuristic algorithms. Then, in the next section, the SCSO algorithm and its features are discussed. Therefore, in this section, the analysis of chaos-based metaheuristic methods is mainly discussed. However, first, let us briefly describe the types of metaheuristic algorithms.

The metaheuristic algorithms are broadly divided into four main categories: evolutionary, physics-based, human behavior, and swarm intelligence algorithms [3]. It is worth noting that there are also hybrid methods consisting of these four main categories. In evolution-based algorithms, the biological behavior of different systems is taken into account. One of the famous algorithms in this category is the Genetic Algorithm (GA) [21]; it is based on Darwin's theory. Among the studies in this category, Refs. [22–24] are recent studies that can be given as examples. Physics-based algorithms are the category that exhibits random behavior inspired by the laws of physics in nature. Some of the studies in this category are presented in [25–27]. Algorithms in the third category are those inspired by the social behavior of humans. Some studies can be cited as examples in this category [28–30]. This category is expected to become widespread by incorporating more and more social sciences in the future [3]. In particular, it should be emphasized that there are multivariate dynamic problems in social sciences, and in solving these problems, these algorithms expected to be used frequently, such as artificial intelligence and machine learning [31]. The last category is Swarm Intelligence (SI) algorithms, which have received a lot of attention recently by researchers. The SI is also defined as the collective behavior of a decentralized or self−organizing system [32]. This approach consists of a large number of members with limited intelligence who interact with each other based on simple principles. Many studies have been performed in this category [32–35]. The hybrid algorithms can be presented for more efficient solutions to some global and/or specific problems. Considering that there are difficult and complex problems faced in our real world, it is inevitable that such algorithms will become widespread. In accordance with this purpose, they are looking for better solutions by combining the pros of the metaheuristic algorithms under consideration. Hybrid methods generally either present different existing metaheuristic algorithms as a single new algorithm or make improvements to existing algorithms. Recently, out−of−the−box hybrid models realize the concept of chaos by adapting them to metaheuristic algorithms. Some examples are listed in [36–38]. A generalized version of these classifications is presented in Figure 1. Its wide-ranging metaheuristic approach is used today for a variety of real problems, ranging from engineering to intelligent systems [39–43]. More examples of studies in these categories [44–65] are referred to in this figure.

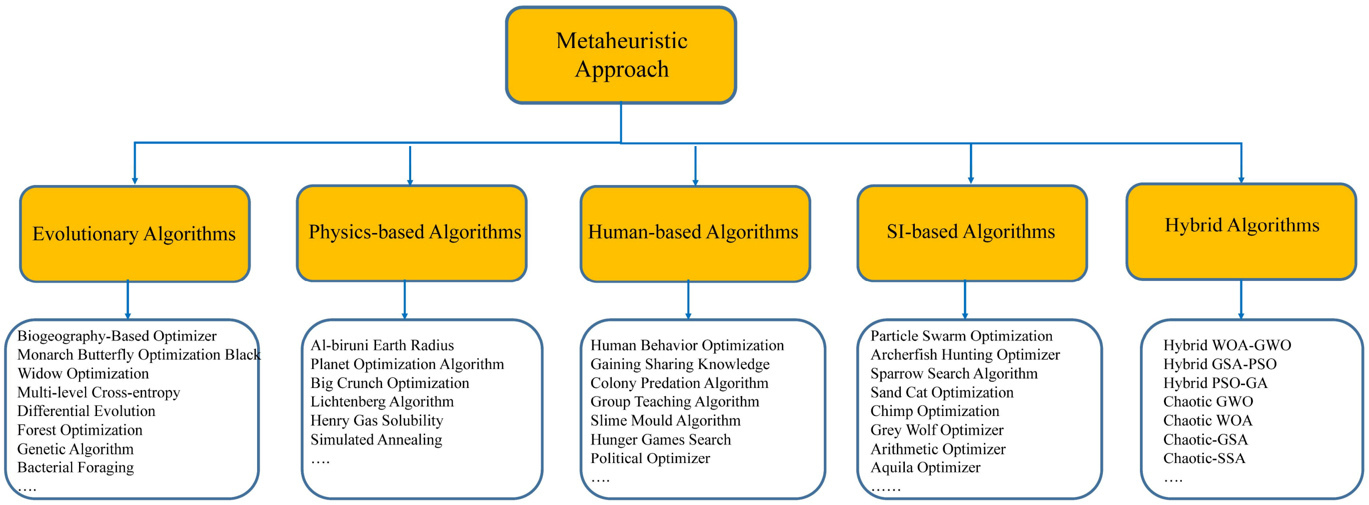


Figure 1. Generalized classification of metaheuristic algorithms [3,21–65].

As mentioned earlier, chaos-based metaheuristic methods are explored in this section. In this regard, the focus is on the well-known population-based SI algorithms as far as possible. In [66], a hybrid chaos-based algorithm was proposed, called Broyden–Fletcher–Goldfarb–Shanno algorithm (Chaos−BFGS). BFGS is a quasi−Newton method for local optimization devised. Methods based on Newton's model have a fast convergence rate and high efficiency, while optimization results are based on selected initial points. The authors introduced pseudo-randomness and disorder by adding chaotic behavior to the relevant algorithm. In [67], researchers proposed a new metaheuristic based on chaotic strategies to improve the performance of power distribution systems. In this algorithm, called Modified Symbiotic Organisms Search (MSOS), they tried to solve the constraints of the economic dispatch system of the relevant system. In this study, they helped the algorithm to find a global optimum solution with a superior convergence rate by applying different logistic chaos maps. Similarly, in [68], the researchers were able to significantly improve the performance of the Big Bang–Big Crunch (BBBC) [69] algorithm with three different chaos maps and five unique chaotic-based strategies.

In [70], the author tried to improve the performance of the Cuckoo Search Algorithm (CSA) by incorporating ten chaotic maps. They claimed to improve the performance of their algorithm in terms of quality solutions and convergence behaviors, based on their results in 27 benchmarking problems. In another study [36], ten specialized chaotic maps were applied to the Grey Wolf Optimization (GWO) algorithm. The authors claim that the algorithm they propose has acceptable performance in the global optimum finding and convergence rate for constrained problems, based on their results. In this regard, the results were compared with the standard GWO. In a study [71], the authors used augmenting chaotic maps for improving the performance of the Krill Herd Optimizer (KHO) [72] in terms of computational time and convergence rate. This conclusion was reached after encountering the standard KHO and a few other algorithms. In another study [73], chaos theory was used to find the local optimum solution and solve slow convergence problems of the GA. In this study, the proposed chaotic GA demonstrated successful performance in the optimum design of critical hydroelectric systems.

In another study [74], the failure of the Dolphin Swarm Algorithm (DSA) [75] in some cases, such as incomplete solution and entrapment in local optima, was discussed. To solve these problems, the authors augmented eight chaotic logistic maps. Their outcomes have shown a significant improvement. The authors claim that their proposed algorithm achieved improvements in the convergence rate, along with the elimination of the above problems, by comparing their results with the standard DSA. The Chaos Ant Colony Algorithm (CACA) was proposed in [76]. In this study, efficient tool path, motion, and handling were estimated. The results show that pocket milling can be optimized with effective tool trajectories with the help of CACA. The last study discussed in this section is a chaos structure applied to the Artificial Bee Colony (ABC) [77]. According to the analyses of the authors, they claimed that the chaotic ABC is superior to the ABC in reaching the global optima and the rate of convergence. In addition, some chaos-based studies, albeit limited, focus on phase transitions, convergence speed, computation time, and quality results [78–81].

In this study, we focused on improving the performance of the SCSO algorithm, especially in solving complex and constrained problems, as mentioned in the previous section. In this context, a new chaos-based algorithm is proposed that converges fast, does not delay transitions between phases, and does not fall into the local optima trap. A detailed explanation of this algorithm will be given in the following sections. Summaries and analyses of some studies from the literature are presented in Table 1.

Table 1. Summary of some current studies.

Algorithms	Strong Points	Weak Points	Focus on
[17]	+ Easy to implement and can handle high dimensional and non−linear optimization problems. + Try to find a suitable solution in a reasonable amount of time. + Balance behavior in phases of transition.	− Sensitive to the choice of parameters. − Possibility of early or late convergence. − Being dependent on the initial population form.	Engineering and global problems.
[36]	+ Fast convergence. + Increasing the probability of population spread. + Successful performance in finding the optimal solutions.	− Using a limited number of chaotic maps. − High complexity. − Tunning based on randomness approach.	Constrained benchmark functions and some engineering design.
[62]	+ Strong global searchability. + Balance between phases.	− Long execution time. − Just using a chaotic map.	A limited number of engineering problems (ANN-based problems).
[66]	+ Fast convergence rate and high efficiency. + Generating a very diverse population initially.	− Possibility of premature convergence. − Utilizing a limited number of chaotic maps.	Numerical optimization and engineering design problems.
[67]	+ Balancing transitions between phases. + Performing a symbiotic relationship between solutions to increase search efficiency and promote diversity in the solution space. + Use of valve−point effects.	− A limited number of chaotic maps are used. − Possibility of early convergence.	Several thermal units' problems.

Table 1. *Cont.*

Algorithms	Strong Points	Weak Points	Focus on
[68]	+ Good convergence rate. + It generates a large number of different search points in a short time.	– Just using a single chaotic map. – Possibility of early convergence. – Being prone to premature convergence.	A specific engineering problem.
[69]	+ Acceptable quality solutions and convergence behavior. + Improvement of the cuckoo search algorithm. + Employing various chaotic maps + Tuning the step size of the cuckoos	– Insufficient exploration of the search space sometimes. – Does not work well in high−dimensional functions.	Some classic benchmarking and engineering problems.
[70]	+ Having a reasonable rate of global convergence. + Strong robustness compared to Krill Herd Algorithm. + Using several chaotic maps.	– Tunning parameters. – Poor efficiency in high−dimensional functions. – Few comparisons.	Some classic benchmarks and a gear train design problem.
[72]	+ Solving slow convergence problems of the Genetic Algorithm. + Supporting multiple objectives. + Faster convergence. + High quality solutions.	– Limited use for real−time applications. – Requiring a large amount of computational resources. – Because it focuses on a particular problem, it depends on data and circumstances and may not always produce current or accurate results.	A limited number of engineering problems.
[73]	+ Using various chaotic maps. + Improvement of Dolphin Swarm Algorithm by chaos theory. + Trying to solve high−dimensional functions. + Trying to prevent low optimization accuracy.	– Low convergence precision. – Low solving efficiency. – May not be sufficient for complex and constraint problems.	Some high−dimensional benchmark problems.
[82]	+ Population diversity. + Good convergence.	– Slow convergence rate. – Just using one chaotic map.	Some benchmark problems. Economic-based problem.Energy optimization in microgrid.
[83]	+ Fast convergence.	– Tuning dependent. – Long execution time.	Complex function optimization problems.
[84]	+ Fast convergence.	– Tuning dependent. – Long execution time.	Solves engineering and dynamic problems.

3. Overview of Sand Cat Swarm Optimization

The Sand Cat Swarm Optimization (SCSO) is a new SI-based metaheuristic algorithm [17]. Based on the special hearing and hunting abilities of these cats in the desert, this algorithm can respond quickly and perform exploration-exploitation processes in a balanced way. The sand cats, who hunt mainly at night, have interesting hearing abilities. In this way, these cats meet at least 10% more food needs than normal cats. Moreover, these cats can travel long distances without rest, increasing the likelihood of better response in final iterations. These cats can track prey movement and location more precisely with these unique features. Based on the behavioral characteristics of sand cats, their foraging consists of two general stages. One of them is to seek prey, and the other is to attack the prey. This easily allows the algorithm to behave in both the exploration and exploitation phases.

In this population-based algorithm, a sand cat is assigned for the uncertain parameter to be found in each problem. Each cat (search agent) is considered a vector, and the length of this vector is equivalent to the size of the problem. The performance measurement of the algorithm is based on the fitness function of each problem (Equation (1)).

$$Fitness = f(Sand\ Cat) = f(SC_1, SC_2, \ldots, SC_n);\ \forall x_i\ (is\ calculated\ for\ t\ time) \tag{1}$$

The mathematical models that are effective in the seeking (exploration) and hunting (exploitation) phases in the SCSO are given below (Equations (2)–(5)). Thanks to these, it is ensured that each cat moves toward the prey and therefore approaches the target (the solution to the problem).

$$\vec{c} = S - \left(\frac{S * t}{T}\right) \tag{2}$$

$$\vec{R} = 2 \times \vec{c} \times rand - \vec{c} \tag{3}$$

$$\vec{r} = \vec{c} \times rand \tag{4}$$

R and r are two coefficients that play a critical role in both phases. R controls the algorithm to behave balanced between the two phases, and r is inspired by the hearing sensitivity of cats. C is a parameter that decreases linearly from 2 to 0 as iterations progress. As it turns out, parameter C affects both R and r. The S is a constant whose value is assumed to be 2 inspired by standard SCSO [17]. However, it is made possible to assign different values. Thanks to this flexibility, different integer values for S can be assigned according to the need of different problems. The t represents the current iteration, and T indicates the maximum number of iterations. In general, the behavior model of the SCSO algorithm in position updating is presented in Equation (5), both in the exploitation phase and in the exploration phase:

$$\vec{X}(t+1) = \begin{cases} \vec{X}_b(t) - \vec{X}_{rnd} \cdot \cos(\theta) . \vec{r} & |R| \leq 1\,; exploitation\ (a) \\ \vec{r} . \left(\vec{X}_c(t) - rand \cdot \vec{X}_k(t)\right) & |R| > 1\,; exploration\ (b) \end{cases} \tag{5}$$

where X_k is the current position of each agent, X_c represents the best candidate position, X_{rand} points to a random position, and X_b represents the position of the global best agent. Theta (θ) used in a cosine plays a role in bringing the cat closer to the prey. When the condition $|R| \leq 1$ is met, the search agents are directed to attack (exploit); otherwise, the cats are tasked with finding a new possible solution in the global area. This value is selected based on Roulette Wheel. The pseudocode of the SCSO algorithm is presented in Algorithm 1.

Algorithm 1. Standard SCSO pseudocode

```
Initialize the population.
Calculate the fitness function based on the objective function.
Obtain a random angle.
Initialize the $c$, $R$, $r$ based on the Equations (2)–(4)
While ($t$ <= maximum iteration)
    For each search agent
        If(abs($R$) <= 1)
            Update the search agent position based on the Equation (5a).
        Else
            Update the search agent position based on the Equation (5b).
        End
    End
  t = t++
End
```

4. CSCSO: Chaotic Sand Cat Swarm Optimization Algorithm

In general, metaheuristic methods are recommended to reduce execution time and computation costs, because realistic engineering or design problems are difficult and complex. However, they may sometimes face problems such as early convergence, low search consistency, local optimum trap, inefficiency search, and low population diversity and may deviate from the optimal solution. Some of these problems are also present in the SCSO algorithm. To overcome these difficulties, a new hybrid algorithm with chaotic maps is proposed in this research. These maps can have advantages such as escaping from the local area and speeding up the search process with their dynamic nature. So, although SCSO has a balanced and reasonable convergence rate, it may not always perform well in finding the global optimum that affects the convergence rate. In addition, it is likely that predators will be blind due to the random operating mechanism and therefore offer a limited rate of exploitation in the search space. Therefore, the chaos concept is introduced into the proposed algorithm, called CSCSO, to reduce this deficiency and increase general efficiency.

In general terms, chaos is a deterministic, random-like method found in a nonlinear, dynamic, period−less, non−converging, and finite system. It is also sensitive to the initial values. Chaos means the property of a complex system whose behavior is very unpredictable. In mathematical terms, chaos describes the randomness of a simple deterministic dynamical system, and from this, chaotic systems can be thought of as sources of randomness. In this regard, various chaotic maps with different mathematical equations are used to introduce chaos into the proposed algorithm. The chaotic mapping ensures a uniform distribution of the population. Chaos maps seek to match or relate the behavior of chaos in the optimization algorithm based on some parameter using a function. In addition, the search space can be scanned more dynamically and globally with the help of these maps, and therefore, dynamic behaviors are gained for the algorithm. These are maps that exhibit complex and dynamic behavior in nonlinear systems. Briefly, it can be used as an alternative to pseudo-random number generators in the search field and often yields better results than pseudo-random numbers. In the global optimization process, the chaos-based optimization algorithm uses chaotic sequences, which are mapped from chaotic maps, to produce design variables instead of random sequences [85,86].

The main idea of the proposed CSCSO is to integrate the chaos feature of non-recurring locations into the SCSO's core search process to improve global search performance. That is, the proposed algorithm combines SCSO with the concept of chaos. The CSCSO algorithm based on probability distributions and random behavior can be advantageous when using the chaotic concept. That is, the proposed algorithm is aimed to improve the convergence rate by using chaotic maps. It also escapes local traps more easily than classical stochastic methods. So, randomness in the SCSO can be replaced by a chaotic map due to similar randomness features with better statistical and dynamic properties. These maps entice chaos in the favorable region, which is predicted for only a very short initial time, and stochastic for a longer period. Some various well-recognized chaotic maps with different mathematical equations, which are listed in Table 2, are used to add chaos to the proposed algorithm. In the CSCSO algorithm, 12 chaotic maps are applied to tune the step size of the standard SCSO algorithm. It increases the probability of the population spread and tries to achieve more robust and balanced solutions.

Table 2. Popular chaotic maps used in the CSCSO algorithm.

No.	Name	Chaotic Map	Range
1	Chebyshev	$\mathrm{X}_{i+1} = \cos\left(icos^{-1}x_i\right)$	$(-1, 1)$
2	Circle	$\mathrm{X}_{i+1} = \mathrm{mod}\left(x_i + \mathrm{b} - \left(\frac{a}{2\pi}\right)\sin(2\pi x_i), 1\right),\ \mathrm{a} = 0.5,\ \mathrm{b} = 0.2$	$(0, 1)$
3	Iterative	$\mathrm{X}_{i+1} = \sin\left(\frac{a\pi}{x_i}\right),\ a = 0.7$	$(-1, 1)$
4	Piecewise	$\mathrm{X}_{i+1} = \begin{cases} \frac{x_i}{P} & 0 \le x_i < P \\ \frac{x_i - P}{0.5 - P} & P \le x_i < 0.5 \\ \frac{1 - P - x_i}{0.5 - P} & 0.5 \le x_i < 1 - P \\ \frac{1 - x_i}{P} & 1 - P \le x_i < 1 \end{cases}\quad P = 0.4$	$(0, 1)$
5	Sinusoidal	$\mathrm{X}_{i+1} = \mathrm{a}X_i{}^2\sin(\pi x_i),\ a = 2.3$	$(0, 1)$
6	Sine	$\mathrm{X}_{i+1} = \frac{\mathrm{a}}{4}\sin(\pi x_i),\ a = 4$	$(0, 1)$
7	Singer	$\mathrm{X}_{i+1} = \mu\left(7.86X_i - 23.31X_i{}^2 + 28.75X_i{}^3 - 13.302875X_i{}^4\right),\ \mu = 1.07$	$(0, 1)$
8	Gauss/Mouse	$\mathrm{X}_{i+1} = \begin{cases} 1 & \mathrm{X}_i = 0 \\ \frac{1}{mod(\mathrm{X}_i, 1)}, & otherwise \end{cases}$	$(0, 1)$
9	Logistic	$\mathrm{X}_{i+1} = \mathrm{a}\mathrm{X}_i(1 - \mathrm{X}_i),\ a = 4$	$(0, 1)$
10	Tent	$\mathrm{X}_{i+1} = \begin{cases} \frac{\mathrm{X}_i}{0.7} & \mathrm{X}_i < 0.7 \\ \frac{10}{3}(1 - \mathrm{X}_i), & \mathrm{X}_i \ge 0.7 \end{cases}$	$(0, 1)$
11	Bernoulli	$\mathrm{X}_{i+1} = 2\mathrm{X}_i(mod\ 1)$	$(0, 1)$
12	Quadratic	$\mathrm{X}_{i+1} = \mathrm{X}_i^2 + c,\ c = [0, 2]$	$(0, 1)$

After finishing the main reproduction process of the SCSO, the chaos map is used to update the newly generated position from SCSO search procedures based on Table 2. It should be noted that the initial value can have significant effects on the fluctuation pattern of some chaotic maps. In this study, the initial value for all maps was accepted as 0.7, inspired by [34,78]. The virtualization of the chaotic maps used in this study is presented in Figure 2. These chaotic maps are deterministic processes that also have random behavior.

Some special advantages of the SCSO algorithm cannot be overlooked, such as balanced behavior between exploration and exploitation phases, and finding suitable solutions with fewer parameters and operations. On the other hand, the pros of the concept of chaotic should also be taken into account. This chaotic behavior, especially in the final stages, helps sand cats to fall into the local optimum trap and solve high−dimensional problems with fast convergence. These chaotic maps are deterministic processes with random behavior. On the other hand, random behavior in the SCSO is greater by its very nature. In this context, the chaotic map will be useful for initializing the population of sand cats to obtain a better initial solution and increase the convergence precision. Therefore, this study plans to increase the performance of the SCSO in solving difficult problems, especially constrained multidisciplinary problems with this concept. The proposed algorithm works with a multi-strategy mechanism based on a hybrid approach. In this regard, the CSCSO algorithm introduces two alternative solutions for position updating in phases. One of them is the normal position update mechanism, and the other is based on the chaotic model [34,63]. Each of them is given a 50% chance to have a balanced weight (Equation (6)):

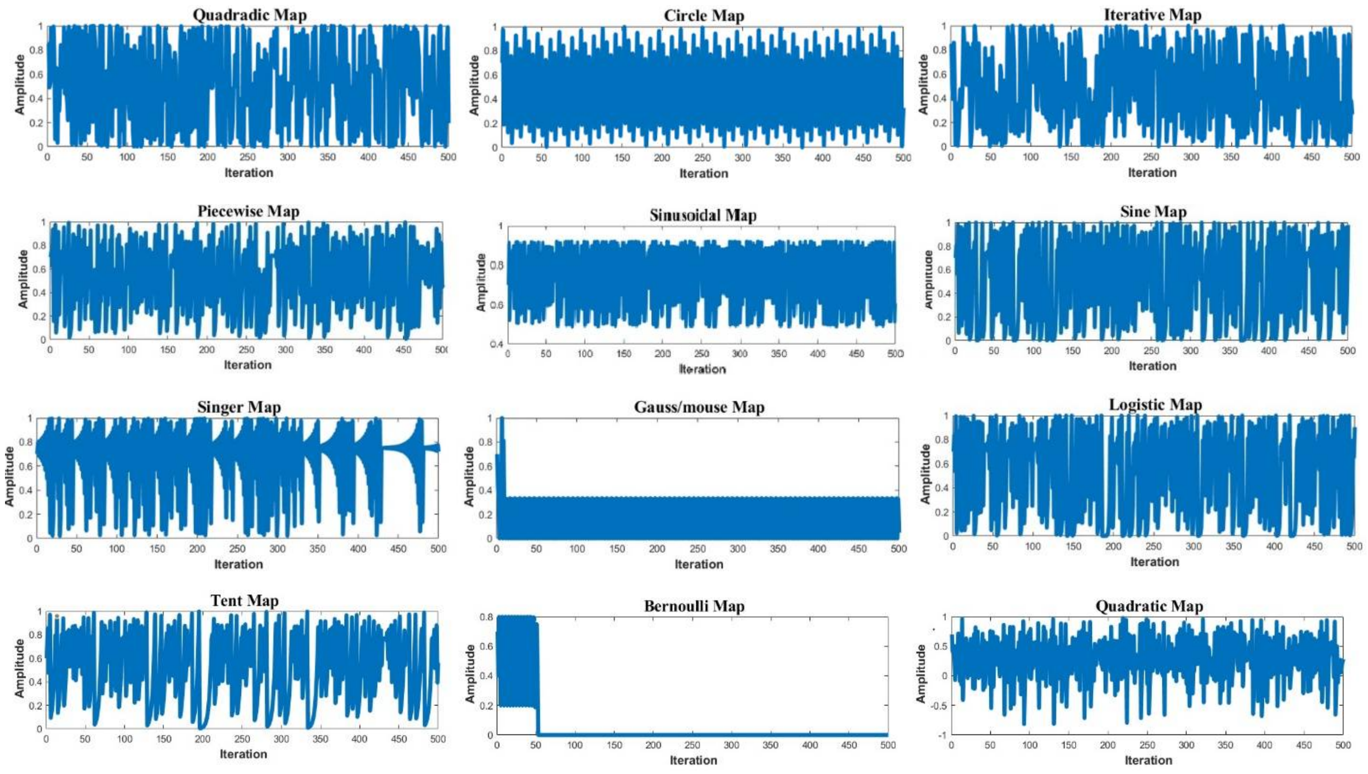


Figure 2. The chaotic maps used in the study.

$$\vec{X}(t+1) = \begin{cases} if\ p < 0.5 & \equiv Eq.5 \\ if\ p \geq 0.5 & \equiv Eq.10 \end{cases} \tag{6}$$

where p is a random value between 0 and 1. The behavior model of each map is presented in Figure 2. The most important parameter in performing a similar task in the SCSO algorithm is the C parameter. This parameter plays a direct role in the R parameter. The equation that finds C (Equation (2)) plays an important role in exhibiting a very balanced and fair behavior for exploration and exploitation but may face slow convergence and unsuccessful exploitation, especially in complex and constrained problems. Accordingly, parameter C is defined for use in the models in Equation (7). Here, it is attempted to improve the functionality of the exploration and exploration phases.

$$s = 2^k;\ k \geq 1$$

$$C = s - s\left(\frac{\sqrt[T]{e^t} - 1}{e - 1}\right) \tag{7}$$

where K is a constant coefficient. This parameter (K) plays an important role in how much weight the algorithm gives to each stage. The exploration phase is given more chances with respect to this equation. For example, assuming k = 2, the algorithm is given more than a 60% chance for exploration. According to the experiments and studies, focusing more on the exploration phase in the CSCSO algorithm yields good results in constrained problems. The C parameter is important in making the CSCSO algorithm more flexible because it plays a one−to−one role in the exploration and exploitation phases. According to this new equation, the parameters R and r are also affected. Since the R parameter depends on C, its fluctuation range is also decreased. R is a random value in the interval [$-2C$, $2C$]. The CSCSO algorithm forces search agents to exploit when R is less than or equal to 1; otherwise, search agents are forced to explore and find prey. This is used when the value of p is less than 0.5. Otherwise, chaotic maps are used. Thanks to this hybrid mechanism, local optimum trapping and premature convergence are avoided. The R and r parameters expected to be obtained by the effect of chaotic maps are calculated according to Equations (8) and (9).

$$\vec{R} = 2 \times \vec{c} \times m - \vec{c} \tag{8}$$

$$\vec{r} = \vec{c} \times m \tag{9}$$

$$\vec{X}(t+1) = \begin{cases} \vec{X_b}(t) - \overrightarrow{X_{rnd}} \cdot \cos(\theta) . \vec{r} & |R| \leq 1 \ (a) \\ \vec{r} . \left(\vec{X_c}(t) - m \cdot \vec{X_t}(t) \right) & |R| > 1 \ (b) \end{cases} \tag{10}$$

The m is a chaotic vector that is calculated based on a chaotic map. The general mathematical model of CSCSO is calculated according to Equation (10), as mentioned before. In various problems, especially constrained optimization problems, it will find other possible local areas in global space with a fast and accurate convergence rate due to the behavior of the proposed algorithm. The proposed CSCSO algorithm improves the performance of the SCSO algorithm while being equal to the original SCSO in terms of complexity analysis. It is useful to remind the reader that the mathematical model of the proposed algorithm is given in Equation (6) as a summary. In addition, the pseudocode and flowchart of the CSCSO are given in Algorithm 2 and Figure 3, respectively.

Algorithm 2. Pseudocode of proposed CSCSO

```
Initialize the population.
Calculate the fitness function based on the objective function to find the best search agent: $X_{best}$.
Obtain a random angle.
Initialize $p$.
Initialize the $r$, $c$, and $R$.
Initialize the value of the chaotic map.
While ($t$ <= Max_Iteration)
    Update the chaotic value using the respective chaotic map.
    For each sand cat (search agent)
        If ($p$ < 0.5)
            Update the search agent position based on the Equation (5)
        Else
            Update the search agent position based on the Equation (10).
        End If
        Update $r$, $R$, and $p$.
    End for
Check if any search agent goes beyond the search space and amend it.
Calculate the fitness of each search agent.
Update $X_{best}$ if there is a better solution.
    t = t++
End while
return $X_{best}$.
```

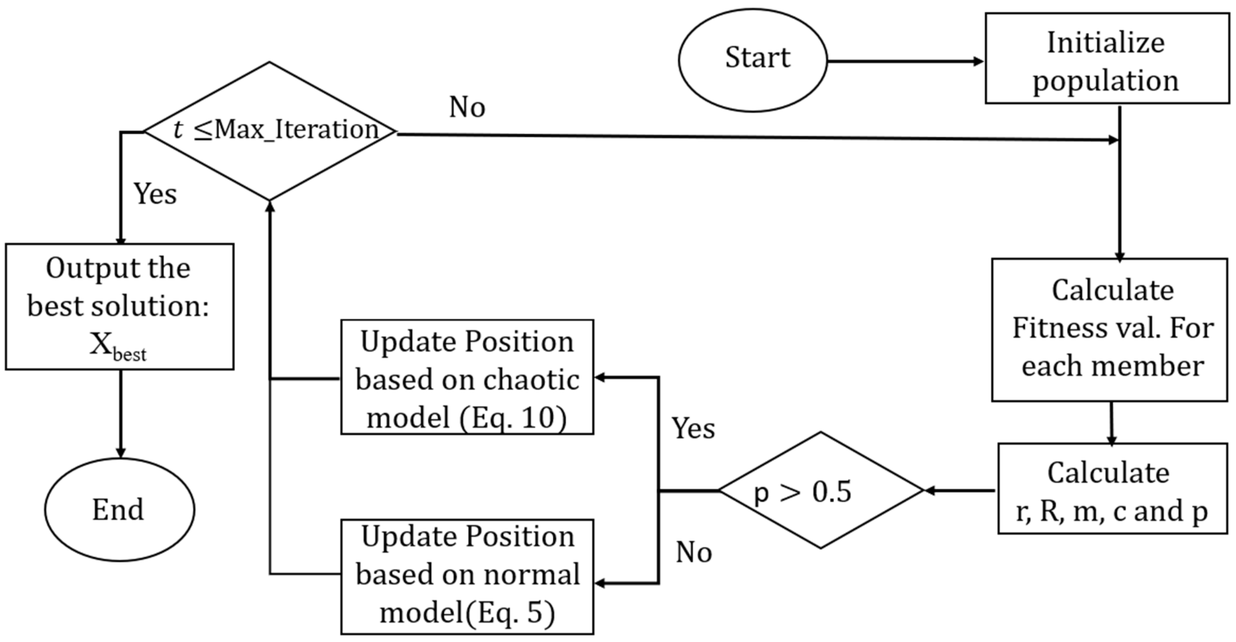

Start
Initialize population
t ≤Max_Iteration
No
Yes
Output the best solution: X_{best}
End
Calculate Fitness val. For each member
Update Position based on chaotic model (Eq. 10)
Yes
p > 0.5
Calculate r, R, m, c and p
Update Position based on normal model(Eq. 5)
No

Figure 3. The flowchart of the proposed algorithm.

5. Simulation Experiments

5.1. Experiments' Settings

In this study, comprehensive problem types are discussed. This section focuses on 16 benchmarking functions (CEC2015 and CEC2016) [25,87,88], 8 competitive functions (CEC2019) [89], 8 real-world optimization problems (CEC2020) [5,90], 5 constrained engineering problems [91,92], and 2 constrained social sciences-based problems [93] to measure the performance of the proposed algorithm (CSCSO). These famous and well-known sample problems from different types were selected to evaluate the performance of the proposed algorithm. In addition, the proposed algorithm is simulated with 12 chaotic maps, and all results are compared with the best SCSO variant algorithm to evaluate and discuss in a wide range [3]. Among the standard SCSO and its improvements in the literature [94–96], it was proven that the best performance belongs to the Stochastic variation and Elite collaboration in SCSO (SE-SCSO) algorithm [94]. The SCSO variant with the best performance was chosen instead of the standard SCSO. In addition, three algorithms (Chaotic Grey Wolf Optimizer (CGWO) [36], Chaotic Marine Predators Algorithm (CMPA) [5], and Chaotic Whale Optimization Algorithm (CWOA) [65]) that are well-known in the literature and recently introduced hybrid chaotic models were compared with the obtained results. Although the major focus of the study is constrained and real-world complex problems, 24 more test functions were used, as mentioned above. Eight of them are unimodal, eight are multimodal, and the other eight are competitive testing functions. The reason for considering benchmark functions is to examine the effect of chaos in the newly proposed algorithm [26,63]. Detailed information on all functions is represented in the Appendix A. Chaos maps can be very helpful in finding the local and global optimum. These functions serve the designated purpose due to their nature. In addition, the behavior of algorithms in phases can be monitored in more detail on different problem types. In this study, we evaluated our proposed algorithm on 12 different maps (Table 2), and in tthis paper, we present the results separately in the relevant sections. The three chaos-based algorithms used (CGWO, CWOA, and CMPA) for comparison are based on the results obtained with its best map. All algorithms in this study were simulated using MATLAB, on the same PC, with a Core i7-11800 U 2.3 processor and 8GB of RAM. Specific parameters and their values are represented in Table 3, according to the working mechanism of the algorithms.

Table 3. Simulation of parameters for each optimization algorithm.

Algorithm	Parameter	Value
SE-SCSO	c	[2, 0]
	R	[−2*c*, 2*c*]
	θ	[−1, 1]
CSCSO	c	[2, 0]
	R	[−2*c*, 2*c*]
	θ	[−1, 1]
	k	1
CGWO	a	[0, 2π]
	A	[2, 0]
	C	[−2*c*, 2*c*]
CWOA	Angle (θ)	[−1, 1]
	α	5
	μ	0.5
CMPA	P	0.5
	FADs	0.2

5.2. Results and Discussion in Unimodal Test Functions

In this section, the proposed algorithm is analyzed independently on several famous unimodal functions with different chaotic maps, and the results are compared with those of other algorithms. The detailed information on these functions is represented in Table A1. Accordingly, in all algorithms, the number of populations is 30, the number of iterations is 500, and the number of independent runs is 10. The results are presented in Table 4. The analysis of the results was handled first from three different perspectives and then as a general analysis and discussion.

Table 4. The simulation results with the different algorithms on the unimodal benchmark functions (pop = 30, iter = 500, and run = 10).

Algorithm		F1	F2	F3	F4	F5	F6	F7	F8
Map1	**Best**	2.41E−125	3.94E−69	8.98E−108	2.43E−58	25.43E+00	1.01E+00	8.90E−06	4.66E−95
	Mean	1.59E−118	3.10E−62	1.84E−97	1.47E−48	27.17E+00	1.76E+00	4.67E−05	1.18E−90
	Std	5.00E−118	8.56E−62	5.20E−97	3.58E−48	1.13E+00	4.99E−01	2.75E−05	2.72E−90
Map2	**Best**	3.37E−131	2.74E−93	7.39E−107	1.56E−56	26.19E+00	1.45E+00	1.71E−05	2.27E−92
	Mean	1.75E−120	1.75E−84	3.78E−100	2.56E−49	27.72E+00	2.03E+00	1.68E−04	1.57E−90
	Std	4.48E−120	5.54E−84	9.96E−100	7.86E−49	9,88E−01	4,23E−01	2.14E−04	3.76E−90
Map3	**Best**	3.44E−129	**5.83E−96**	3.59E−114	2.06E−53	27.11E+00	1.02E+00	1.01E−05	8.24E−96
	Mean	2.48E−119	**7.82E−87**	1.11E−96	9.67E−51	27.95E+00	1.22E+00	1.43E−04	6.12E−91
	Std	6.05E−119	**2.36E−86**	3.47E−96	1.71E−51	6,93E−01	2,22E−01	2.60E−04	1.19E−90
Map4	**Best**	1.33E−128	3.35E−93	5.15E−113	2.06E−53	26.18E+00	1.24E+00	2.32E−06	3.15E−94
	Mean	4.70E−122	3.40E−86	7.34E−98	1.74E−50	27.64E+00	1.72E+00	7.36E−05	3.02E−90
	Std	1.33E−121	1.02E−85	2.27E−97	2.87E−50	1.00E+00	3,99E−01	1.01E−04	8.82E−90
Map5	**Best**	4.50E−128	4.08E−92	2.26E−98	2.10E−48	26.14E+00	5.71E−01	4.01E−06	3.67E−95
	Mean	7.52E−123	2.01E−86	4.72E−91	8.62E−46	27.59E+00	1.87E+00	9.16E−05	8.31E−88
	Std	1.28E−122	4.48E−86	1.07E−90	1.65E−45	9,88E−01	7,70E−01	1.11E−04	2.62E−87
Map6	**Best**	3.42E−142	3.70E−93	4.48E−110	1.82E−54	26.06E+00	1.25E+00	3.27E−05	1.72E−96
	Mean	2.63E−133	6.59E−85	1.60E−100	7.86E−50	27.56E+00	2.30E+00	1.94E−04	3.96E−93
	Std	7.36E−133	2.07E−84	3.56E−100	1.44E−49	9.84E−01	6.33E−01	2.20E−04	8.94E−93
Map7	**Best**	3.14E−134	2.90E−94	7.30E−110	2.28E−53	26.15E+00	1.27E+00	5.25E−06	6.54E−95
	Mean	4.24E−122	1.08E−86	8.87E−91	2.94E−49	27.35E+00	2.05E+00	1.49E−04	2.99E−88
	Std	1.34E−121	2.14E−86	2.79E−90	4.69E−49	6.68E−01	5.02E−01	2.37E−04	5.66E−88
Map8	**Best**	6.46E−158	4.90E−89	3.86E−123	**3.01E−61**	28.05E+00	1.74E+00	2.06E−06	4.89E−105
	Mean	1.26E−148	5.61E−84	1.13E−106	**1.45E−56**	28.15E+00	2.56E+00	6.35E−05	8.59E−98
	Std	3.91E−148	1.15E−83	3.39E−106	**2.38E−56**	6.48E−02	8.92E−01	5.90E−05	2.59E−97
Map9	**Best**	6.39E−137	4.44E−93	7.38E−111	1.93E−52	26.22E+00	5.91E−01	**9.09E−07**	6.15E−98
	Mean	1.06E−125	4.50E−86	2.66E−99	1.25E−47	27.89E+00	1.95E+00	**1.09E−05**	9.00E−91

Table 4. *Cont.*

Algorithm		F1	F2	F3	F4	F5	F6	F7	F8
	Std	3.31E−125	1.16E−85	7.74E−99	3.52E−47	1.02E+00	7.92E−01	**1.59E−05**	1.63E−90
	Best	**1.54E−170**	1.09E−92	**1.61E−141**	2.41E−53	**26.02E+00**	**4.57E−01**	1.74E−06	**9.98E−184**
Map10	**Mean**	**1.18E−158**	5.15E−84	**9.76E−118**	2.53E−47	**26.56E+00**	**1.01E+00**	1.48E−04	**3.84E−169**
	Std	**3.69E−158**	1.61E−83	**3.08E−117**	5.33E−47	**4.86E−01**	**2.39E−01**	2.70E−04	**0.00E+00**
	Best	9.73E−137	2.28E−92	7.99E−109	6.20E−56	26.16E+00	1.16E+00	7.90E−06	1.94E−95
Map11	**Mean**	3.64E−129	5.51E−79	2.26E−91	8.86E−51	27.99E+00	2.03E+00	3.29E−04	1.59E−89
	Std	1.09E−128	1.74E−78	7.13E−91	2.20E−50	1.09E+00	6.52E−01	3.25E−04	3.48E−88
	Best	1.15E−157	8.65E−84	2.90E−109	1.53E−55	26.17E+00	2.03E+00	2.69E−05	7.11E−113
Map12	**Mean**	1.21E−145	8.49E−79	3.41E−94	6.20E−47	28.26E+00	2.56E+00	2.70E−04	3.75E−105
	Std	3.79E−145	2.52E−78	9.04E−94	1.96E−46	1.08E+00	3.22E−01	4.00E−04	1.69E−104
	Best	4.83E−133	6.99E−74	2.44E−100	8.56E−52	27.08E+00	1.33E+00	1.99E−06	5.94E−99
SE-SCSO	**Mean**	2.02E−120	2.23E−64	3.85E−87	2.24E−42	28.07E+00	2.31E+00	1.31E−04	2.27E−95
	Std	3.48E−120	7.05E−64	1.19E−86	6.86E−42	7.15E−01	3.36E−01	1.43E−04	2.76E−95
	Best	5.98E−121	1.80E−65	3.31E−87	4.44E−45	28.71E+00	5.29E+00	7.46E−05	4.45E−92
CGWO	**Mean**	4.57E−118	1.50E−63	7.27E−80	1.70E−42	28.91E+00	5.56E+00	2.03E−04	4.73E−90
	Std	7.04E−118	1.97E−63	1.59E−79	2.25E−42	7.26E−02	1.47E−01	1.44E−04	7.36E−90
	Best	1.36E−12	2.44E−08	1.38E−06	1.90E−09	26.01E+00	**1.65E−04**	4.96E−04	7.99E−18
CMPA	**Mean**	3.38E−12	4.51E−07	8.88E−04	3.51E−09	27.51E+00	**5.64E−02**	1.13E−03	6.58E−16
	Std	2.07E−12	4.20E−07	1.88E−03	1.36E−09	8.76E−01	**7.41E−02**	5.19E−04	4.90E−16
	Best	1.21E−80	3.06E−48	4.24E+04	1.38E+00	28.24E+00	1.75E+00	3.46E−05	3.85E−53
CWOA	**Mean**	1.44E−74	8.48E−45	7.13E+04	3.91E+01	28.76E+00	2.15E+00	1.74E−03	4.29E−48
	Std	4.16E−74	2.41E−44	2.00E+04	2.22E+01	1.87E−01	3.41E−01	1.86E−03	1.33E−47

The optimum results obtained from the algorithms are bold and highlighted.

1-Analysis of the proposed algorithm within itself and between the maps used

In order to discuss the results in more detail, each function is evaluated separately. The optimum results obtained from the algorithms are highlighted, as shown in Table 4. In the F_1 evaluation, the CSCSO algorithm based on Map10 is in the first rank in best, mean, and std parameters. In this study, mean and std values were taken into account in the rankings. The second place is the scenario where Map8 is applied. According to the results, the last ranked algorithm (Rank 12) is CSCSO-Map1. In the F_2 evaluation, the CSCSO-Map3 achieved the best performance on both the best and mean parameters. The deviation value is also better than the results from the other 11 maps. In this function, our Map10-based algorithm took the seventh place, and, still, Map1 took the last place. In the F_3 evaluation, Map10 performed best in all parameters. Among the 12 maps, the worst result belongs to Map7. When other functions are evaluated similarly, the best result in F_4 is from Map8, and the weakest result is from Map1. In F_5, the best result is from Map8, and the worst result is from Map5. In function F_6, the best result is from Map10, and the poor outcome belongs to Map12. In F_6, the best result was nevertheless obtained from Map10, and the weakest result from Map12. Although the results are very close to each other in the F_7 evaluation, it is understood that the best result belongs to Map9, and the weakest result belongs to Map11, by a small margin, as a result of running the methods in many independent runs and considering mean and std values. In the final evaluation function (F_8), it is understood that, among the maps, Map10 and Map5 have the best and worst performances, respectively. Consequently, the proposed algorithm was analyzed within itself and among the maps used, and it is understood that the best results are obtained with the Tent Map (Map10). This map-based algorithm ranked first in five functions out of eight functions.

2-Comparison and analysis of the proposed algorithm with the SE-SCSO

A general pairwise comparison was made to evaluate the advantages and contributions of the Chaotic SCSO algorithm to the best SCSO variant. In the F_1 function, even the worst performing CSCSO-Map1 and Map3 appear to be better than SE-SCSO. The F_2 analysis shows that the SE-SCSO is better than only CSCSO-Map1 and lags behind 11 other maps. In F_3, F_4, F_5, and F_8, like in F_1, the CSCSO appears to be better than the SE-SCSO in all scenarios. In F_6, the SCSO seems to work better than Map12 and Map8, but not as well in other situations. In F_7 and F_8, the SE-SCSO found better results than four maps and worse than eight maps. It is understood that the proposed algorithm is generally better than the SE-SCSO algorithm in all maps.

3-Comparison and analysis of CSCSO algorithm with other three chaotic-based algorithms

When the results are examined in general, it is understood that the proposed algorithm outperforms other algorithms. The recently published CMPA algorithm found the best results in F_6. According to the results, in the functions F_1, F_2, F_3, F_4, F_7, and F_8, the CGWO algorithm performed better than the other three algorithms but weaker than the CSCSO. In F_5, it is concluded that although the CMPA algorithm is better than other algorithms, it is not as good as the CSCSO.

In general, the CSCSO, which could not find the good answer in only one function, found the best result in four functions when using Map10 and 3 functions based on Map3, Map8, and Map9. Therefore, it is understood that the proposed algorithm is superior to both the SE-SCSO and the other three chaos-based algorithms with different maps. This indicates that the CSCSO is a useful method for problems such as unimodal functions with one global optimum and no local optima.

In addition to the analysis made, Table 5 presents the statistical differences between CSCSO-Map10 (best-performing method) and the other maps and algorithms. For this, the popular Wilcoxon rank-sum and Friedman tests were used, inspired by the study in [5]. The *p*-values were generated by the Wilcoxon test, with a 0.05 significance level and over 10 independent runs. In this table, plus notation (+) demonstrates the superiority of the Tent-based proposed algorithm, minus notation (–) indicates that the obtained solution of the Tent-based proposed algorithm is worse than compared algorithms, and (~) notation indicates that a pair-wise comparison over both algorithms obtains equal values. The rank of each algorithm in each function and the average and overall ranking of each algorithm in the total of eight unimodal functions are presented in the *Avg_Rank* and *Overall_Rank* columns.

Table 5. Wilcoxon rank-sum test for the considered unimodal functions and the ranking of each algorithm.

Algorithms	F1	F2	F3	F4	F5	F6	F7	F8	Avg_Rank	Overall_Rank
Map1	13 (+)	14 (+)	7 (+)	8 (−)	2 (+)	5 (+)	2 (−)	7 (+)	7.250	7
Map2	10 (+)	7 (−)	4 (+)	6 (−)	8 (+)	8 (+)	10 (+)	8 (+)	7.625	8
Map3	12 (+)	1 (−)	8 (+)	3 (−)	10 (+)	3 (+)	6 (−)	5 (+)	6.000	2
Map4	9 (+)	4 (−)	6 (+)	4 (−)	7 (+)	4 (+)	4 (−)	10 (+)	6.000	2
Map5	7 (+)	3 (−)	11 (+)	12 (−)	6 (+)	6 (+)	5 (−)	14 (+)	8.000	10
Map6	4 (+)	6 (−)	3 (+)	5 (−)	5 (+)	12 (+)	11 (+)	4 (+)	6.250	6
Map7	8 (+)	2 (−)	12 (+)	7 (−)	3 (+)	10 (+)	8 (+)	13 (+)	7.875	9
Map8	2 (+)	9 (+)	2 (+)	1 (−)	13 (+)	15 (+)	3 (−)	3 (+)	6.000	2
Map9	6 (+)	5 (−)	5 (+)	9 (−)	9 (+)	7 (+)	1 (−)	6 (+)	6.000	2
Map10	1	8	1	10	1	2	7	1	3.875	1
Map11	5 (+)	10 (+)	10 (+)	2 (−)	11 (+)	9 (+)	14 (+)	12 (+)	9.125	11
Map12	3 (+)	11 (+)	9 (+)	11 (+)	14 (+)	14 (+)	13 (+)	2 (+)	9.625	12
SE-SCSO	11 (+)	12 (+)	13 (+)	14 (+)	12 (+)	13 (+)	9 (+)	9 (+)	11.625	13
CGWO	14 (+)	13 (+)	14 (+)	13 (+)	16 (+)	16 (+)	12 (+)	11 (+)	13.375	15
CMPA	16 (+)	16 (+)	15 (+)	15 (+)	4 (+)	1 (−)	15 (+)	16 (+)	12.250	14
CWOA	15 (+)	15 (+)	16 (+)	16 (+)	15 (+)	11 (+)	16 (+)	15 (+)	14.875	16

Our final analysis of unimodal functions is the analysis of the data distribution in each function of each algorithm, using boxplots. In these plots, the performance of each algorithm is presented separately for each function (Figure 4). Based on this standardized approach, the distribution, locality, and skewness groups of numerical data are graphically represented by their quartiles. The lowest and highest data points of the algorithm, which are whisker edges, are the minimum and maximum. In general, a compact boxplot shows strong data agreement. According to the analysis of these results, the proposed algorithm also performs well according to this perspective.

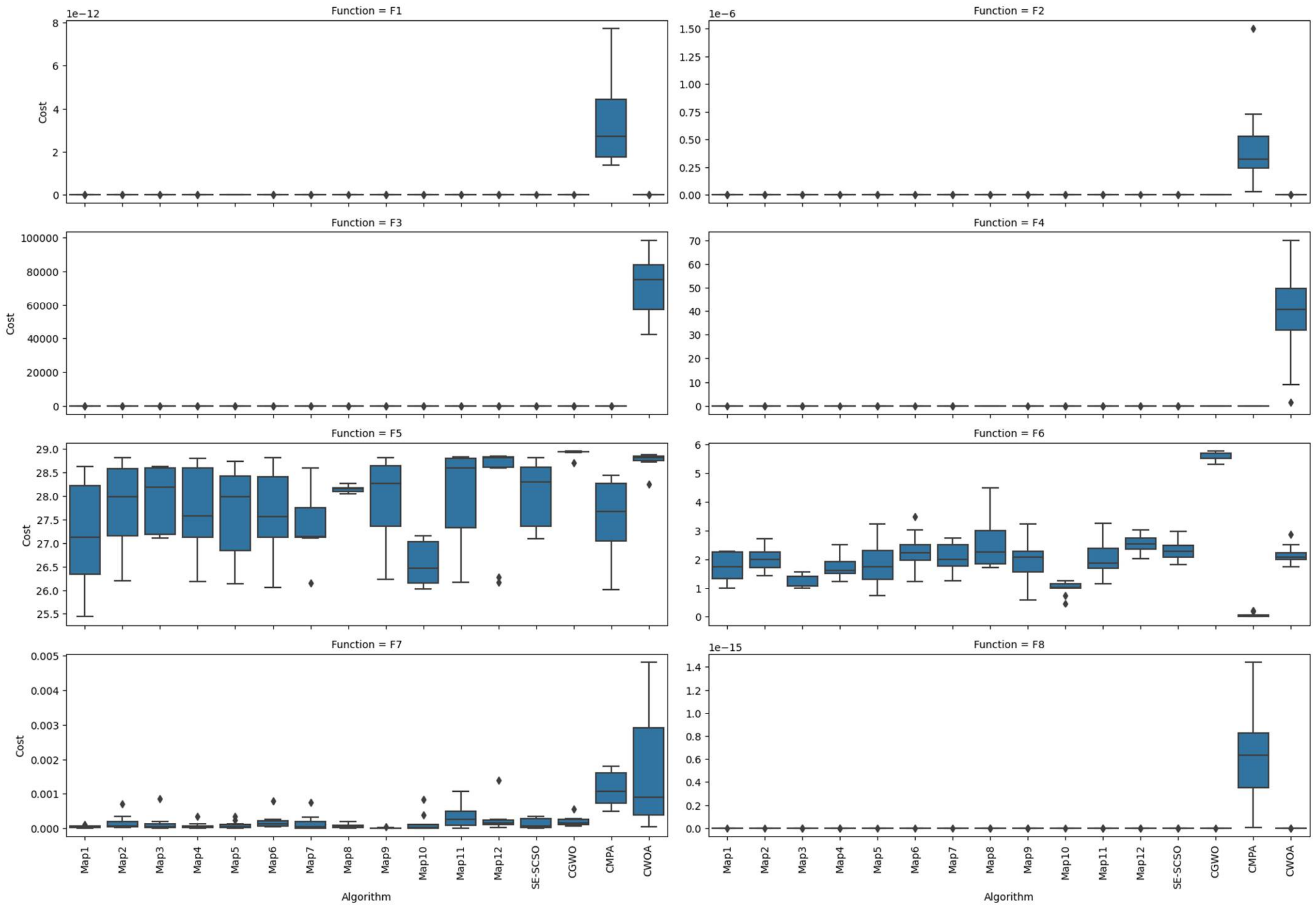


Figure 4. Boxplots of the results obtained by all algorithms on unimodal benchmark functions.

5.3. *Results and Discussion in Multimodal Test Functions*

In this section, the performance of the CSCSO algorithm is evaluated on some famous multimodal test functions, using different chaotic maps, and the obtained results are compared with other algorithms. Details of these functions are presented in Table A2. The results performed on these benchmark functions are presented in Table 6. The number of populations, the number of iterations, and the number of independent runs are assumed to be 30, 500, and 10, respectively. As in the previous section, analyses of the results are evaluated from different perspectives.

Table 6. The simulation results with the different algorithms on the multimodal benchmark functions (pop = 30, iter = 500, and run = 10).

Algorithm		MF1	MF2	MF3	MF4	MF5	MF6	MF7	MF8
	Best	−8.73E+03	5.69E−04	9.23E−02	1.98E+00	1.71E−14	8.11E−305	8.72E−49	−1.00E+00
Map1	**Mean**	−7.11E+03	1.13E−03	1.82E−01	2.12E+00	3.93E−12	1.14E−298	4.82E−13	−8.99E−01
	Std	6.55E+02	8.65E−04	7.00E−02	8.99E−02	7.34E−12	0.00E+00	1.48E−12	5.16E−01
	Best	−8.13E+03	5.68E−04	4.42E−02	1.84E+00	**9.15E−17**	0.00E+00	1.54E−45	−1.00E+00
Map2	**Mean**	−7.13E+03	1.50E−03	1.97E−01	2.28E+00	**2.39E−12**	1.02E−301	3.74E−09	−5.00E−01
	Std	6.36E+02	1.12E−03	1.07E−01	2.47E−01	**4.89E−12**	0.00E+00	1.18E−08	5.27E−01
	Best	−7.97E+03	7.17E−04	6.04E−02	1.82E+00	8.55E−15	1.08E−312	8.06E−41	−1.00E+00
Map3	**Mean**	−7.01E+03	1.44E−03	2.51E−01	2.18E+00	9.10E−12	9.53E−300	4.87E−11	−8.00E−01

Table 6. *Cont.*

Algorithm		MF1	MF2	MF3	MF4	MF5	MF6	MF7	MF8
	Std	8.64E+02	5.62E−04	8.99E−02	2.18E−01	1.35E−11	0.00E+00	1.54E−10	4.22E−01
	Best	−7.71E+03	5.69E−04	1.42E−01	1.90E+00	8.53E−14	0.00E+00	3.77E−47	−1.00E+00
Map4	**Mean**	−7.15E+03	1.31E−03	2.25E−01	2.19E+00	6.92E−12	4.24E−299	4.63E−07	−7.00E−01
	Std	5.06E+02	8.80E−04	5.69E−02	1.76E−01	1.01E−11	0.00E+00	1.46E−06	4.83E−01
	Best	−7.57E+03	7.13E−04	8.39E−02	1.48E+00	4.50E−14	**0.00E+00**	1.46E−46	−1.00E+00
Map5	**Mean**	−6.99E+03	1.50E−03	1.53E−01	2.07E+00	4.38E−12	**0.00E+00**	7.64E−13	−8.00E−01
	Std	3.99E+02	1.31E−03	6.69E−02	2.73E−01	6.66E−12	**0.00E+00**	2.05E−12	4.22E−01
	Best	−8.16E+03	5.68E−04	1.27E−01	1.92E+00	2.54E−13	0.00E+00	2.51E−49	−1.00E+00
Map6	**Mean**	−6.87E+03	1.24E−03	2.46E−01	2.22E+00	8.05E−12	1.72E−289	7.83E−23	−8.00E−01
	Std	1.08E+03	7.03E−04	6.83E−02	2.05E−01	1.01E−11	0.00E+00	2.48E−22	4.22E−01
	Best	−7.71E+03	6.52E−04	1.08E−01	1.85E+00	4.01E−14	0.00E+00	4.59E−53	−1.00E+00
Map7	**Mean**	−6.93E+03	1.22E−03	1.95E−01	2.11E+00	5.19E−12	5.16E−280	7.78E−19	−2.00E−01
	Std	9.19E+02	3.89E−04	5.21E−02	1.66E−01	8.29E−12	0.00E+00	2.46E−18	4.22E−01
	Best	−7.67E+03	1.07E−03	1.79E−01	2.31E+00	4.30E−14	0.00E+00	2.53E−48	**−1.00E+00**
Map8	**Mean**	−7.15E+03	1.88E−03	3.74E−01	2.57E+00	1.96E−11	4.59E−289	1.24E−15	**−1.00E+00**
	Std	3.64E+02	7.04E−04	1.34E−02	1.32E−01	4.65E−11	0.00E+00	3.90E−15	**0.00E+00**
	Best	−7.70E+03	6.46E−04	1.14E−01	1.63E+00	5.19E−14	0.00E+00	1.21E−45	−1.00E+00
Map9	**Mean**	−6.91E+03	1.20E−03	2.24E−01	2.10E+00	2.32E−11	4.16E−281	4.69E−13	−7.00E−01
	Std	7.25E+02	4.89E−04	7.06E−02	1.32E−01	5.30E−11	0.00E+00	1.48E−12	4.83E−01
	Best	**−8.51E+03**	**5.69E−04**	**3.45E−02**	**1.43E+00**	1.52E−14	**0.00E+00**	2.49E−41	−1.00E+00
Map10	**Mean**	**−7.35E+03**	**1.04E−03**	**1.04E−01**	**1.84E+00**	7.82E−12	**0.00E+00**	5.97E−19	−2.00E−01
	Std	**5.94E+02**	**4.27E−04**	**3.70E−02**	**2.60E−01**	9.65E−12	**0.00E+00**	1.86E−18	4.22E−01
	Best	−8.01E+03	7.19E−04	5.53E−02	1.65E+00	2.33E−14	0.00E+00	9.91E−43	−1.00E+00
Map11	**Mean**	−7.02E+03	1.26E−03	1.53E−01	2.16E+00	2.12E−11	8.32E−280	8.45E−07	−1.00E−01
	Std	6.60E+02	4.76E−04	7.64E−02	3.39E−01	4.68E−11	0.00E+00	2.66E−06	3.16E−01
	Best	−8.02E+03	1.48E−03	1.31E−01	2.10E+00	2.57E−13	5.85E−299	**5.65E−165**	−1.00E+00
Map12	**Mean**	−6.85E+03	3.16E−03	3.11E−01	2.21E+00	1.64E−11	1.52E−269	**6.62E−63**	−9.00E−01
	Std	6.40E+02	1.92E−03	6.77E−02	7.76E−02	2.51E−11	0.00E+00	**2.09E−62**	3.16E−01
	Best	−7.78E+03	4.07E−04	4.60E−02	2.02E+00	3.28E−16	5.29E−241	3.53E−49	**−1.00E+00**
SE-SCSO	**Mean**	−7.01E+03	2.19E−02	1.97E−01	2.29E+00	7.92E−12	3.97E−208	1.43E−16	**−1.00E+00**
	Std	5.46E+02	1.35E−02	1.20E−01	2.66E−01	2.18E−11	0.00E+00	4.37E−16	**0.00E+00**
	Best	−3.69E+03	2.52E−02	1.20E+00	2.90E+00	7.42E−10	2.15E−292	6.21E−133	−1.00E+00
CGWO	**Mean**	−2.53E+03	4.53E−02	1.36E+00	3.05E+00	7.15E−08	4.10E−246	1.89E−60	−9.77E−02
	Std	5.26E+02	1.41E−02	6.98E−02	8.14E−02	6.77E−08	0.00E+00	5.98E−60	3.17E−01
	Best	−1.02E+04	5.66E−04	1.00E−02	1.02E+00	**1.97E−31**	9.12E−149	1.05E−48	−9.52E−01
CMPA	**Mean**	−9.50E+03	1.50E−03	2.78E−01	2.07E+00	**1.52E−25**	2.70E−64	2.64E−32	−9.52E−02
	Std	4.15E+02	2.19E−03	2.20E−01	5.94E−01	**4.57E−25**	8.54E−64	8.33E−32	3.01E−01
	Best	**−1.15E+04**	5.66E−04	7.53E−02	**1.29E+00**	1.66E−12	9.01E−288	5.74E−71	−1.00E+00
CWOA	**Mean**	**−9.71E+03**	1.10E−03	1.29E−01	**1.65E+00**	1.61E−10	9.35E−09	9.66E−09	−9.98E−02
	Std	**1.17E+03**	5.34E−04	4.47E−02	**3.96E−01**	1.95E−10	1.35E−08	3.00E−08	3.16E−01

The optimum results obtained from the algorithms are bold and highlighted.

1-Analysis of the proposed algorithm within itself and between the maps used.

The results obtained by applying maps to the proposed algorithm were analyzed. Multimodal functions have more than one local optima, but they also have a single global optimum. This type of benchmarking function is useful for the performance evaluation of optimization algorithms in the exploration and exploitation phases. In the evaluation of eight multimodal functions, CSCSO-Map10 found the best results in five functions, while the other three functions, Map2, Map5, and Map8, found the best results. In the assessment of MF_1, Map10-based CSCSO is in first place, and the second place is Map2. In this function, the most inefficient map is the number 12. It should be emphasized that, in this function, Map1 is better than other maps in its best parameter. However, as stated before, the evaluations in this study are based on mean and std values. In MF_2 evaluation, Map10 is better than other maps. The remaining maps ranks from best to worst as follows: Map1, Map9, Map7, Map6, Map11, Map4, Map2, Map3, Map5, Map8, Map8, and Map12 based CSCSO. In the MF_3 analysis, Map10 performs best in all parameters. Among the

12 maps, the worst result belongs to Map3. Map10 received the best results in MF$_4$, and the worst was Map8. In other functions, in turn, in MF5, the best map was Map2, and the worst was Map11. The best Map in MF$_6$ was Map5, and the worst was Map2. The best map in MF$_7$ was Map12, and the worst was Map11. In MF$_8$, the best result was Map8, while Map11 found results as the worst map. As a result, the CSCSO was analyzed within itself and among the maps used, and it is understood that the best results are obtained with the Tent Map (Map10) that ranked first in five functions out of eight multimodal test functions.

2-Comparison and analysis of the proposed algorithm with the SE-SCSO

As emphasized in the previous section, the best approach for measuring the performance of the proposed algorithm is to compare it with the SE-SCSO algorithm. The CSCSO algorithm based on MF$_1$, MF$_2$, MF$_3$, MF$_4$, MF$_5$, MF$_6$, and MF$_7$ is better than SE-SCSO. However, it is the best in MF$_8$, together with Map10, so it finds better results than the other 11 maps in MF$_8$. Briefly, it is understood that the proposed algorithm is better than the SE-SCSO algorithm in seven multimodal functions.

3-Comparison and analysis of CSCSO algorithm with other three chaos-based algorithms

The CSCSO algorithm is compared with other chaotic-based algorithms based on the maps that give the best results in each function. Accordingly, it is understood that the proposed algorithm is better than the other three algorithms in MF$_2$, MF$_3$, MF$_4$, MF$_6$, and MF$_7$. This means 62.5% success. While the CWOA algorithm in MF$_1$ and the CMPA algorithm in MF$_5$ found the best results, the SE-SCSO and CSCSO-Map10 jointly took the first place in MF$_8$. In general, the CSCSO, which does not find good answers only in two functions, has a 75% success rate in multimodal functions, with multiple local optimums and one global optima.

Here, the pairwise comparison test is performed for multimodal functions in line with similar definitions and expressions as it is performed for unimodal functions. According to the analysis of the results, as presented in Table 7, the proposed algorithm based on the Tent Map ranked first. In addition, the boxplots of the results by all algorithms on multimodal benchmark functions are presented in Figure 5.

Table 7. Wilcoxon rank-sum test for the considered multimodal functions and the ranking of each algorithm.

Algorithms	MF1	MF2	MF3	MF4	MF5	MF6	MF7	MF8	Avg_Rank	Overall_Rank
Map1	7 (+)	3 (+)	5 (+)	7 (+)	3 (−)	6 (+)	10 (+)	4 (−)	5.625	2
Map2	6 (+)	11 (+)	7 (+)	13 (+)	2 (−)	3 (+)	13 (+)	10 (−)	8.125	6
Map3	9 (+)	9 (+)	12 (+)	9 (+)	10 (+)	4 (+)	12 (+)	5 (−)	8.750	10
Map4	5 (+)	8 (+)	10 (+)	10 (+)	6 (−)	5 (+)	15 (+)	8 (−)	8.375	8
Map5	11 (+)	10 (+)	3 (+)	3 (+)	4 (−)	1 (~)	11 (+)	5 (−)	6.000	3
Map6	14 (+)	6 (+)	11 (+)	12 (+)	9 (+)	7 (+)	4 (−)	5 (−)	8.500	9
Map7	12 (+)	5 (+)	6 (+)	6 (+)	5 (−)	10 (+)	6 (+)	11 (~)	7.625	4
Map8	4 (+)	13 (+)	15 (+)	15 (+)	12 (+)	8 (+)	8 (+)	1 (−)	9.500	12
Map9	13 (+)	4 (+)	9 (+)	5 (+)	14 (+)	9 (+)	9 (+)	8 (−)	8.875	11
Map10	3	1	1	2	7	1	5	11	3.875	1
Map11	8 (+)	7 (+)	4 (+)	8 (+)	13 (+)	11 (+)	16 (+)	13 (+)	10.000	14
Map12	15 (+)	14 (+)	14 (+)	11 (+)	11 (+)	12 (+)	1 (−)	3 (−)	10.125	15
SE-SCSO	10 (+)	15 (+)	8 (+)	14 (+)	8 (+)	14 (+)	7 (+)	2 (−)	9.750	13
CGWO	16 (+)	16 (+)	16 (+)	16 (+)	16 (+)	13 (+)	2 (−)	15 (+)	13.750	16
CMPA	2 (−)	12 (+)	13 (+)	4 (+)	1 (−)	15 (+)	3 (−)	16 (+)	8.250	7
CWOA	1 (−)	2 (+)	2 (+)	1 (−)	15 (+)	16 (+)	14 (+)	14 (+)	8.125	5

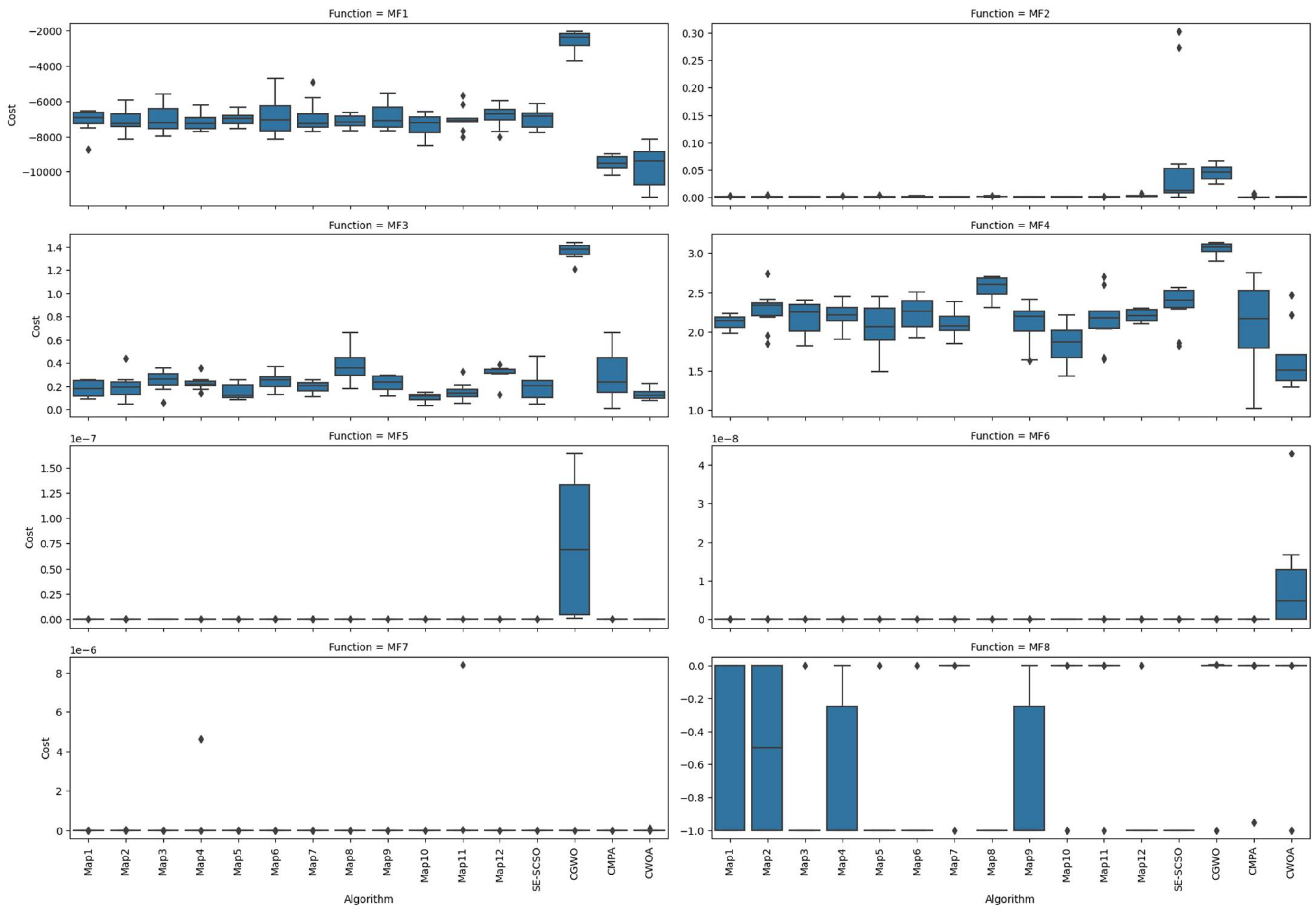


Figure 5. Boxplots of the results obtained by all algorithms on multimodal benchmark functions.

5.4. Results and Discussion in Competitive Test Functions

The latest assessment and analysis discussed in this section are made on the competitive test functions of CEC2019. They are the 100−digit challenge also known as "CEC−C06 benchmark test functions" [89]. They calculate the function values of the "*horizontal*" slices of the convergence graph. These test functions are used for the evaluation of optimization algorithms in large-scale problems [97]. All functions are scalable, and their global optimums are one. The definition and details of the functions are presented in Table A3. The results of these functions were applied to twelve maps in the proposed algorithm and compared to four algorithms. According to the results presented in Table 8, the suggested algorithm found the best results in the other five functions, except the CEC03 and CEC05. The Map10-based CSCSO algorithm performs best in five functions, and Map7 performs best in CEC07, while the CMPA algorithm gives the best results in CEC03 and CEC05.

Table 8. The simulation results with the different algorithms on the competitive benchmark functions (pop = 30, iter = 500, and run = 10).

Algorithm		CEC01	CEC02	CEC03	CEC04	CEC05	CEC06	CEC07	CEC08
	Best	3.92E+04	1.73E+01	5.39E+01	1.18E+00	5.75E+00	1.67E+02	3.92E+00	3.47E+00
Map1	**Mean**	4.48E+04	1.74E+01	4.50E+02	1.46E+00	8.79E+00	3.13E+02	5.09E+00	4.43E+01
	Std	3.60E+03	1.06E−01	1.18E+03	3.50E−01	1.66E+00	1.51E+02	5.59E−01	7.71E−01
	Best	3.92E+04	1.73E+01	6.00E+01	1.31E+00	7.86E+00	1.78E+01	4.00E+00	2.71E+00
Map2	**Mean**	4.38E+04	1.73E+01	4.33E+02	1.67E+00	9.61E+00	2.66E+02	5.29E+00	4.46E+01
	Std	5.18E+03	6.30E−04	7.73E+02	4.16E−01	1.12E+00	1.94E+02	7.67E−01	9.30E−01

Table 8. *Cont.*

Algorithm		CEC01	CEC02	CEC03	CEC04	CEC05	CEC06	CEC07	CEC08
	Best	3.92E+04	1.73E+01	4.49E+01	1.20E+00	6.20E+00	3.28E+01	4.56E+00	3.25E+00
Map3	**Mean**	4.42E+04	1.73E+01	6.11E+02	1.77E+00	8.10E+00	3.74E+02	5.31E+00	4.64E+00
	Std	3.10E+03	2.43E−04	4.49E+02	5.06E−01	1.29E+00	2.36E+02	5.41E−01	1.26E+00
	Best	3.80E+04	1.73E+01	6.76E+01	1.24E+00	4.82E+00	1.29E+02	4.16E+00	3.39E+00
Map4	**Mean**	4.21E+04	1.74E+01	1.67E+03	1.68E+00	6.28E+00	3.23E+02	5.05E+00	4.43E+00
	Std	2.27E+03	1.05E−03	1.69E+03	4.06E−01	8.95E−01	1.88E+02	5.56E−01	8.46E−01
	Best	3.87E+04	1.73E+01	7.46E+01	1.18E+00	8.52E+00	1.04E+02	4.41E+00	3.38E+00
Map5	**Mean**	4.40E+04	1.73E+01	5.55E+02	1.50E+00	1.02E+01	4.14E+02	5.29E+00	4.64E+00
	Std	5.62E+03	1.05E−03	8.32E+02	2.09E−01	9.64E−01	2.52E+02	6.37E−01	8.05E−01
	Best	4.11E+04	1.73E+01	3.00E+01	1.28E+00	6.03E+00	1.39E+02	4.37E+00	3.99E+00
Map6	**Mean**	4.38E+04	1.74E+01	4.15E+02	1.71E+00	7.87E+00	4.11E+02	5.32E+00	5.67E+00
	Std	2.33E+03	1.40E−01	5.93E+02	3.27E−01	1.12E+00	2.03E+02	6.19E−01	1.04E+00
	Best	3.90E+04	1.73E+01	7.06E+01	1.24E+00	7.04E+00	1.64E+01	5.09E+00	3.67E+00
Map7	**Mean**	4.31E+04	1.73E+01	7.47E+02	1.49E+00	9.63E+00	3.23E+02	5.58E+00	5.09E+00
	Std	2.61E+03	6.01E−04	1.12E+03	1.35E−01	1.54E+00	1.80E+02	3.61E−01	7.57E−01
	Best	4.07E+04	1.73E+01	4.68E+01	1.36E+00	4.20E+00	1.27E+01	**2.95E+00**	3.97E+00
Map8	**Mean**	5.00E+04	1.75E+01	1.67E+03	1.82E+00	6.86E+00	2.99E+02	**4.06E+00**	5.62E+00
	Std	6.35E+03	3.35E−01	1.53E+03	4.83E−01	1.43E+00	1.56E+02	**8.80E−01**	1.47E+00
	Best	3.77E+04	1.73E+01	4.37E+01	1.27E+00	6.31E+00	3.38E+01	4.20E+00	2.78E+00
Map9	**Mean**	4.31E+04	1.74E+01	3.99E+02	1.78E+00	8.67E+00	4.03E+02	5.25E+00	3.82E+00
	Std	4.96E+03	1.05E−01	7.77E+02	5.35E−01	1.52E+00	2.21E+02	4.98E−01	1.13E+00
	Best	**3.94E+04**	**1.72E+01**	4.93E+01	**1.02E+00**	6.01E+00	**1.12E+02**	2.98E+00	**2.34E+00**
Map10	**Mean**	**4.19E+04**	**1.72E+01**	5.48E+02	**1.06E+00**	9.92E+00	**2.65E+02**	4.27E+00	**2.42E+00**
	Std	**1.69E+03**	**7.09E−03**	8.42E+02	**3.72E−02**	1.76E+00	**1.28E+02**	7.97E−01	**7.24E−02**
	Best	3.73E+04	1.73E+01	7.47E+00	1.22E+00	8.16E+00	2.16E+01	4.86E+00	3.11E+00
Map11	**Mean**	5.07E+04	1.74E+01	7.04E+02	1.48E+00	9.82E+00	3.27E+02	5.79E+00	4.63E+00
	Std	1.47E+04	8.86E−02	7.78E+02	2.68E−01	9.54E−01	1.43E+02	5.02E−01	1.17E+00
	Best	4.00E+04	1.73E+01	2.47E+00	1.58E+00	4.24E+00	2.72E+01	4.58E+00	4.85E+00
Map12	**Mean**	5.26E+04	1.74E+01	2.42E+02	2.02E+00	6.24E+00	2.83E+02	5.30E+00	4.63E+00
	Std	1.07E+04	1.41E−01	3.16E+02	5.10E−01	1.86E+00	1.76E+02	5.71E−01	1.17E+00
	Best	4.08E+04	1.73E+01	1.43E+01	1.17E+00	5.94E+00	1.43E+02	4.49E+00	4.49E+00
SE-SCSO	**Mean**	4.50E+04	1.73E+01	1.19E+02	1.57E+00	8.17E+00	3.95E+02	5.43E+00	5.60E+00
	Std	2.97E+03	1.64E−01	1.59E+03	3.51E−01	9.83E−01	1.74E+02	6.52E−01	5.58E−01
	Best	5.22E+04	1.82E+01	1.25E+04	4.42E+00	1.06E+01	1.11E+03	5.76E+00	6.45E+00
CGWO	**Mean**	6.07E+04	1.84E+01	2.73E+04	5.03E+00	1.16E+01	1.47E+03	6.33E+00	9.26E+00
	Std	4.65E+03	1.44E−01	6.67E+03	3.47E−01	4.87E−01	2.59E+02	4.16E−01	1.91E+00
	Best	4.77E+04	1.72E+01	**9.95E+00**	1.26E+00	**1.32E+00**	8.64E+02	4.05E+00	4.41E+00
CMPA	**Mean**	1.11E+05	1.73E+01	**1.62E+01**	1.54E+00	**2.36E+00**	1.30E+03	5.38E+00	5.39E+00
	Std	7.41E+04	8.96E−08	**4.35E+01**	2.59E−01	**5.76E−01**	2.39E+02	5.84E−01	8.52E−01
	Best	1.07E+06	1.73E+01	2.19E+03	2.29E+00	7.98E+00	3.72E+02	5.58E+00	6.87E+01
CWOA	**Mean**	2.32E+08	1.76E+01	5.67E+03	3.22E+00	1.02E+01	6.61E+02	6.21E+00	3.85E+02
	Std	6.01E+08	3.27E−01	2.48E+03	6.01E−01	1.00E+00	2.03E+02	3.53E−01	2.69E+00

The optimum results obtained from the algorithms are bold and highlighted.

When the results were discussed in terms of the overall evaluation, the best algorithm in CEC01 is based on Map10, while the worst-performing algorithm is CWOA. In this function, the proposed algorithm is better than the other three chaotic-based algorithms and also found promising results from the SE-SCSO algorithm, except for Map8, Map11, and Map12. While Map10 ranks first in CEC02 and CMPA, Map5, Map2, Map7, and Map3 are the top in SE-SCSO; and Map4, Map11, Map1, Map6, Map9, Map12, and Map8 rank as the top in the CWOA and CGWO algorithms. Although CMPA ranked first according to the "mean" parameter in CEC03, Map12 and Map11, respectively, found the best solutions in the "best" parameter. In CEC04, Map10 and Map1 are at the top, indicating that the proposed algorithm is more performant. The CMPA received the best solution in CEC05, and the algorithm using Map12 took second place. On the other hand, the CSCSO algorithm is better than the SE-SCSO algorithm, except for Map2, Map5, Map7, Map10, and Map11.

In the CEC06 evaluation, Map10 is in the first rank, and the last rank belongs to the CGWO algorithm. Based on the obtained results, the CSCSO algorithm is better than the SE-SCSO algorithm, excluding Map5, Map6, and Map9. In the CEC07 function, the best and worst performances belong to the CSCSO-Map8 and CGWO algorithms. The CSCSO, which found good results in general, is better on other Maps, except for Map7 and Map11, also compared to the SE-SCSO algorithm. In the last function (CEC08), CSCSO-Map10 found the best results, while CWOA performed the poorest. In general, the CSCSO, which does not find good answers only in two functions, has a 75% success rate in competitive test functions.

For our analyses of competitive functions, pairwise comparison results in the form of p-values are presented in Table 9, and boxplots of the obtained results are presented in Figure 6. As mentioned earlier, the boxplot analysis is a tool that shows the characteristics of the distributions of the data. Therefore, box plots are widely used in the literature to show quarter-based data distributions. Judging by the results, the CSCSO algorithm based on the Tent Map failed to rank well in CEC03 and CEC05. However, it has always been first in the rest of the other competitive functions. Piecewise− and Logistic-Maps-based proposed algorithms are in the second and third ranks.

Table 9. Wilcoxon rank-sum test for the considered competitive functions and the ranking of each algorithm.

Algorithms	CEC01	CEC02	CEC03	CEC04	CEC05	CEC06	CEC07	CEC08	Avg_Rank	Overall_Rank
Map1	9 (+)	8 (+)	6 (−)	2 (+)	8 (−)	5 (+)	4 (+)	14 (+)	7.000	6
Map2	6 (+)	2 (+)	5 (−)	8 (+)	10 (−)	2 (+)	6 (+)	15 (+)	6.750	4
Map3	8 (+)	2 (+)	9 (+)	11 (+)	6 (−)	9 (+)	9 (+)	6 (+)	7.500	7
Map4	2 (+)	8 (+)	14 (+)	9 (+)	3 (−)	6 (+)	3 (+)	3 (+)	6.000	2
Map5	7 (+)	2 (+)	8 (+)	5 (+)	14 (+)	13 (+)	7 (+)	7 (+)	7.875	10
Map6	5 (+)	8 (+)	4 (−)	10 (+)	5 (−)	12 (+)	10 (+)	12 (+)	8.125	11
Map7	3 (+)	2 (+)	11 (+)	4 (+)	11 (−)	7 (+)	13 (+)	9 (+)	7.500	7
Map8	11 (+)	14 (+)	13 (+)	13 (+)	4 (−)	4 (+)	1 (−)	11 (+)	8.875	13
Map9	4 (+)	8 (+)	3 (−)	12 (+)	7 (−)	11 (+)	5 (+)	2 (+)	6.500	3
Map10	1	1	7	1	13	1	2	1	3.375	1
Map11	12 (+)	8 (+)	10 (+)	3 (+)	12 (−)	8 (+)	14 (+)	4 (+)	8.875	13
Map12	13 (+)	8 (+)	2 (−)	14 (+)	2 (−)	3 (+)	8 (+)	5 (+)	6.875	5
SE-SCSO	10 (+)	2 (+)	12 (+)	7 (+)	9 (−)	10 (+)	12 (+)	8 (+)	8.750	12
CGWO	14 (+)	16 (+)	16 (+)	16 (+)	16 (+)	16 (+)	16 (+)	13 (+)	15.375	16
CMPA	15 (+)	2 (+)	1 (−)	6 (+)	1 (−)	15 (+)	11 (+)	10 (+)	7.625	9
CWOA	16 (+)	15 (+)	15 (+)	15 (+)	15 (+)	14 (+)	15 (+)	16 (+)	15.125	15

5.5. Results and Discussion in Numerical Real-World Optimization Problems

In this section, the performance of the CSCSO algorithm is studied on the CEC 2020 test suite consisting of eight numerically constrained functions, as detailed in Table A4. The benchmarks in this intriguing suite are selected from multimodal, new hybrid, and composite functions. For a fair comparison, the results of the state-of-the-art CGWO, CMPA, and CWOA algorithms are compared with those of the CSCSO, as in the other sections. Table 10 presents the statistical results of the CSCSO algorithm with different chaotic maps in eight CEC2020 numerical optimization functions versus other metaheuristics. The results indicate that the proposed algorithm based on the Tent Map (Map10) finds the best answers when compared to the other algorithms in six functions (75% success rate). According to these results, the best performance in R_5 and R_6 belongs to the CMPA algorithm. In these functions, Map5 took the second place in R_5, and Map10 took second place in R_6. According to the results, the CGWO, CWOA, and SE-SCSO algorithms failed to find the optimal solution for CEC 2020 numerical problems. As a result, our Tent-based algorithm found the best results for the problems covered in this category.

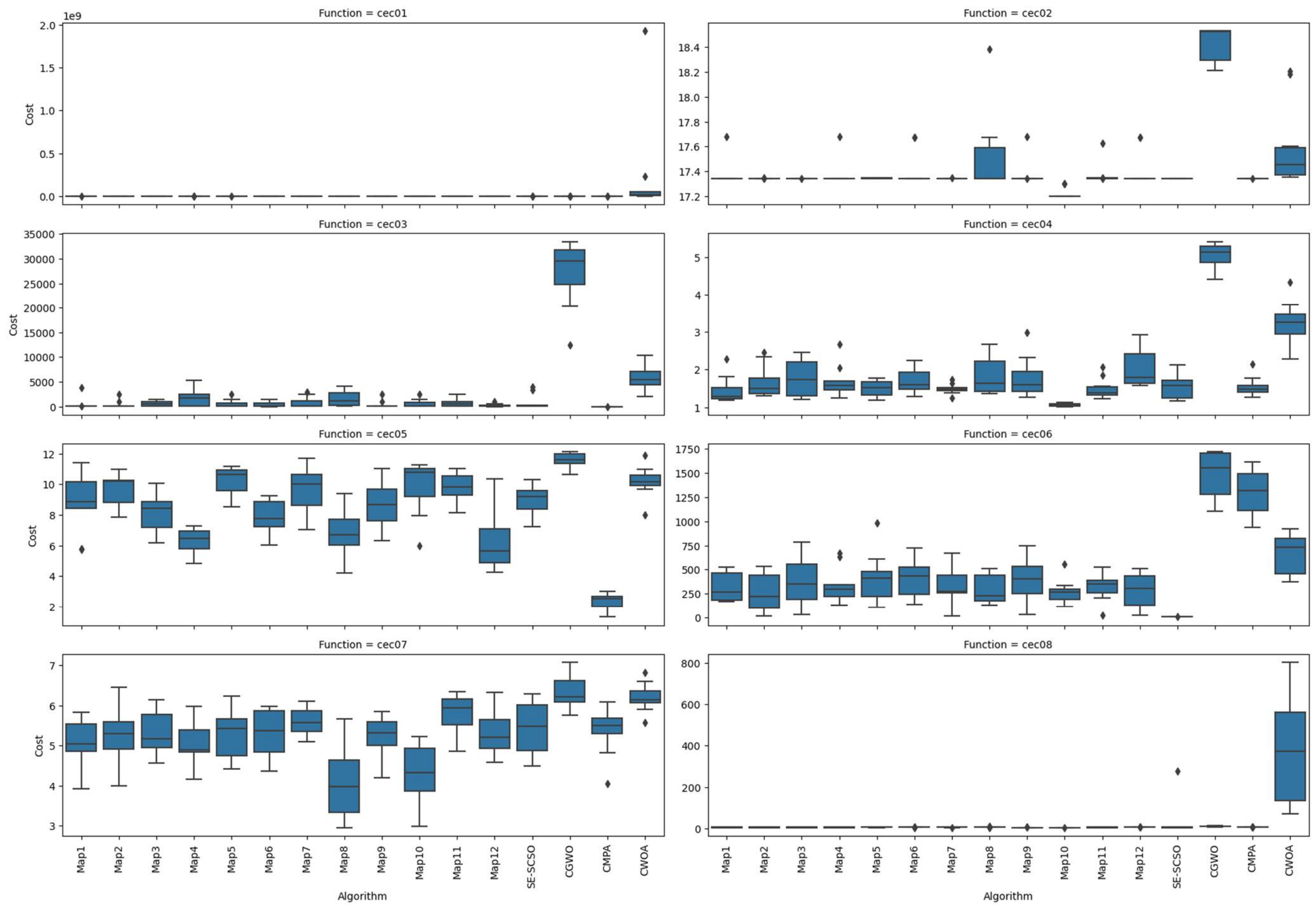


Figure 6. Boxplots of the results obtained by all algorithms on competitive functions.

Table 10. The simulation results with the different algorithms on the numerical real-world problems (pop = 40, iter = 1000, and run = 10).

Algorithm		R1	R2	R3	R4	R5	R6	R7	R8
	Best	8.18E+02	5.24E+02	3.10E+02	6.16E+02	5.09E+03	7.08E+02	2.94E+03	1.31E+03
Map1	**Mean**	9.77E+02	6.46E+02	3.51E+02	6.36E+02	6.58E+03	8.38E+02	3.84E+03	3.47E+03
	Std	1.85E+02	8.11E+01	3.17E+01	2.13E+01	9.73E+02	9.52E+01	7.34E+02	1.16E+03
	Best	7.47E+02	5.17E+02	3.45E+02	6.14E+02	4.59E+03	7.37E+02	2.93E+03	1.13E+03
Map2	**Mean**	9.49E+02	6.49E+02	3.71E+02	6.40E+02	7.14E+03	7.93E+02	4.67E+03	2.89E+03
	Std	2.17E+02	1.23E+02	1.91E+01	2.81E+01	1.60E+03	7.32E+01	9.96E+02	9.62E+02
	Best	7.09E+02	5.30E+02	3.08E+02	6.04E+02	5.33E+03	7.70E+02	2.96E+03	1.86E+03
Map3	**Mean**	8.61E+02	6.53E+02	3.36E+02	6.20E+02	6.60E+03	8.48E+02	4.20E+03	3.66E+03
	Std	1.75E+02	1.22E+02	2.57E+01	1.17E+01	1.15E+03	7.97E+01	6.96E+02	1.13E+03
	Best	7.22E+02	5.38E+02	3.24E+02	6.09E+02	5.02E+03	7.38E+02	1.81E+03	2.32E+03
Map4	**Mean**	9.31E+02	6.16E+02	3.55E+02	6.31E+02	5.99E+03	8.18E+02	2.51E+03	3.98E+03
	Std	1.73E+02	4.32E+01	2.19E+01	2.41E+01	9.77E+02	5.33E+01	7.34E+02	1.47E+03
	Best	7.14E+02	5.09E+02	3.03E+02	6.04E+02	3.88E+03	7.12E+02	2.63E+03	1.30E+03
Map5	**Mean**	9.22E+02	6.00E+02	3.32E+02	6.16E+02	5.14E+03	8.69E+02	3.12E+03	2.68E+03
	Std	1.83E+02	9.83E+01	1.95E+01	9.12E+00	1.25E+03	1.50E+02	4.00E+02	1.12E+03
	Best	7.65E+02	5.19E+02	3.04E+02	6.08E+02	3.66E+03	7.41E+02	2.60E+03	2.29E+03
Map6	**Mean**	1.07E+03	6.35E+02	3.48E+02	6.27E+02	7.54E+03	8.47E+02	4.16E+03	3.20E+03
	Std	3.37E+02	8.31E+01	2.95E+01	1.09E+01	2.60E+03	8.55E+01	6.37E+02	5.33E+02
	Best	7.47E+02	5.15E+02	3.04E+02	6.03E+02	5.07E+03	7.18E+02	2.39E+03	1.54E+03
Map7	**Mean**	9.07E+02	6.86E+02	3.57E+02	6.24E+02	7.13E+03	7.79E+02	3.27E+03	2.90E+03
	Std	1.24E+02	9.97E+01	3.19E+01	1.25E+01	1.65E+03	6.29E+01	7.21E+02	1.48E+02

Table 10. *Cont.*

Algorithm		R1	R2	R3	R4	R5	R6	R7	R8
Map8	**Best**	7.88E+02	5.09E+02	3.10E+02	6.07E+02	5.01E+03	7.07E+02	2.03E+03	2.30E+03
	Mean	1.00E+03	5.87E+02	3.34E+02	6.16E+02	6.79E+03	1.02E+03	3.20E+03	3.00E+03
	Std	1.54E+02	6.15E+01	1.49E+01	8.28E+00	1.09E+03	1.27E+02	7.09E+02	7.81E+02
Map9	**Best**	7.33E+02	5.48E+02	3.22E+02	6.12E+02	5.03E+03	7.11E+02	1.74E+03	1.22E+03
	Mean	9.23E+02	6.35E+02	3.51E+02	6.37E+02	7.23E+03	8.09E+02	2.50E+03	2.58E+03
	Std	1.57E+02	6.18E+01	2.18E+01	2.63E+01	1.69E+03	9.12E+01	4.51E+02	8.50E+03
Map10	**Best**	**7.28E+02**	**5.03E+02**	**3.03E+02**	**6.00E+02**	3.95E+03	7.08E+02	**1.13E+03**	**1.00E+03**
	Mean	**8.38E+02**	**5.79E+02**	**3.25E+02**	**6.14E+02**	5.32E+03	7.39E+02	**1.61E+03**	**2.46E+03**
	Std	**9.37E+01**	**7.65E+01**	**1.44E+01**	**1.26E+01**	1.08E+03	2.49E+01	**4.19E+02**	**3.50E+03**
Map11	**Best**	8.23E+02	5.25E+02	3.25E+02	6.17E+02	4.79E+03	7.38E+02	2.03E+03	1.56E+03
	Mean	9.49E+02	6.03E+02	3.57E+02	6.49E+02	6.55E+03	7.78E+02	2.63E+03	4.70E+03
	Std	1.29E+02	5.98E+01	3.07E+01	2.99E+01	1.34E+03	3.43E+01	5.46E+02	4.41E+03
Map12	**Best**	1.67E+03	7.10E+02	3.38E+02	6.20E+02	1.05E+04	1.21E+03	2.65E+03	4.09E+03
	Mean	2.23E+03	8.26E+02	3.89E+02	6.51E+02	1.94E+04	1.31E+03	2.97E+03	4.74E+03
	Std	2.91E+02	5.20E+01	2.54E+01	3.02E+01	3.35E+03	5.53E+01	2.04E+02	5.81E+02
SE-SCSO	**Best**	8.02E+02	5.62E+02	3.21E+02	6.21E+02	6.86E+03	7.42E+02	2.05E+04	1.37E+03
	Mean	1.68E+03	6.56E+02	3.82E+02	6.42E+02	1.03E+04	9.08E+02	2.82E+04	3.95E+03
	Std	5.73E+02	5.05E+01	3.96E+01	1.54E+01	3.73E+03	1.59E+02	7.16E+03	1.01E+03
CGWO	**Best**	8.47E+02	5.98E+02	3.26E+02	6.09E+02	2.97E+03	8.45E+02	1.09E+04	1.25E+03
	Mean	1.35E+03	1.07E+03	3.73E+02	6.50E+02	9.01E+03	1.08E+03	2.32E+04	4.36E+03
	Std	4.70E+02	4.46E+02	3.28E+01	3.15E+01	2.81E+03	2.52E+02	7.03E+03	2.45E+03
CMPA	**Best**	7.73E+02	5.12E+02	3.02E+02	6.08E+02	**2.85E+03**	**7.04E+02**	2.44E+03	1.70E+03
	Mean	9.71E+02	6.17E+02	3.34E+02	6.37E+02	**4.41E+03**	**7.34E+02**	2.96E+03	2.50E+03
	Std	2.04E+02	7.50E+01	2.38E+01	1.68E+01	**1.25E+03**	**3.01E+01**	4.34E+02	5.28E+02
CWOA	**Best**	8.68E+02	5.42E+02	3.44E+02	6.54E+02	1.94E+04	8.10E+02	8.87E+03	2.01E+03
	Mean	1.25E+03	6.04E+02	3.95E+02	6.82E+02	2.77E+04	9.76E+02	1.68E+04	2.97E+03
	Std	4.03E+02	5.27E+01	2.34E+01	1.44E+01	5.82E+03	9.38E+01	5.94E+03	7.21E+02

The optimum results obtained from the algorithms are bold and highlighted.

Our algorithm based on the Tent Map showed the best performance in numerical real-world problems. Based on this, the CSCSO-Map10 algorithm and the other 15 methods were compared with each other for the pairwise comparison test (Table 11). It is understood that the algorithm based on the Tent Map offers a very successful performance in these problems. It is understood that Map5, CMPA, Map8, and Map4 take place in the top five ranks, and the worst performance belongs to the SE-SCSO algorithm. This means that the proposed algorithm is better than the SE-SCSO and, in general, other chaos-based metaheuristic algorithms. In addition, the performance of each algorithm on these problems is shown in Figure 7, with a standard boxplot.

5.6. Local Optima and Convergence Curve

The CSCSO algorithm was developed to find more accurate candidate solutions in the exploration and exploitation processes of the SE-SCSO. In this regard, this section analyzes the diversity and convergence curves. The convergence behavior is an important tool in the analysis and evaluation of the results. The convergence behaviors of each algorithm on all functions are handled in 500 and 1000 iterations, and the movements and behaviors of each algorithm in two phases are examined. As presented in Figure 8, the convergence model and velocity of each algorithm are illustrated in a single run as the best-case sample. According to [98,99], unexpected changes in the movement of search agents are normal in the initial steps of optimization algorithms. As the results are analyzed, this change has a greater impact on the search space of the exploration process. However, these changes are normally reduced in final iterations for good use. As shown in Figure 8, the concept of chaos tries to balance the excess fluctuations in a way. In addition, these sudden changes cause agents to discover a possible solution and take advantage of it. At the same time, the exploitation of the discovered areas is expected. In this regard, a good algorithm is expected to do well in this phase (exploitation). Therefore, a convergence analysis of algorithms is considered an important tool to show how well each performs. While examining the convergence behavior of the algorithms, it is understood that the CSCSO algorithm has a good performance in the convergence behavior. This is because search agents can explore space in certain iterations and exploit them properly.

Table 11. Wilcoxon rank-sum test for the considered numerical real-world problems and the ranking of each algorithm.

Algorithms	R1	R2	R3	R4	R5	R6	R7	R8	Avg_Rank	Overall_Rank
Map1	10 (+)	10 (+)	8 (+)	8 (+)	6 (+)	8 (+)	10 (+)	10 (+)	8.750	10
Map2	7 (+)	11 (+)	12 (+)	11 (+)	10 (+)	5 (+)	13 (+)	5 (+)	9.250	11
Map3	2 (+)	12 (+)	5 (+)	4 (+)	7 (+)	10 (+)	12 (+)	11 (+)	7.875	8
Map4	6 (+)	6 (+)	9 (+)	7 (+)	4 (+)	7 (+)	3 (+)	13 (+)	7.000	5
Map5	4 (+)	3 (+)	2 (+)	3 (+)	2 (−)	11 (+)	7 (+)	4 (+)	4.500	2
Map6	12 (+)	9 (+)	6 (+)	6 (+)	12 (+)	9 (+)	11 (+)	9 (+)	9.250	11
Map7	3 (+)	14 (+)	10 (+)	5 (+)	9 (+)	4 (+)	9 (+)	6 (+)	7.375	7
Map8	11 (+)	2 (+)	3 (+)	2 (+)	8 (+)	14 (+)	8 (+)	8 (+)	7.000	5
Map9	5 (+)	8 (+)	7 (+)	10 (+)	11 (+)	6 (+)	2 (+)	3 (+)	6.500	4
Map10	1	1	1	1	3	2	1	1	1.375	1
Map11	8 (+)	4 (+)	11 (+)	13 (+)	5 (+)	3 (+)	4 (+)	15 (+)	7.875	8
Map12	16 (+)	15 (+)	15 (+)	15 (+)	12 (+)	16 (+)	6 (+)	16 (+)	13.875	15
SE-SCSO	15 (+)	13 (+)	14 (+)	12 (+)	15 (+)	12 (+)	16 (+)	12 (+)	13.625	14
CGWO	14 (+)	16 (+)	13 (+)	14 (+)	14 (+)	15 (+)	15 (+)	14 (+)	14.375	16
CMPA	9 (+)	7 (+)	4 (+)	9 (+)	1 (−)	1 (−)	5 (+)	2 (+)	4.750	3
CWOA	13 (+)	5 (+)	16 (+)	16 (+)	16 (+)	13 (+)	14 (+)	7 (+)	12.500	13

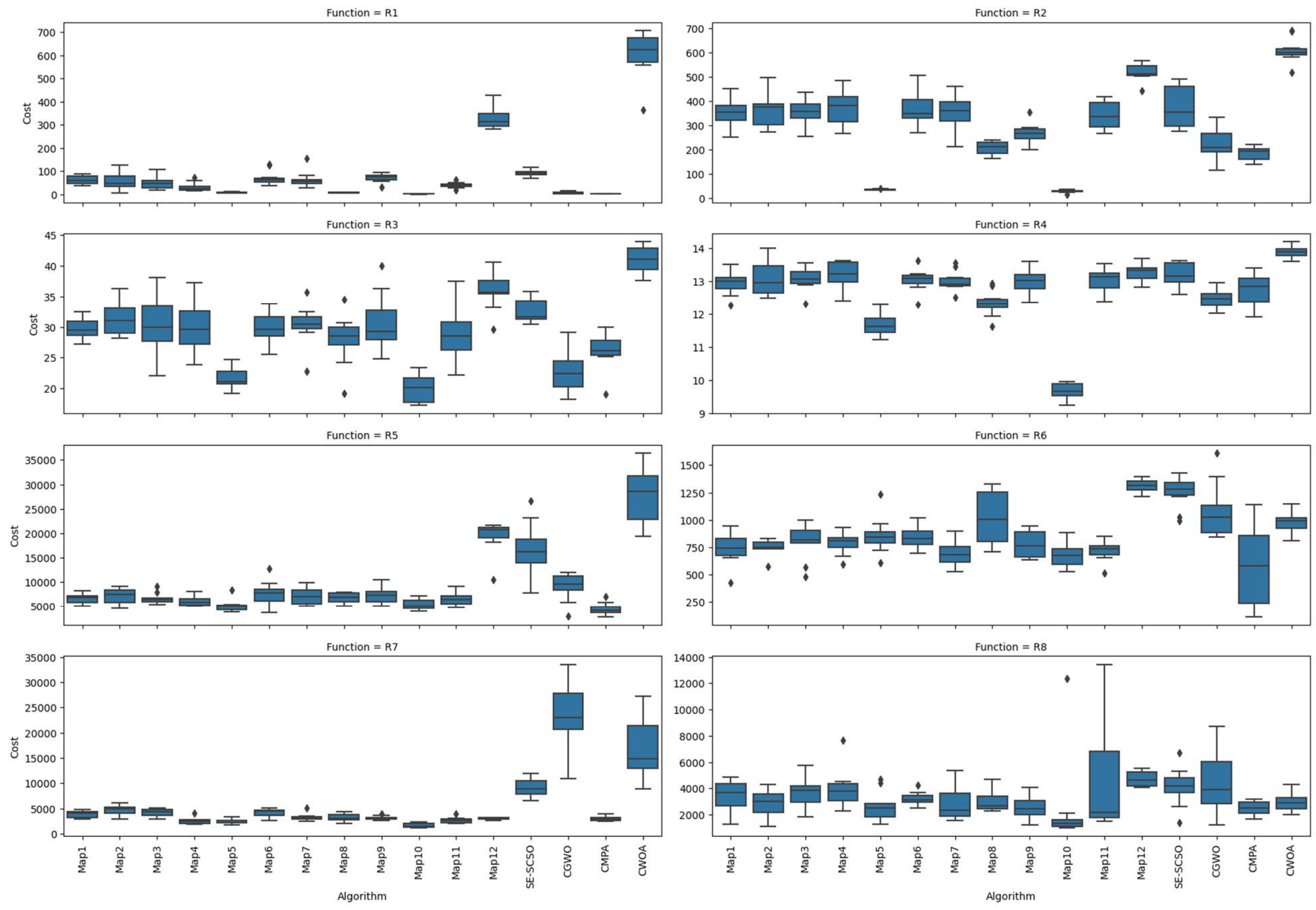


Figure 7. Boxplots of the results obtained by all algorithms on numerical real-world functions.

Besides that, the analyzes are also performed with Function Evaluations (FEs), and the results for some functions are presented in Figure 9. In this regard, there are two other parameters: population size and the number of iterations. This figure illustrates the fitness values of all calls, providing insight into the behavior of chaotic functions when making decisions between exploitation and exploration. The transition between phases occurs rapidly, resulting in varying fitness values for the agent during the exploration phase compared to the best fitness. The fluctuations in agent performance across calls demonstrate the evaluation of algorithms and their efforts to explore the search space for robust global optima.

5.7. Diversity Analysis

In general, the algorithm that prioritizes the exploitation phase will have a high convergence rate, but there is a risk of falling into the local optimum trap. On the other hand, since the algorithm that concentrates more on the exploration phase navigates the search space more, the probability of finding the global answer increases. However, the convergence speed may be low. Therefore, a reasonably working algorithm tries to be balanced between these two phases. One of the techniques used for balance analysis between these two phases is the diversity metric. The average distance between all solutions during iterations is shown by the diversity curves, as shown in Figure 10 for some functions. The diversity calculation was inspired by some studies [100,101]. The diversity is obtained according to Equation (11), inspired by these studies.

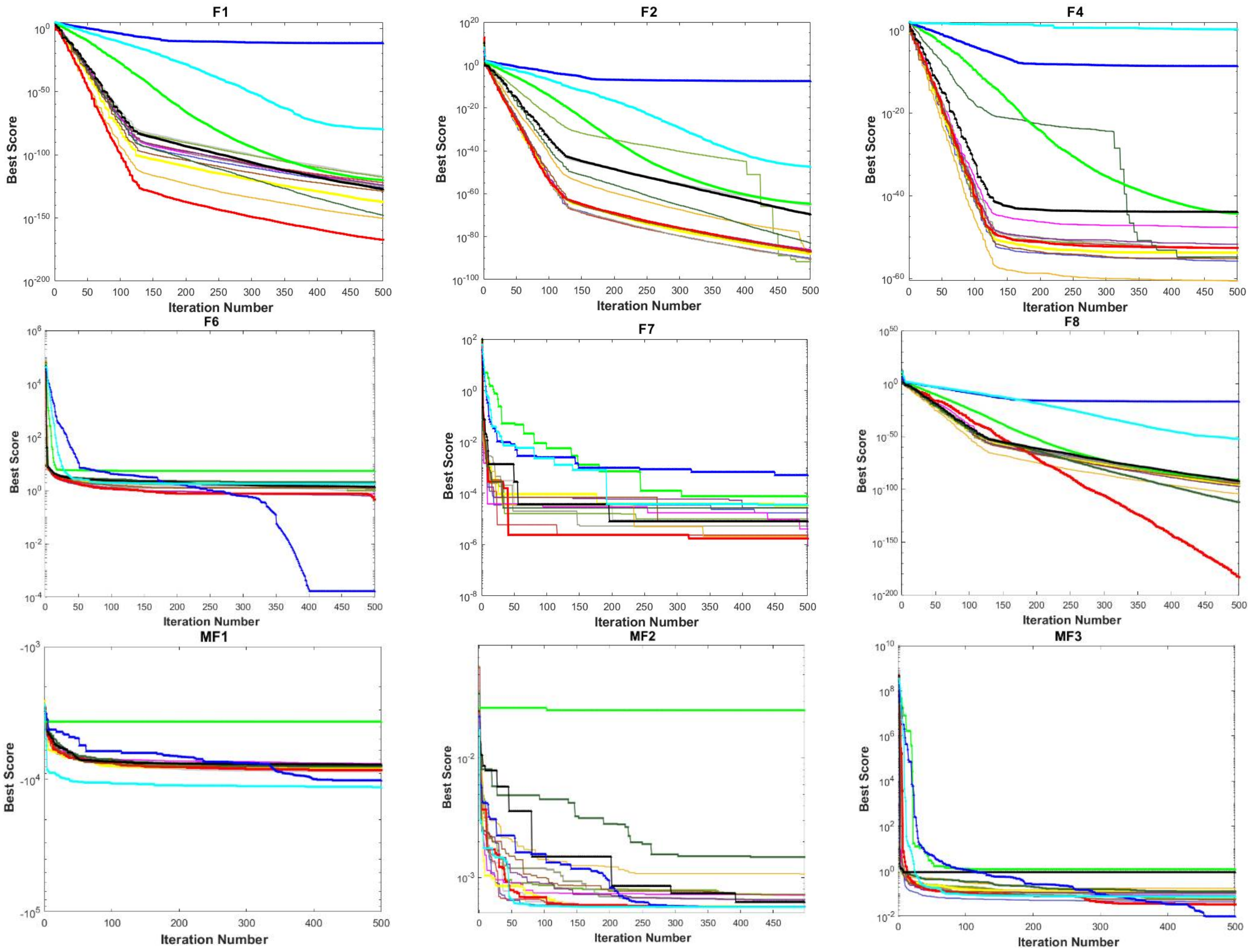


Figure 8. *Cont.*

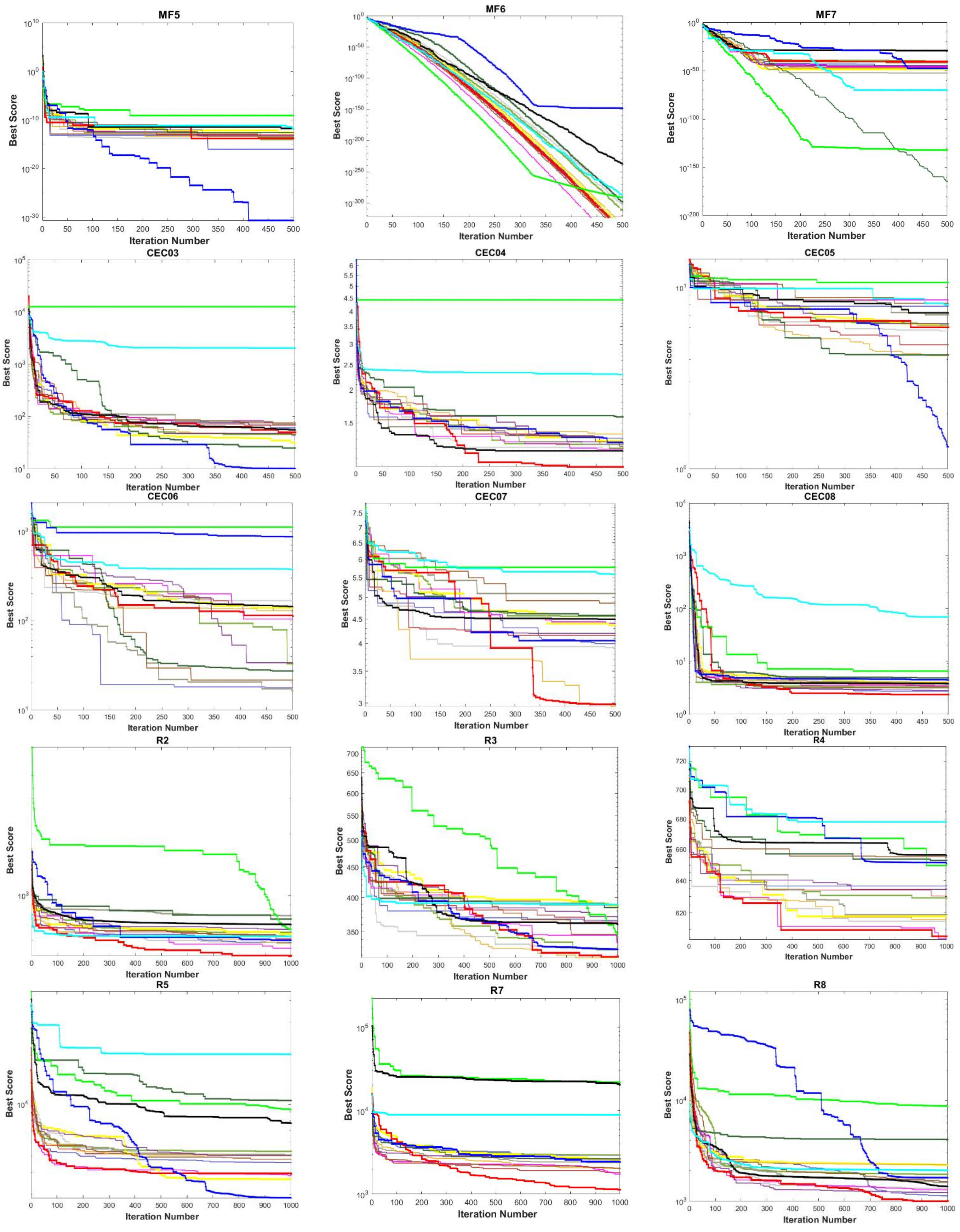


Figure 8. *Cont.*

Figure 8. Convergence curve analysis of each algorithm on some benchmark functions.

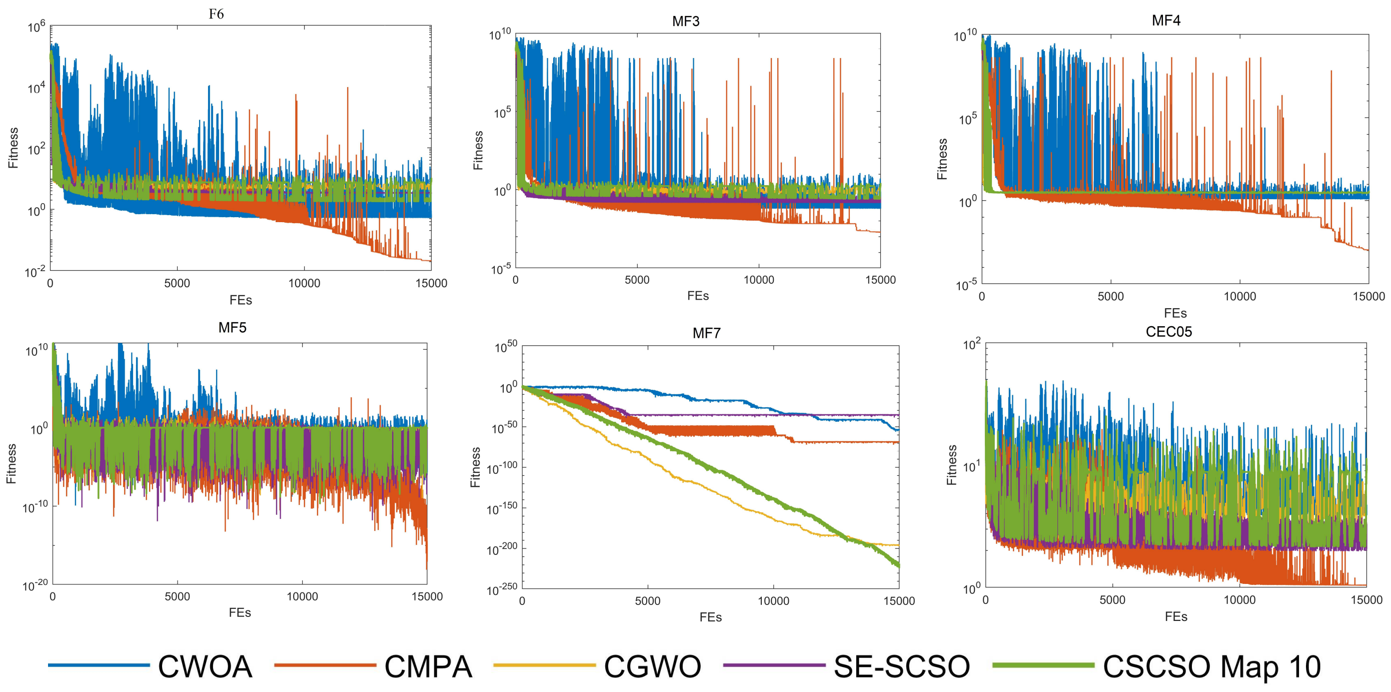


Figure 9. Performance analysis of each algorithm based on Function Evaluations (FEs).

$$\forall\, i \in I \;\rightarrow\; Diversity_i = \frac{1}{D}\sum_{d=1}^{D} \frac{1}{A}\sqrt{\sum_{a=1}^{A}\left(x_{i.d,a} - \overline{x}_{i,d}\right)^2} \tag{11}$$

where I, D, and A are sets of iterations, dimensions, and agents, respectively; x_{da} is the position of the dth dimension of the ath search agent; and $\overline{x}_d$ defines for the mean of dimension d in the total popluation.

Thanks to the chaos strategies, the population's diversity level is preserved. Thus, the probability of bias for local optimal solutions during the search is reduced. The CSCSO algorithm based on the chaos approach focuses on the population's diversity level, reducing the probability of falling into the local optima trap during the search. During the initial iterations, there is a significant average distance between the search agents, which progressively decreases as the number of iterations increases. The ratio between exploitation and exploration significantly impacts the diversity of agents across iterations. Exploration disperses agents to explore the global search space in swarm optimization algorithms, while exploitation converges agents towards local optima to enhance fitness. The observed fluctuations in diversity, as shown in Figure 10, stem from the swift phase changes occurring during the process. Figure 10 illustrates that, across most test functions, the CSCSO algorithm consistently exhibits higher average distances per iteration than the SCSO. This observation indicates that the proposed algorithm possesses superior search capabilities within the search space, particularly when exploring novel search regions compared to SCSO.

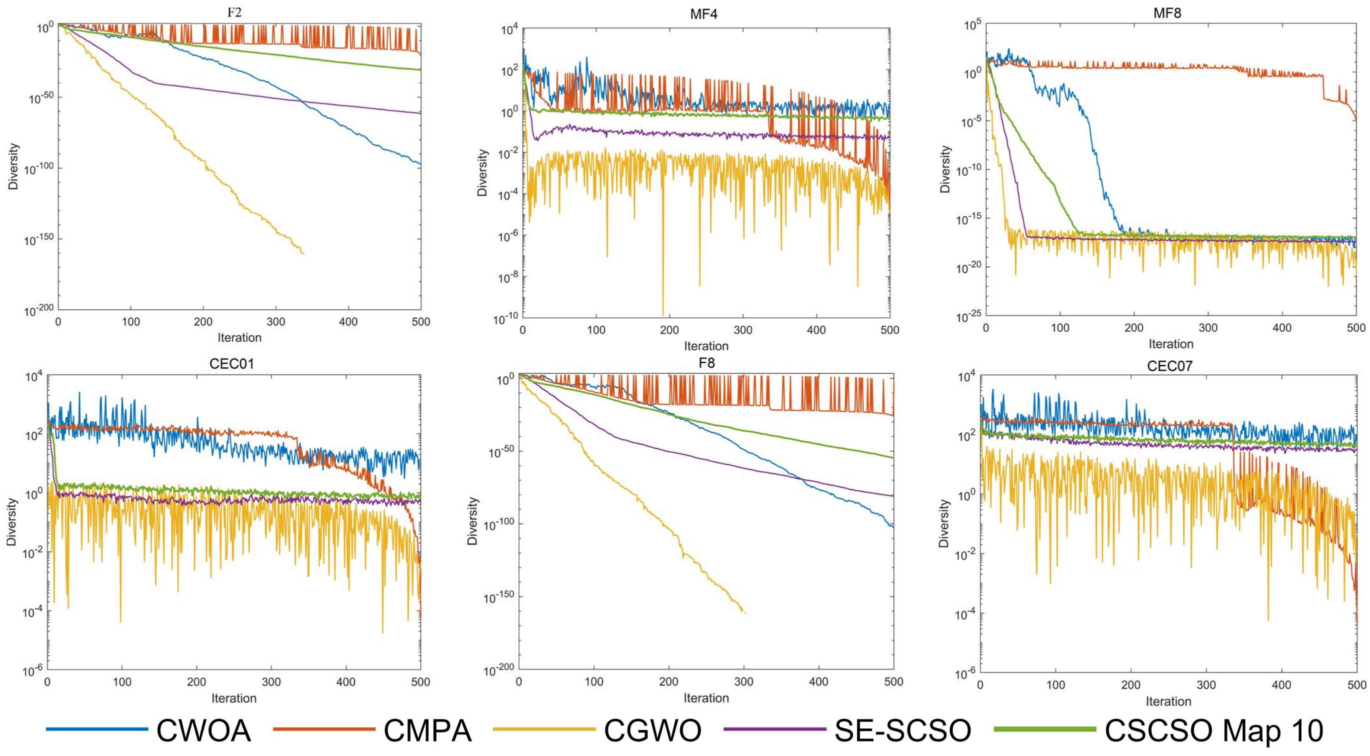


Figure 10. Diversity analysis of each method on some benchmark functions.

5.8. Exploration and Exploitation Analysis

As described in the introduction and analysis sections of the work, exploration and exploitation behaviors are illustrated in Figure 11 concerning the numbers of FEs, as these are two important aspects of an effective algorithm. Here, analyses were made from two different perspectives. Here, the effect of each map implemented in CSCSO is discussed in relation to an example function in exploration and exploitation phases. While other algorithms focus more on the exploitation process after a while, the proposed algorithm continues a certain and stable trend. Indeed, while some maps give weight to the exploration phase, it is possible to follow the opposite results in some cases. Considering that the algorithm behaves in a balanced manner due to its nature, those whose ratios in the two phases are close to each other (in all functions considered) were able to find better results in general. In this subsection, the test result on the R_7 function is presented for all maps. While the proposed algorithm (e.g., Map10) focuses on exploration in the initial stages, exploitation is becoming more and more important. On some maps, the situation is reversed. However, in general, their close ratio shows that the proposed algorithm behaves in a balanced way. Accordingly, the results based on Map10, emphasizing exploration and exploitation in two phases, were also better than the others. It is worth noting that only one is presented in a relevant way since the behavior and rate of the exploitation phase is the opposite of exploration.

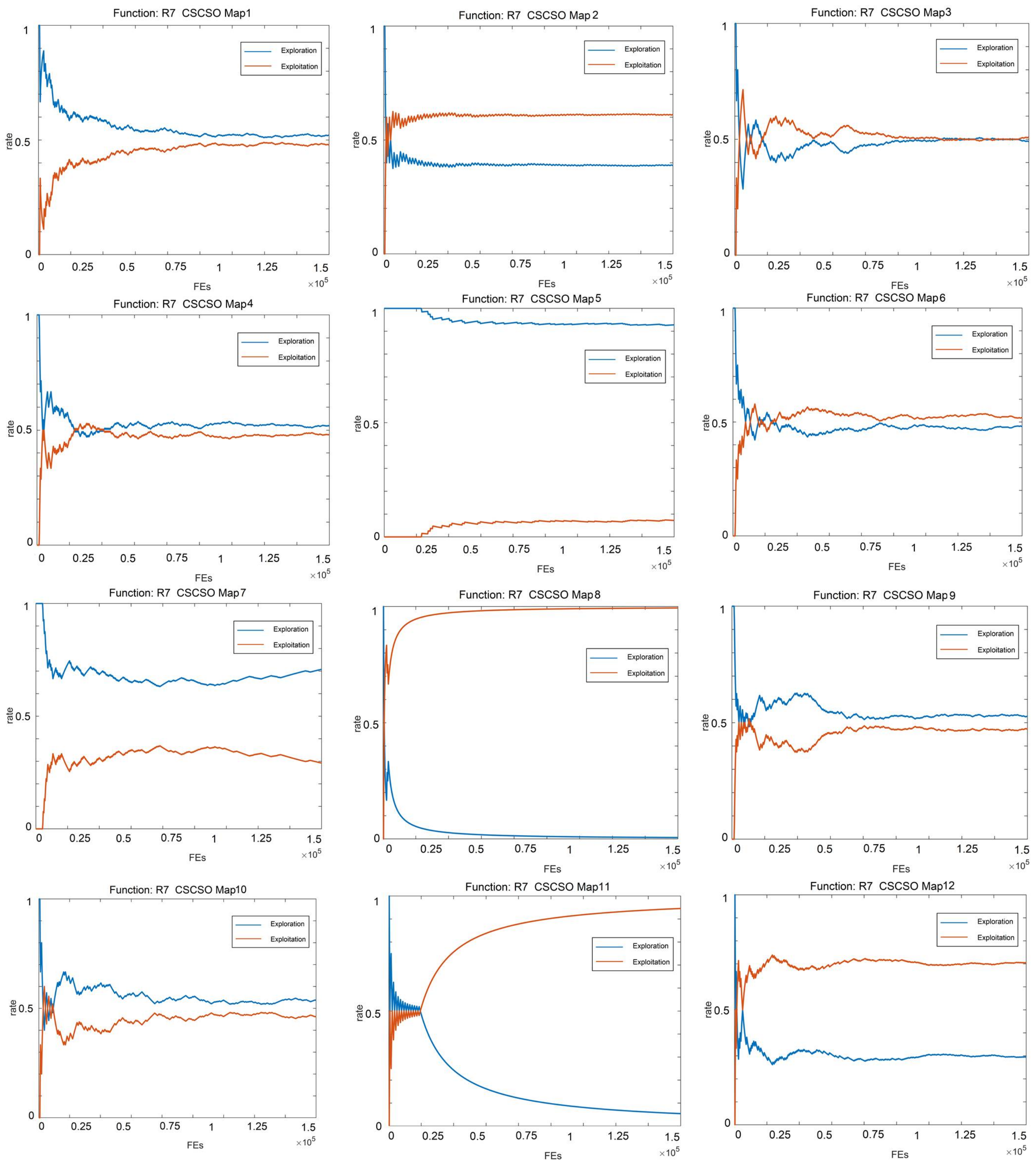


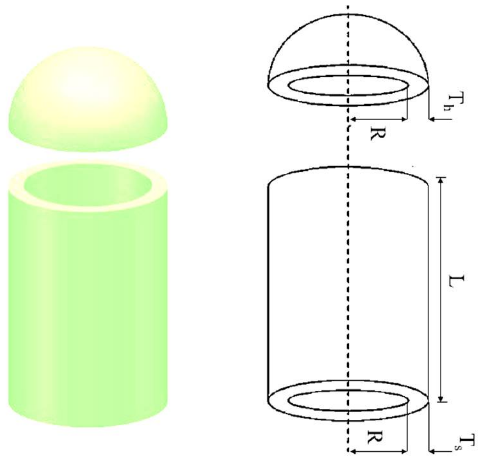
Figure 11. Exploration and exploitation plots of each map on R$_7$ function.

6. CSCSO in Constrained Engineering Problems

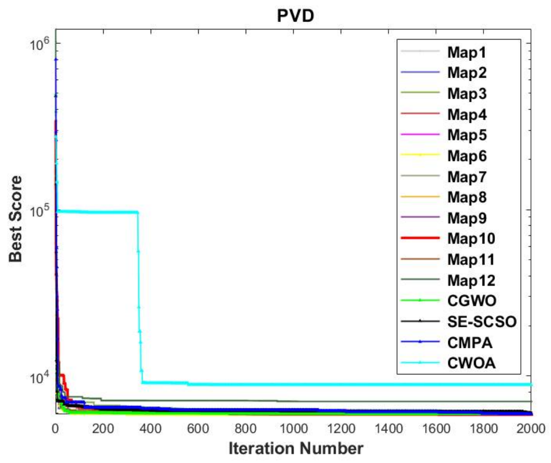
In this study, in the performance analysis of the proposed algorithm, in addition to various complex test functions in the previous section, constrained engineering problems are used in this section. In this section, empirical evaluations are made by considering five of these real-world applications. These problems are limited and difficult in pressure vessel design [102], gear train design [103], compression spring design [104], speed reducer design [105], and three-bar truss design [106] applications. Simulations and analyses of all problems were carried out under the same conditions.

6.1. Results and Discussion in Pressure Vessel Design Problem

The first example studied in this section is the pressure vessel design problem. It is a mixed−integer problem, and its primary objective is minimizing the total cost of a cylindrical pressure vessel, including welding, material, and forming [2,102]. For this purpose, the optimum values of four variables defined in the problem should be found. These variables are the thickness of the head (T_h), the thickness of the shell (T_s), the inner radius (R), and the length of the cylindrical section without considering the head (L). The visual representation of the problems is given in Figure 12a. The explanation of the problem by using with mathematical equations is given in Table A5. The proposed algorithm was tested separately with 12 maps, and the results were compared with four other algorithms. The results are presented in Table 12, and the convergence curve is in Figure 12b.



(**a**)



(**b**)

Figure 12. (**a**) The pressure vessel design problem. (**b**) Convergence curve.

Table 12. Experimental results of the pressure vessel design problem (pop = 30; iter = 2000).

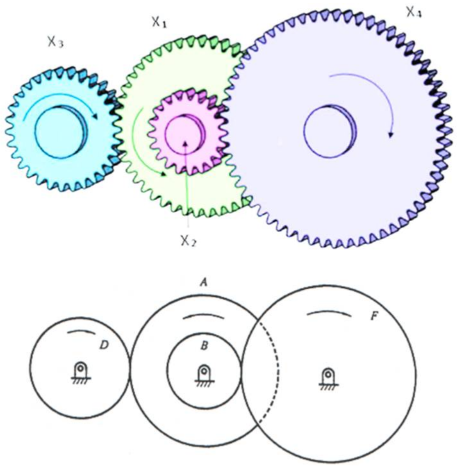
Algorithms	**Optimum Variables**				**Optimum Cost**	**Overall_Rank**
	T_s	**T_h**	**R**	**L**		
Map1	0.8840296	0.4376542	45.72797	136.3239	6103.5354	14
Map2	0.8645994	0.4272778	44.76719	146.2359	6055.5355	13
Map3	0.7786206	0.3862715	40.3372	200	5896.1887	6
Map4	0.8578314	0.4246596	44.41697	149.9077	6042.4655	12
Map5	0.7791666	0.385388	40.3693	199.3147	5888.1342	2
Map6	0.778796	0.3889989	40.32789	199.8965	5901.5966	7
Map7	0.7784384	0.3892492	40.33261	200	5902.7252	8
Map8	0.7795517	0.3854967	40.38957	199.2261	5892.6565	5
Map9	0.7803899	0.3866241	40.4314	198.4661	5892.4868	4
Map10	**0.7781687**	**0.3846492**	**40.31962**	**200**	**5885.3328**	**1**
Map11	0.7801906	0.3894495	40.41601	198.8482	5904.9624	10
Map12	1.18462	0.587487	61.3585	27.8221	7023.4302	15
SE-SCSO	0.8292	0.43172	42.9236	168.8026	5986.1419	11
CGWO	0.7887976	0.3900441	40.86958	192.4996	5904.6293	9
CMPA	0.7807125	0.3863639	40.45044	198.2424	5892.3573	3
CWOA	1.209648	0.6385708	47.10937	123.0912	8823.5514	16

The best algorithm is shown in bold and highlight.

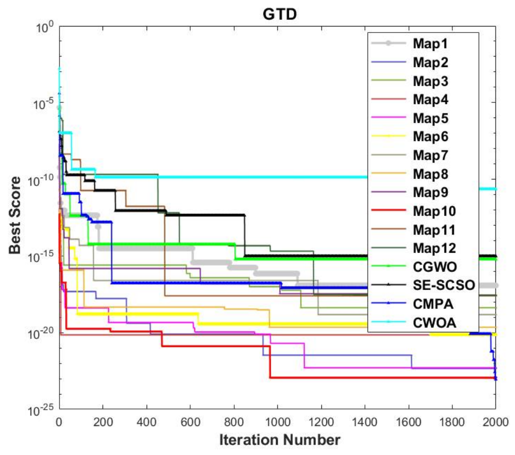
According to the results, the proposed algorithm based on the Tent Map achieved the best performance compared to other maps and other algorithms. The sinusoidal-based CSCSO took second place. The point that draws attention here is that the CMPA based on the Guass/Mouse Map is better than our other 10 maps and ranks third.

6.2. Gear Train Design Problem

The widely researched gear design problem [103] is the second problem addressed in this section. This problem aims to minimize the cost of the gear ratio. In this problem, there are four variables, matching the number of teeth of the gearwheels. These are called n_a, n_b, n_f, and n_d. They are denoted here as x_1, x_2, x_3, and x_4 [107]. Their range of all is limited to between 12 and 60. A two− and three−dimensional illustration of this problem is shown in Figure 13a. This problem is formulated mathematically in Table A5. The proposed algorithm was tested separately with 12 maps, and, also, the results were compared with four other algorithms. The results are presented in Table 13, and the convergence curve is shown in Figure 13b.



(**a**)



(**b**)

Figure 13. (**a**) The gear train design problem. (**b**) Convergence curve.

Table 13. Experimental results of the gear train design problem (pop = 30; iter = 2000).

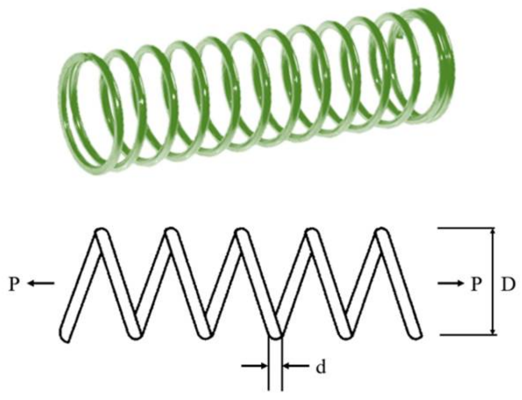
Algorithms	Optimum Variables				Optimum Cost	Overall_Rank
	X_1	X_2	X_3	X_4		
Map1	60	12	12	16.63	1.189053441454946E−17	13
Map2	48.02	21.59	17.76	55.34	4.517400503839139E−23	3
Map3	43.73	12.08	29.86	57.17	4.218489743140509E−19	9
Map4	60	42.02	12	58.25	7.167907310992734E−21	5
Map5	50.74	35.36	12.42	60	5.037306687704821E−23	4
Map6	16.90	12	12.02	59.14	7.526271461486777E−21	6
Map7	49.40	13.46	12	22.66	1.526716660992023E−19	8
Map8	30.44	17.77	12	48.57	2.230916200498150E−20	7
Map9	35.80	12	12.28	28.53	2.866133049573352E−18	12
Map10	52.96	19.86	16.31	42.40	1.122944971843512E−23	2
Map11	28.58	12.18	13.50	39.88	2.528126808979344E−18	10
Map12	48.97	17.13	20.54	49.81	2.689792357227320E−18	11
SE-SCSO	60	12.08	12.86	32.77	5. 204244123165204E−16	14
CGWO	60	13.62	14.69	23.10	5.925901619563185E−16	15
CMPA	**45.60**	**12.23**	**12.85**	**23.88**	**8.725285232783098E−24**	**1**
CWOA	35.28	12	12	28.29	2.226642147367809E−11	16

The best algorithm is shown in bold and highlight.

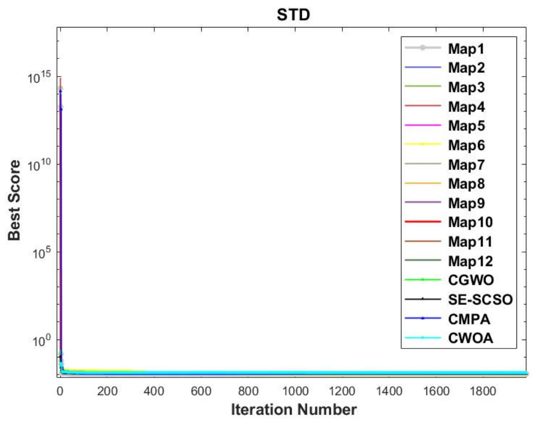
When the results are analyzed, it is understood that the most optimal values for the solution of this problem belong to the Chebyshev-based CMPA algorithm. In second place was the CSCSO algorithm based on the Tent Map. It was observed that the Circle-based CSCSO algorithm took third place. The weakest performance, like the previous problem, belongs to the CWOA algorithm.

6.3. Tension/Compression Spring Design Problem

The third problem examined in this section is the tension/compression spring design problem. Its primary objective is reducing the spring's mass based on four constraints and three variables [104]. Its variables are wiring diameter (*d*), mean coil diameter (*D*), and the number of active coils (*N*). Moreover, the constraints of the problem are surge frequency, minimum deflection, and shear stress. The two− and three−dimensional representations of the problems are given in Figure 14a. Its mathematical equations are given in Table A5. The proposed algorithm was tested separately with 12 maps, and, also, the results were compared with four other algorithms. The results are presented in Table 14, and the convergence behavior of each algorithm is in Figure 14b.



(**a**)



(**b**)

Figure 14. (**a**) The tension/compression spring design problem. (**b**) Convergence curve.

Table 14. Experimental results of the tension/compression spring design problem (pop = 30; iter = 2000).

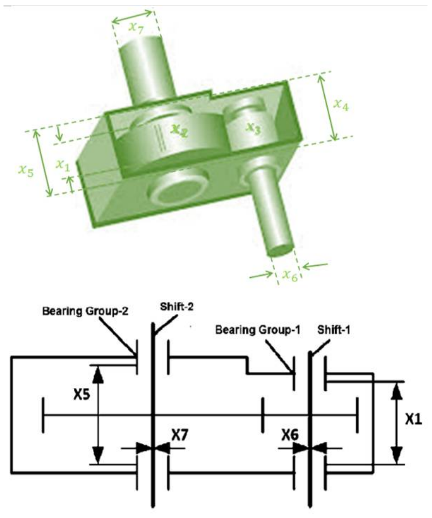
Algorithms	Optimum Variables			Optimum Cost	Overall_Rank
	d	D	N		
Map1	0.05	0.348873	10.5678	0.010961	2
Map2	0.05	0.348495	10.6104	0.010987	14
Map3	0.05	0.348725	10.5801	0.010967	11
Map4	0.05	0.348818	10.5786	0.010969	13
Map5	0.0500054	0.349008	10.56	0.010961	2
Map6	0.05	0.348878	10.5698	0.010963	9
Map7	0.053962	0.44723	6.8114	0.011475	15
Map8	0.05	0.348771	10.5794	0.010968	12
Map9	0.05	0.348881	10.5656	0.010961	2
Map10	**0.05**	**0.348908**	**10.5634**	**0.010957**	**1**
Map11	0.05	0.348894	10.5676	0.010962	7
Map12	0.05	0.348881	10.5661	0.010961	2
SE-SCSO	0.05	0.356471	10.5809	0.010962	7
CGWO	0.05	0.348853	10.5683	0.010961	2
CMPA	0.05	0.348915	10.5622	0.010964	10
CWOA	0.057215	0.54039	4.8765	0.012165	16

The best algorithm is shown in bold and highlight.

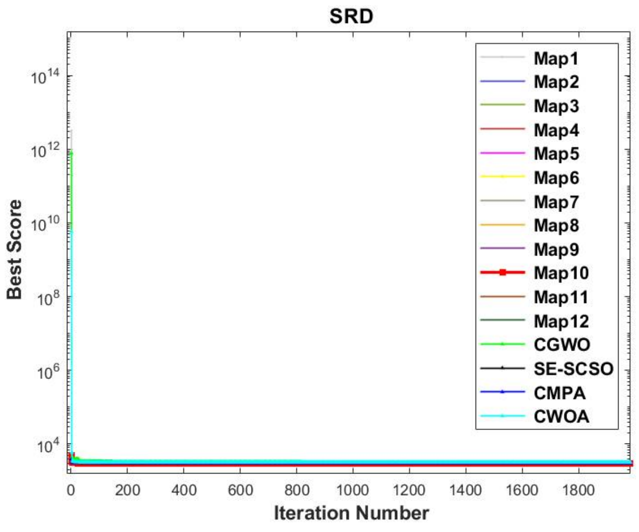
According to the results, the CSCSO-Map10 found the best solution than others. The SE-SCSO algorithm took second place. In this problem, Chebyshev, Sinusoidal, Logistic, and Quadratic maps found similar results in CSCSO. Therefore, Map1, Map5, Map9, and Map12 jointly took third place. The weakest performance appears to belong to the CWOA algorithm. It should also be emphasized that the Singer Map and the CSCSO algorithm found the weakest results.

6.4. Speed-Reducer Design Problem

The speed-reducer problem is the fourth problem considered in this section. The main goal of this mechanical engineering problem is minimizing the total weight of a speed reducer under 11 constraints [105]. Four of these constraints are linear inequality constraints, and the other seven are nonlinear. The linear constraints are the bending stress of the gear teeth, transverse deflections of the shafts, surface stress, and stresses in the shafts. In addition, this problem has seven variables, the face width b(x_1), the module of teeth m(x_2), the number of teeth in the pinion z(x_3), the length of the first shaft between bearings l1(x_4), the length of the second shaft between bearings l2(x_5), the diameter of first shafts d1(x_6), and the diameter of second shafts d2(x_7). The 3D and 2D models of this problem are presented in Figure 15a. Moreover, the mathematical equations of the problem are given in Table A5. The experimental results are presented in Table 15. The convergence behavior of each algorithm is also exhibited in Figure 15b.



(**a**)



(**b**)

Figure 15. (**a**) The speed reducer design problem. (**b**) Convergence curve.

According to the results, CSCSO-Map5 (Sinusoidal Map) and CSCSO-Map10 (Tent Map) ranked first and second. The worst performance of the proposed algorithm was obtained with the Chebyshev Map. When comparing all algorithms, we found that the weakest performance belongs to the CWOA algorithm. The proposed algorithm in this problem is understood to be better than other chaotic-based metaheuristic algorithms and SE-SCSO, like the previous three problems.

6.5. Three-Bar Truss Design Problem

The last constrained engineering problem addressed is the three-bar truss problem. The main objective of this two−dimensional problem is to minimize relevant weights subjected to three non−linear constraints. In this problem, two design variables, A_1 and A_2, are defined. These restraints are stress, deflection, and buckling. This problem is reported in [106] as a highly constrained search space. This test problem is described mathematically in Table A5. The structure of the three-bar truss is presented in two dimensions in Figure 16a.

Four algorithms from the literature were used for the performance analysis of the CSCSO algorithm. In addition, the famous 12 maps are applied separately to the problem to clearly understand the effect of the proposed algorithm, like other problems. The experimental results are presented in Table 16. In this particular problem, convergence behaviors similar to those in other problems are presented (Figure 16b).

Table 15. Experimental results of the speed-reducer design problem (pop = 30; iter = 2000).

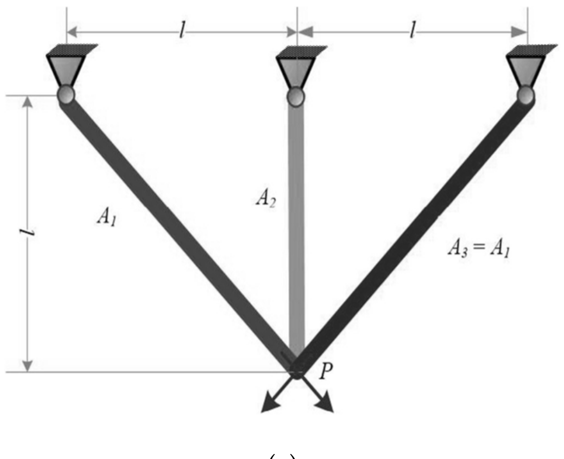
Algorithms	Optimum Variables							Optimum Cost	Overall_Rank
	X_1	X_2	X_3	X_4	X_5	X_6	X_7		
Map1	3.50	0.70	17	7.39	8.11	3.35	5.29	3.002499198459249E+03	14
Map2	3.50	0.70	17	7.55	7.76	3.36	5.29	2.999239924699496E+03	11
Map3	3.50	0.70	17	7.30	7.73	3.35	5.29	2.995209823639083E+03	5
Map4	3.50	0.70	17	7.32	7.82	3.36	5.29	2.997467922122005E+03	9
Map5	**3.50**	**0.70**	**17**	**7.30**	**7.721**	**3.35**	**5.29**	**2.994471067842177E+03**	**1**
Map6	3.50	0.70	17	7.30	7.76	3.35	5.29	2.995138822665665E+03	4
Map7	3.50	0.70	17	7.30	7.94	3.35	5.29	2.999340737545768E+03	12
Map8	3.50	0.70	17	7.30	7.83	3.36	5.29	2.996572732895894E+03	7
Map9	3.50	0.70	17	7.30	7.76	3.36	5.29	2.996861016039304E+03	8
Map10	3.50	0.70	17	7.30	7.723	3.35	5.29	2.994508089032852E+03	2
Map11	3.50	0.70	17	7.38	7.72	3.38	5.29	2.999630362959565E+03	6
Map12	3.51	0.70	17	7.30	8.30	3.37	5.29	2.999788897881505E+03	13
SE-SCSO	3.49	0.70	17	8.29	7.84	3.36	5.29	2.996944440275123E+03	10
CGWO	3.50	0.70	17	8.28	8.27	3.89	5.34	3.016583145706655E+03	15
CMPA	3.50	0.70	17	7.30	7.722	3.35	5.29	2.994516229441376E+03	3
CWOA	3.58	0.70	17	8.05	7.99	3.79	5.29	3.162932713322597E+03	16

The best algorithm is shown in bold and highlight.

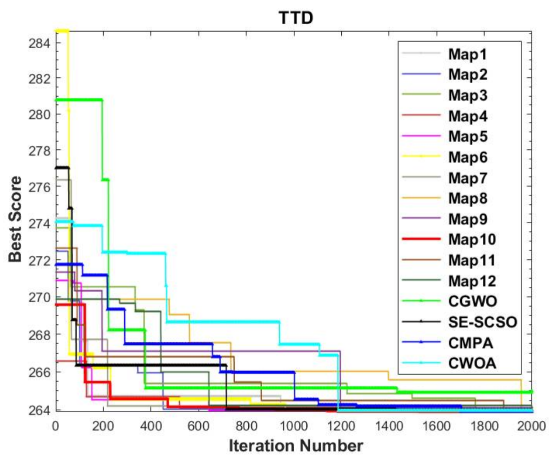
Table 16. Experimental results of the three-bar truss design problem (pop = 30; iter = 2000).

Algorithms	Optimum Variables		Optimum Cost	Overall_Rank
	X_1	X_2		
Map1	0.788196392484220	0.409924295610334	2.639280351739794E+02	3
Map2	0.791395442262100	0.402027178648411	2.640431513943041E+02	12
Map3	0.781140214144988	0.430879816857226	2.640277986774958E+02	11
Map4	0.786386913515764	0.416828890667142	2.641066967400567E+02	14
Map5	0.779970952119523	0.433641291581441	2.639732289070413E+02	8
Map6	0.784899761302183	0.420208383161735	2.640240158235639E+02	10
Map7	0.786634196693848	0.414409164378916	2.639346663560726E+02	4
Map8	0.799339104565123	0.379137312273734	2.640009717496060E+02	9
Map9	0.801186907090069	0.374945296375728	2.641044076380784E+02	13
Map10	**0.789803744838697**	**0.405067253792508**	**2.638969588920399E+02**	**1**
Map11	0.785834430117980	0.416708708381071	2.639384126086230E+02	5
Map12	0.768942064885116	0.467384746170158	2.642281339849564E+02	15
SE-SCSO	0.792810428422592	0.397186898447411	2.639593418979518E+02	7
CGWO	0.797839392122505	0.384069666300697	2.64904295142864E+02	16
CMPA	0.788757980922057	0.408038963897476	2.638983431997460E+02	2
CWOA	0.781157281506678	0.430088660802161	2.639535104508645E+02	6

The best algorithm is shown in bold and highlight.



(**a**)



(**b**)

Figure 16. (**a**) The three-bar truss design problem. (**b**) Convergence curve.

The results presented in Table 16 illustrate that the CSCSO algorithm is more dominant than the others. This algorithm achieved the best results with the least number of Function Evaluations. The CSCSO achieved more successful results with Tent and Chebyshev maps. It was noted that the CMPA algorithm, which ranks second, is based on the Chebyshev Map. In this problem, the weakest performer (Rank 16) was CSCSO based on the Quadratic Map. This result indicates that this map is not suitable for this problem. However, it should be emphasized that the results obtained are very close to each other.

7. CSCSO in Constrained Social Sciences-Based Problems

In the last section of this study, firstly, the performance of the proposed algorithm is analyzed on the p-median problem in the field of management and business in social science. The p-median problem is one of the most well-known NP-hard problems in the category of facility location problems and focuses on a facility layout and assignment model [93,108]. Many studies have been carried out in the literature to solve this problem [109]. The p-median problem falls under the category of minimum sum facility location problems. In [110], a formulation of the problem focusing on the nodes of optimum placement in a network with triangular inequality was made. The most basic version of the p-median problem, also called the 1-median problem, is the model that aims to determine the location of a median facility that will serve all demand points on the network. In this problem where only one establishment location is selected, the aim is to minimize the total cost. Problems with more than one median point are called p-median problems. The p-median problem is concerned with placing p facilities, which will serve n-demand points, on the network in a way that minimizes the weighted cost of the entire system [111,112]. In this problem, the said cost may be the distance, time, or monetary amount between the demand points and the service points. The assumptions of the p-median problem are to find the number of facilities to be opened without a time limit based on a linear relationship between cost and distance. In the definition of this problem, there is no capacity limit and facility opening cost for the facilities serving. In addition, all facilities have equal features, and customer demands are constant. This problem, which has certain points where facilities can be opened, is of the discrete type. In the p-median problem assumed in the discrete structure, the n nodes and p facilities to be opened consist of a combination situation $\binom{p}{n}$, and its mathematical model is given in Equation (12):

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} a_i d_{i,j} z_{i,j} \tag{12}$$

This fitness function aims to minimize the weighted total cost between the facilities and demand points. Here, n represents the total number of demand points, a_i is the demand at point (node) i, and d_{ij} is the shortest distance between point i and point j. The variable p represents the number of facilities (median) to be placed. In this problem, there are four (4) constraints that are presented in Equation (13):

$$\sum_{j=1}^{n} z_{i,j} = 1 \ \forall i \ i,j = 1,2,\ldots,n \tag{13}$$

$$z_{i,j} \le y_j \ \forall i,j \ i,j = 1,2,\ldots,n$$
$$\sum_{j=1}^{n} y_j = p$$
$$z_{i,j},\ y_j \in \{0,1\} \ i,j = 1,2,\ldots,n$$

The total z (first constrained) expresses the condition that all demands of the demand point are met from only one facility. Indeed, each demand point is assigned to only one facility. The second constraint indicates the condition of not assigning a demand point to a facility that is not open. In other words, an attempt is made to ensure consistency here. The third constraint limits the number of facilities to be opened and serviced to the p. The z and y can take values between zero and one for all i and j values. In addition, in this problem, the decision variables are defined as follows (Equation (14)). An exemplary model network representing this problem is presented in Figure 17.

$$z_{i,j} = \begin{cases} 1 \ if \ customer \ i \ is \ assigned \ to \ facility \ j \\ 0 \ otherwise \end{cases} ; \ y_j = \begin{cases} 1 \ if \ a \ facility \ is \ opened \ at \ point \ j \\ 0 \ otherwise \end{cases} \tag{14}$$

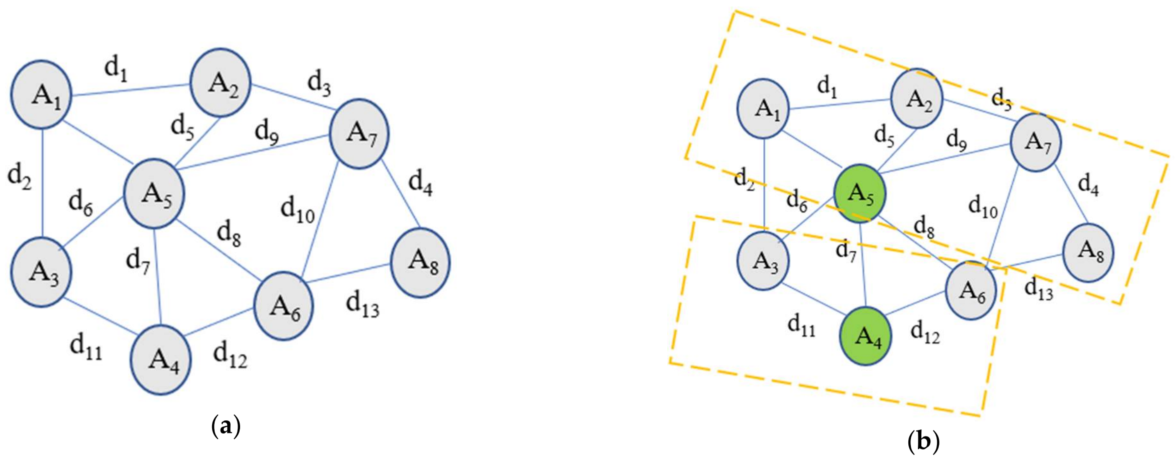


Figure 17. (**a**) A sample of p-median problem when n = 8. (**b**) Selected nodes for sample median facilities and demand points assigned to facilities when n = 8 and p = 2 (A4 and A5).

The number of points and the distances between them are inspired by [113,114]. In this study, the results of the proposed algorithm on this problem represented by the matrix are presented in Table 17. According to the results of the comparisons with other metaheuristic algorithms, it is understood that the CSCSO algorithm performs better. As the value of p increases and, naturally, the number of nodes, it is understood that the proposed algorithm finds significantly better results in all maps. Based on this, it appears that it will perform even better in larger and more complex scenarios.

The other problem addressed in this study is based on water resources management. This problem, especially the management and allocation of groundwater, is a social problem that directly triggers people's lives, although it is considered an engineering subject in the literature [115,116]. This problem is largely related to climate change and the lack of a sustainable water management perspective. It is worth emphasizing that climate change is the result of extreme humanitarian interventions [117]. At this point, it is especially important that natural resources are used through a circular economy, not a linear economy (Figure 18a). In this regard, the United Nations Development Agency draws attention to this problem in its universally accepted sustainable development goals [118]. This agency states that it is essential to consider 17 goals together in order to create a holistic sustainable development and valid until 2030 (Figure 18b). Internationally, schemes such as the Green New Deal and the European Union Green Deal are promising initiatives so that people can live equally in a cleaner and greener world [119]. Here, we evaluate the role of the metaheuristic approach in solving various constrained problems, as well as examine the impact from the social sciences dimension. Therefore, by addressing this general problem, we analyze and discuss it from a multidisciplinary perspective.

Table 17. Experimental results of the p-median problem (n = 200, pop = 100, and iter = 500).

Algorithms	**Optimum Cost for Various *p*-Values**				**Overall_Rank**
	5	**10**	**15**	**20**	
Map1	9999.89	7909.88	6418.91	5718.66	2
Map2	10,377.46	8609.39	7755.74	6959.55	10
Map3	10,107.83	8449.07	7705.16	6902.51	8
Map4	10,227.71	8556.44	7742.02	6928.42	9
Map5	10,088.39	8247.83	7371.97	6624.27	4
Map6	10,103.42	8437.41	7655.79	6821.93	7
Map7	10,074.11	8369.32	7607.65	6801.49	5
Map8	10,017.78	8011.54	7326.14	6401.74	3
Map9	10,101.22	8409.63	7645.08	6807.37	6
Map10	**9834.16**	**7482.01**	**6184.11**	**5426.14**	**1**
Map11	10,155.11	8594.75	7755.82	6960.04	11
Map12	10,159.77	8604.81	7757.96	6961.19	12
SE-SCSO	10,159.11	8597.37	7759.44	6969.44	13
CGWO	11,622.41	10,222.45	9086.83	8372.57	15
CMPA	10,101.77	8456.48	7769.66	7241.28	14
CWOA	12,205.65	10,926.93	9888.89	9174.29	16

The best algorithm is shown in bold and highlight.

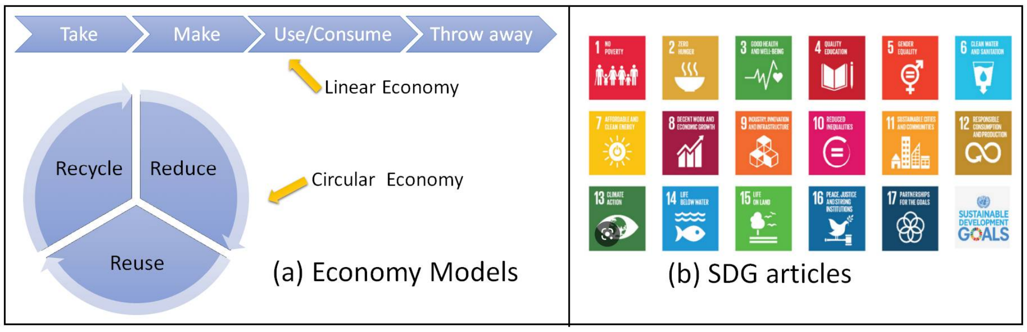


Figure 18. (**a**) Linear vs. circular economy. (**b**) The 17 Sustainable Development Goals (SDGs) [120].

Water resource management becomes a complex problem due to the stochastic nature of the inflow, as well as various demands, such as domestic use, agriculture, farming, and downstream environmental flow. Considering that the human body consists of an average of 60% water and that there is an essential need for drinking water, this problem becomes even more imperative. As this problem continues, it causes local and global water shortages, which will affect people's normal lives such as forced migration [117]. As such, it is appropriate to consider it as a national and even an international issue. Unconscious agricultural irrigation, pollution, and population growth are among the main causes of water scarcity. While 70 percent of the world's accessible fresh water is used in agriculture, 60 percent of the water used is wasted due to inefficiency and wrong farming methods [121].

Therefore, water scarcity is also defined as the greatest danger of the future. On the other hand, if this problem is managed efficiently, it promises a sustainable life.

In this study, this problem was addressed from a multidisciplinary perspective, not from a purely social sciences perspective. Recently, an increasing number of studies have been carried out focusing on the engineering approach that positively affects people's social life [116]. In addition to or alternatively to traditional water transfer and management, technological and algorithmic-based solutions are vital in conserving water and preventing future water shortages. Therefore, artificial intelligence, machine learning, and various heuristic-based methods have come to the fore. In this study, we focused on the heuristic-based approach, as these optimization techniques can play an important role in managing and delivering better solutions. Based on this, the problem was addressed from an appropriate side, and therefore optimizations of proposed and developed projects are important for planning, designing, operational, and implementation activities. There are also some studies in the literature on this subject, albeit a few [122,123].

The water resources management problem can be addressed from many different dimensions, such as the water distribution network problem [124], the river-basin planning [125], the ground surface and underground management problem [126], the agricultural land allocation [115], and the irrigation problem [11,127]. Regarding water resource management, in addition to climate change, the amount and quality of water resources, especially in arid geographies, and their impact on hydrological processes, including groundwater, raise concerns. Therefore, it is important to use water in a balanced way and to manage resources correctly. In this study, we focused on the reservoir operation to the efficient use of water resources to reduce water and energy scarcity. The aim of this problem is to maximize the energy production of the multi-reservoir system and is formulated as follows (Equation (15)):

$$Max \sum_{i=1}^{T}\sum_{j=1}^{P} S_{i,j}\Delta i \,;S_{i,j} = r_j q_{i,j} h_{i,j} \tag{15}$$

Important parameters are the water level, which represents the change in output, and the power output, which is related to turbine oscillation and hydraulic head. Based on this, the aim is to find optimum solutions for outflow, hydraulic head, and power of output. The fitness function is the total energy production (kWh) of all facilities over the entire operating period, and as mentioned, the goal is to maximize it. T is the number of operation intervals, and P is the number of hydro plants. $S_{i,j}$ is the power output of the jth plant in the ith operating interval and is in MW; Δi is the operating range and is on an hourly basis; r_j is the power coefficient for jth plant; and $q_{i,\mathrm{j}}$ is the emissive turbine, and $h_{i,j}$ is the hydraulic head in the ith operating range of the jth plant, and their outputs are m^3/s and m, respectively. Furthermore, this problem has five constraints (Equation (16)). These constraints are power output, water level, continuity equation, boundary, and outflow constraints, which are presented below, respectively:

$$\lfloor S_{j,i}\rfloor \le S_{j,i} \le \lceil S_{j,i}\rceil \,;\, \lfloor S_i\rfloor \le \sum_{j=1}^{P}\lceil S_{j,i}\rceil \le \lceil S_i\rceil \tag{16}$$

$$\begin{gathered}\lfloor Z_{j,i}\rfloor \le Z_{j,i} \le \lceil Z_{j,i}\rceil \\ V_{j,\,i+1} = V_{j,\,i} + \alpha\left(Y_{j,i} - W_{j,i}\right);\; V_{j+1,\,i+1} = V_{j+1,\,i} + \alpha\left(Y_{j+1,i} - W_{j+1,i}\right);\; Y_{j+1,\,i} \\ = Y\left(W_{j+1,i} + W_{j,i-\gamma_i}\right) \\ Z_{j,1} = Z_{j,b}\,,\; Z_{j,T+1} = Z_{j,e}\lfloor W_{i,j}\rfloor \le W_{i,j} \le \lceil W_{i,j}\rceil\end{gathered}$$

In this study, a famous real structure representing this problem was considered as a case study, and the results were evaluated. This structure is represented in two dimensions in Figure 19. Here, four hydropower reservoirs were considered. Details regarding this multi-reservoir were presented in [128].

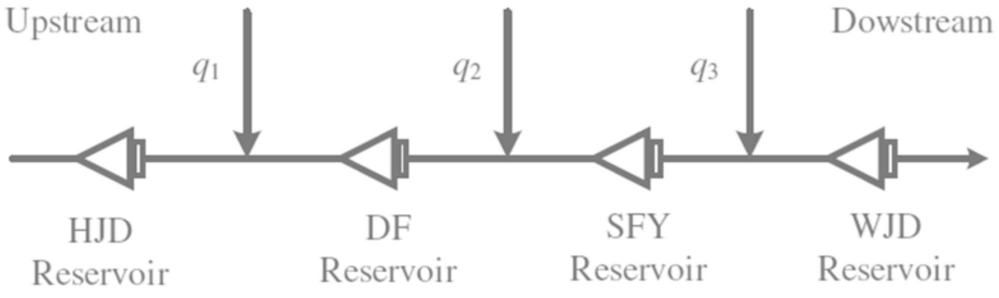


Figure 19. Sketch of cascade reservoir in WJ River basin [128].

Considering the assumptions in the [128] study, the results are presented in Table 18. According to the results, our proposed Tent-based algorithm demonstrates the best performance. Based on this analysis, a more efficient system design will be achieved in the system of multiple reservoirs if the relevant algorithm is used, and, therefore, the water resource will be able to be managed in the best way. As a result, efficient water consumption will be achieved by finding efficient solutions for optimization problems such as the design of improved systems and well placement. Therefore, local and global risks will be reduced, a better future will be left for future generations, and problems such as forced migration due to thirst will be eliminated.

Table 18. Experimental results of the multi-reservoirs design problem (period = 10, pop = 100, and iter = 500).

Algor-ithms	Optimum Cost for Outflow				Optimum Cost for Hydraulic Head				Optimum Cost for Power of Output				Optimal COST	Overall_ Rank
	HJD	DF	SFY	WJD	HJD	DF	SFY	WJD	HJD	DF	SFY	WJD		
Map1	116.9	367.9	469.1	408.8	136.4	128.3	69.0	132.9	113.9	447.9	377.1	479.1	1887.9	2
Map2	143.5	339.4	328.7	476.9	138.1	128.7	69.0	132.8	268.7	347.8	227.7	400.9	1849.5	6
Map3	189.8	327.9	313.9	456.5	138.5	128.8	69.0	132.8	218.5	387.3	287.9	398.5	1845.1	7
Map4	174.8	251.5	328.3	612.8	131.8	128.0	69.0	129.8	112.9	299.4	193.8	796.8	1329.8	10
Map5	127.9	353.1	357.2	506.8	138.2	129.1	69.0	133.0	186.4	365.2	259.9	431.2	1852.8	5
Map6	327.3	321.4	316.5	301.7	129.8	128.9	69.0	132.2	299.3	337.2	194.6	399.8	1279.8	12
Map7	355.2	328.3	337.9	322.9	129.1	128.6	69.0	132.5	307.9	345.6	213.4	386.8	1283.3	11
Map8	119.5	377.4	363.8	526.5	138.1	128.1	69.0	132.1	218.5	387.3	287.9	408.7	1874.3	3
Map9	223.8	398.7	427.9	501.9	155.5	128.9	69.0	133.0	266.2	405.7	258.1	494.3	1509.3	8
Map10	**97.7**	**386.3**	**496.5**	**888.6**	**149.8**	**127.6**	**69.0**	**132.6**	**83.5**	**652.6**	**407.7**	**939.8**	**2087.3**	**1**
Map11	317.2	328.8	326.1	341.3	129.6	128.7	69.0	132.8	317.4	355.7	194.8	376.6	1276.9	13
Map12	214.8	338.3	339.8	381.1	151.4	128.8	69.0	133.2	298.6	394.9	237.3	405.5	1269.7	14
SE-SCSO	247.9	281.8	290.8	324.6	139.4	129.1	69.0	132.9	293.8	303.9	170.6	352.4	1120.6	15
CGWO	218.2	289.4	288.5	296.1	142.4	129.1	69.0	133.0	290.2	290.1	168.4	389.7	1117.4	16
CMPA	100.3	276.3	501.7	786.3	147.9	128.8	69.0	132.5	299.6	603.8	313.9	834.9	1854.5	4
CWOA	102.5	263.9	324.7	807.2	120.1	127.2	70.6	125.5	72.3	277.8	168.5	801.4	1329.8	9

The best algorithm is shown in bold and highlight.

8. Conclusions

This study proposed a novel hybrid Chaotic Sand Cat Swarm Optimization (CSCSO) algorithm and investigated its performance in relation to various real-world optimization problems. The concept of chaos is a suitable solution to resolve problems such as possible early convergence, low search consistency, local optimum problem, inefficient search, and low population diversity of commodity. The SCSO has the possibility of not finding an effective global optimum. In addition, it is likely that agents will be blind due to the random operating mechanism and therefore offer a limited rate of exploitation in the search space. Therefore, chaotic maps are integrated into the fundamental SCSO algorithm that results in a hybrid algorithm. In this way, CSCSO uses chaotic sequences mapped from chaotic maps to generate design variables instead of random sequences. Moreover, it will also benefit from the randomness characteristics of SCSO. In order to manage this hybrid structure, a criterion was determined in the algorithm, and a random variable was defined in this context. This variable can be greater than/equal to or less than 0.5. Therefore, the CSCSO algorithm introduces two alternative solutions for position updating in the exploration and exploitation phases: one of them is the normal position update mechanism, and the other is based on the chaotic model. Twelve different chaotic maps are used in the new algorithm that uses the concept of chaos. The CSCSO algorithm based on maps has the same complexity though slightly more runtime compared to the standard version and SE-SCSO. In this study, the focus of the performance evaluation was the behavior of the proposed hybrid algorithm in real-world problems. In this regard, current and well-known problems in numerical and engineering optimization were discussed. In addition to these, eight unimodal, eight multimodal, and eight comparative test functions are also used. For this, functions and problems were selected from CEC2015, 2016, 2019, and 2020. In general, the results exhibit the superiority of the chaotic SCSO algorithm over standard and current variants of SCSO and other tested chaotic-based metaheuristics (CGWO, CMPA, and CWOA) on various complex and constrained multidisciplinary problems in terms of convergence behavior and global optimum solution. In a computational analysis, the CSCSO algorithm with a set of maps, which is proposed to solve multi-constrained and nonlinear real-world optimization problems, was successful, like any other complex problem considered. In the analysis of the results, the proposed algorithm found the best solutions with a Tent-based map in unimodal functions. This behavior and consequence are similarly valid in multimodal competitive functions and numerical real-world problems. Moreover, seven different constrained multidisciplinary problems were also discussed in the analysis section. In these applications, the proposed algorithm was used to solve certain problems. According to the results obtained, Tent-based CSCSO, Chebyshev-based CMPA, Tent-based CSCSO, Sinusoidal-based CSCSO, and Tent-based CSCSO performed better than other maps and algorithms in these problems, respectively. Moreover, in the p-median and water resource management problems, Tent-based CSCSO found the best result. Briefly, it achieved the best solution for a total of six problems based on different maps in seven constrained optimization problems. It is worth emphasizing that the proposed algorithm can work well with different maps for various types of problems. According to the results, the proposed algorithm found the best answer at a total rate of 87.5% in the unimodal test functions considered. This ratio resulted in 62.5% superiority in multimodal benchmark functions, 75% in competitive modern functions, and 75% in CEC2020 real-world problems. In conclusion, these extensive experiments indicate that the CSCSO algorithm excels in providing acceptable results for a wide variety of problems.

As for future works, researchers can focus on multi-objective and multidimensional design optimization problems. In addition, binary classification and multiclassification may be other focal issues. On the other hand, the proposed algorithm can be used on different real applications (e.g., image processing, feature selection, smart traffic management, sustainable agriculture, logistics, etc.). Blended with the chaos feature, this method can also be used in mobile ad hoc systems and IoT architectures for problems such as extending coverage and optimizing node localization. Moreover, it can be tested by adapting to other competition functions, such as CEC2022. Furthermore, a broader perspective can be gained under the concept of hyper-heuristic. In addition, interested researchers can further analyze its performance by improving this method.

Author Contributions: Conceptualization, F.K., S.N. and F.A.A.; methodology, F.K.; software, F.K. and S.N.; validation, F.K. and S.N.; formal analysis, F.K. and S.N.; investigation, F.K., F.A.A. and M.A.F.; resources, F.K., F.A.A. and M.A.F.; writing-original draft preparation, F.K., F.A.A., M.A.F. and S.N.; writing-review and editing, F.K., F.A.A., M.A.F. and S.N.; visualization, F.K. and S.N.; supervision, F.K. and F.A.A.; project administration, F.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data Available on Request.

Acknowledgments: We thank Mahsa Torkamanian-Afshar for her effort and help in this study.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Appendix A

Table A1. Unimodal benchmark functions.

Formula	Dim	Range	Global Minimum
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	$[-10, 10]$	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{i=j-1}^{i} x_j \right)^2$	30	$[-100, 100]$	0
$f_4(x) = max_i\{\lvert x_i, 1 \le i \le n\}$	30	$[-100, 100]$	0
$f_5(x) = \sum_{i=1}^{n-1} [100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	$[-100, 100]$	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	30	$[-1.28, 1.28]$	0
$f_8(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	$[-100, 100]$	0

Table A2. Multimodal benchmark functions.

Formula	Dim	Range	Global Minimum
$Mf_1(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	$[-500, 500]$	$-418.9829 \times$ dim
$MF_2(x) = (\sum_{i=1}^{n} \lvert x_i\rvert) \exp\left(-\sum_{i=1}^{n} Sin\left(x_i{}^2\right)\right)$	50	$[-100, 100]$	0
$Mf_3(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i-1)^2\ [1+10sin^2(\pi y_{1+1})] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10{,}100{,}4)$; $y_i = 1 + \frac{x_i+1}{4}$; $u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a < x_i < 1 \\ k(-x_i-a)^m & x_i < -a \end{cases}$	30	$[-50, 50]$	0
$Mf_4(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i-1)^2\,[1+sin^2(3\pi x_i+1)] + (x_n-1)^2[1+sin^2(2\pi x_{ni})]\right\} + \sum_{i=1}^{n} u(x_i, 5{,}100{,}4)$	30	$[-50, 50]$	0
$MF_5(x) = \sum_{i=1}^{n} \left(\mathrm{x}^2 - i\right)^2$	50	$[-5.12, 5.12]$	0
$MF_6(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{X_i}{\sqrt{i}}\right) + 1$	50	$[-10, 10]$	0.9
$MF_7(x) = \sum_{i=1}^{n} \epsilon_i \lvert x_i\rvert^i$	50	$[-500, 500]$	0
$MF_8(x) = \left(\sum_{i=1}^{n} sin^2 x_i - e^{-\sum_{i=1}^{n} x_i^2}\right) e^{-\sum_{i=1}^{n} sin^2\left(\sqrt{\lvert x_i\rvert}\right)}$	50	$[-10, 10]$	-1

Table A3. Modern benchmark competitive test functions from CEC2019 (CEC−C06).

Function	Benchmark Function	Dim	Range	f_{min}
CEC01	Storn's Chebyshev Polynomial Fitting Problem	9	$[-8192, 8192]$	1
CEC02	Inverse Hilbert Matrix Problem	16	$[-16{,}384, 16{,}384]$	1
CEC03	Rastrigin's Function	10	$[-100, 100]$	1
CEC04	Grıenwank's Function	10	$[-100, 100]$	1
CEC05	Weiersrass Function	10	$[-100, 100]$	1
CEC06	Modified Schwefel's Function	10	$[-100, 100]$	1
CEC07	Expanded Schaffer's F6 Function	10	$[-100, 100]$	1
CEC08	Ackley Function	10	$[-100, 100]$	1

Table A4. Numerical real-world functions from CEC2020.

Function	Benchmark Function	Dim	Range	f_{min}
R_1	Shifted and Rotated Griewank's Function without Bounds	50	$[-600, 600]$	700
R_2	Shifted and Rotated Rastrigin's Function	50	$[-5, 5]$	500
R_3	Shifted and Rotated Weierstrass Function	50	$[-0.5, 0.5]$	300
R_4	Shifted and Rotated Expanded Scaffer's	50	$[-100, 100]$	600
R_5	Shifted and Rotated Schwefel's function	10	$[-100, 100]$	1100
R_6	Shifted and Rotated Lunacek bi−Rastrigin function	10	$[-100, 100]$	700
R_7	Hybrid function 2 (N = 4)	10	$[-100, 100]$	1600
R_8	Composition function 2 (N = 4)	10	$[-100, 100]$	2400

Table A5. Mathematical equations of constrained engineering problems.

The mathematical equations of pressure vessel design problem

$\vec{x} = [x_1\ x_2\ x_3\ x_4] = [T_s\ T_h\ R\ L]$,

$Min_x f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$,

$g_1\left(\vec{x}\right) = -x_1 + 0.0193x_3 \leq 0,\ g_2(\vec{x}) = -x_2 + 0.00954x_3 \leq 0,\ g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0,\ g_4(\vec{x}) = x_4 - 240 \leq 0,$

$0 \leq x_1,\ x_2 \leq 99, 10\ \leq x_{3,}\ ,\ x_4 \leq 200$

The mathematical equations of gear train design problem

$GearRatio = \frac{n_bn_d}{n_an_f}$

$Min_x f(x) = \left(\frac{1}{6.931} - \frac{x_3x_2}{x_1x_4}\right)^2\ \ 12 \leq x_i \leq 60;\ i = 1,\ 2,\ 3,\ 4$

The mathematical equations of tension/compression spring design problem

$\vec{x} = [x_1\ x_2\ x_3] = [d\ D\ N]$,

$Min_x f(\vec{x}) = (x_3 + 2)x_2x_1^2$,

$g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0,\ g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566\ \left(x_2x_1^3 - x_1^4\right)} + \frac{1}{5108x_1^2} - 1 \leq 0,\ \ g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0,\ g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$

$0.05\ \leq x_1 \leq 2.00, 0.25\ \leq x_2 \leq 1.30,\ 2.00\ \leq x_3 \leq 15.0$

The mathematical equations of speed reducer design problem

$Min\ f(b,\ m,\ z, l_1, l_2,\ d_1,\ d_2\ ) =$

$0.7854x_1{x_2}^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)$

$G_1 = \frac{27}{x_1x_2^2x_3} - 1 \leq 0;\ \ G_2 = \frac{397.5}{x_1x_2^2{x_3}^2} - 1 \leq 0;\ \ G_3 = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \leq 0;\ \ G_4 = \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \leq 0;\ G_5 = \frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9\times10^6}}{110x_6^3} - 1 \leq 0;$

$G_6 = \frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.5\times10^6}}{85x_7^3} - 1 \leq 0;\ G_7 = \frac{x_2x_3}{40} - 1 \leq 0;\ G_8 = \frac{5x_2}{x_1} - 1 \leq 0;\ G_9 = \frac{x_1}{12x_2} - 1 \leq 0;\ \ G_{10} = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0;$

$G_{11} = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0;$

$2.6 \leq x_1 \leq 3.6;\ 0.7 \leq x_2 \leq 0.8;\ 17 \leq x_3 \leq 28;\ 7.3 \leq x_4;\ x_5 \leq 8.3;\ 2.9 \leq x_6 \leq 3.9; 5 \leq x_7 \leq 5.5$

The mathematical equations of three-bar truss design problem

$Min\ f(A1,\ A2) = l \times \left(2\sqrt{2}x_1 + x_2\right)$

$G_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}{x_1}^2 + 2x_1x_2}P - \sigma \leq 0;\ \ G_2 = \frac{x_2}{\sqrt{2}{x_1}^2 + 2x_1x_2}P - \sigma \leq 0;\ \ \ G_3 = \frac{1}{\sqrt{2}x_2 + x_1}P - \sigma \leq 0;\ l = 100cm;\ \ P = \frac{2kN}{cm^2};\ \ \sigma = \frac{2kN}{cm^2}$

$0 \leq x_1,\ x_2 \leq 1$

References

1. Abdel-Basset, M.; Mohamed, R.; Jameel, M.; Abouhawwash, M. Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. *Knowl. Based Syst.* **2023**, *262*, 136. [CrossRef]
2. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The Arithmetic Optimization Algorithm, Computer Methods. *Appl. Mech. Eng.* **2021**, *376*, 113609. [CrossRef]
3. Kiani, F.; Anka, F.A.; Erenel, F. PSCSO: Enhanced Sand Cat Swarm Optimization Inspired by the Political System to Solve Complex Problems. *Adv. Eng. Softw.* **2023**, *178*, 103423. [CrossRef]
4. Kalinin, K.P.; Berloff, N.G. Computational complexity continuum within using formulation of NP problems. *Nat. Commun. Phys.* **2022**, *5*, 20.
5. Kumar, S.; Yildiz, B.S.; Mehta, P.; Panagant, N.; Sait, S.M.; Mirjalili, S.; Yildiz, A.R. Chaotic marine predators algorithm for global optimization of real-world engineering problems. *Knowl. Based Syst.* **2023**, *261*, 110192. [CrossRef]
6. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
7. Kiani, F.; Seyyedabbasi, A.; Aliyev, R.; Gulle, M.U.; Basyildiz, H.; Shah, M.A. Adapted-RRT: Novel hybrid method to solve three-dimensional path planning problem using sampling and metaheuristic-based algorithms. *Neural Comput. Appl.* **2021**, *33*, 15569–15599. [CrossRef]
8. Nematzadeh, S.; Torkamanian-Afshar, M.; Seyyedabbasi, A.; Kiani, F. Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: An efficient metaheuristic-based method for environment-aware node deployment. *Neural Comput. Appl.* **2023**, *33*, 15569–15599. [CrossRef]
9. Salgotra, R.; Singh, S.; Singh, U.; Mirjalili, S.; Gandomi, A. Marine predator inspired naked molerat algorithm for global optimization. *Expert Syst. Appl.* **2023**, *212*, 118822. [CrossRef]
10. Al-Gharaibeh, R.; Ali, M. Real-parameter constrained optimization using enhanced quality-based cultural algorithm with novel influence and selection schemes. *Inf. Sci.* **2021**, *576*, 242–273. [CrossRef]
11. Kiani, F.; Seyyedabbasi, A.; Nematzadeh, S.; Candan, F.; Çevik, T.; Anka, F.A.; Randazzo, G.; Lanza, S.; Muzirafuti, A. Adaptive Metaheuristic-based Methods for Autonomous Robot Path Planning: Sustainable Agricultural Applications. *Appl. Sci.* **2022**, *12*, 943. [CrossRef]
12. Kochkarov, R. Research of NP-Complete Problems in the Class of Prefractal Graphs. *Mathematics* **2021**, *9*, 2764. [CrossRef]
13. Arasteh, B.; Fatolahzadeh, A.; Kiani, F. Savalan: Multi objective and homogeneous method for software modules clustering. *J. Softw. Evol. Process* **2022**, *34*, e2408. [CrossRef]
14. Seyyedabbasi, A.; Kiani, F.; Allahviranloo, T.; Fernandez-Gamiz, U.; Noeiaghdam, S. Optimal data transmission and pathfinding for WSN and decentralized IoT systems using I-GWO and Ex-GWO algorithms. *Alex. Eng. J.* **2023**, *63*, 339–357. [CrossRef]
15. Ghasemi, M.; Kadkhoda Mohammadi, S.; Zare, M.; Mirjalili, S.; Gil, M.; Hemmati, R. A new firefly algorithm with improved global exploration and convergence with application to engineering optimization. *Decis. Anal. J.* **2022**, *5*, 100125. [CrossRef]
16. Nematzadeh, S.; Kiani, F.; Torkamanian-Afshar, M.; Aydin, N. Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases. *Comput. Biol. Chem.* **2022**, *97*, 107619. [CrossRef] [PubMed]
17. Seyyedabbasi, A.; Kiani, F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Eng. Comput.* **2022**, *2021*, 1–25. [CrossRef]
18. Osuna-Enciso, V.; Cuevas, E.; Castañeda, B.M. A diversity metric for population-based metaheuristic algorithms. *Inf. Sci.* **2022**, *586*, 192–208. [CrossRef]
19. Agushaka, J.O.; Ezugwu, A.E. Initialisation Approaches for Population-Based Metaheuristic Algorithms: A Comprehensive Review. *Appl. Sci.* **2022**, *12*, 896. [CrossRef]
20. Gezici, H.; Livatyali, H. Chaotic Harris hawks optimization algorithm. *J. Comput. Des. Eng.* **2022**, *9*, 216–245. [CrossRef]
21. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
22. He, Y.; Zhang, F.; Mirjalili, S.; Zhang, T. Novel binary differential evolution algorithm based on Taper-shaped transfer functions for binary optimization problems. *Swarm Evol. Comput.* **2022**, *69*, 101022. [CrossRef]
23. Bao, C.; Gao, D.; Gu, W.; Xu, L.; Goodman, L. A new adaptive decomposition-based evolutionary algorithm for multi- and many-objective optimization. *Expert Syst. Appl.* **2023**, *213*, 119080. [CrossRef]
24. Hayyolalam, V.; Pourhaji Kazem, A.A. Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103249. [CrossRef]
25. El-Kenawy, E.M.; Abdelhamid AIbrahim, A.; Mirjalili, S. Al-biruni earth radius (ber) metaheuristic search optimization algorithm. *Comput. Syst. Sci. Eng.* **2023**, *45*, 1917–1934. [CrossRef]
26. Sang-To, T.; Hoang-Le, M.; Wahab, M.A. An efficient Planet Optimization Algorithm for solving engineering problems. *Sci. Rep.* **2022**, *12*, 8362. [CrossRef] [PubMed]
27. Ahmadianfar, I.; Heidari, A.; Gandomi, A.H. RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta Method. *Expert Syst. Appl.* **2021**, *181*, 115079. [CrossRef]
28. Ahmadi, S.A. Human behavior-based optimization: A novel metaheuristic approach to solve complex optimization problems. *Neural Comput. Appl.* **2017**, *28*, 233–244. [CrossRef]
29. Dehghani, M.; Trojovská, E.; Trojovský, P.A. new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. *Sci. Rep.* **2022**, *12*, 9924. [CrossRef]

30. Askari, Q.; Younas, I.; Saeed, M. Political Optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowl.-Based Syst.* **2020**, *195*, 1–25. [CrossRef]
31. Sindhiya, R.; Perumal, B.; Pallikonda, M. A hybrid deep learning based brain tumor classification and segmentation by stationary wavelet packet transform and adaptive kernel fuzzy c means clustering. *Adv. Eng. Softw.* **2022**, *170*, 103146. [CrossRef]
32. Seyyedabbasi, A.; Kiani, F. I-GWO and Ex-GWO: Improved algorithms of the Grey Wolf Optimizer to solve global optimization problems. *Eng. Comput.* **2021**, *37*, 509–532. [CrossRef]
33. Zitouni, F.; Harous, S.; Belkeram, A.; Hammou, L.E.B. The Archerfish Hunting Optimizer: A Novel Metaheuristic Algorithm for Global Optimization. *Arab J. Sci. Eng.* **2022**, *47*, 2513–2553. [CrossRef]
34. Saremi, S.; Mirjalili, S.; Lewis, A. Biogeography-based optimization with chaos. *Neural Comput. Appl.* **2014**, *25*, 1077–1097. [CrossRef]
35. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
36. Kohli, M.; Arora, S. Chaotic grey wolf optimization algorithm for constrained optimization problems. *J. Comput. Des. Eng.* **2018**, *5*, 458–472. [CrossRef]
37. Amirteimoori, A.; Mahdavi, I.; Solimanpur, M.; Ali, S.S.; Tirkolaee, E.B. A parallel hybrid PSO-GA algorithm for the flexible flow-shop scheduling with transportation. *Comput. Ind. Eng.* **2022**, *173*, 1–16. [CrossRef]
38. Naik, A. Chaotic Social Group Optimization for Structural Engineering Design Problems. *J. Bionic Eng.* **2023**, *2023*, 1–26. [CrossRef]
39. Kiani, F.; Seyyedabbasi, A.; Mahouti, P. Optimal characterization of a microwave transistor using grey wolf algorithms. *Analog. Integr. Circ. Sig. Process* **2021**, *109*, 599–609. [CrossRef]
40. Sharma, V.; Tripathi, A.K. A systematic review of meta-heuristic algorithms in IoT based application. *Array* **2022**, *14*, 100164. [CrossRef]
41. Anka, F.; Seyyedabbasi, A. Metaheuristic Algorithms in IoT: Optimized Edge Node Localization. In *Engineering Applications of Modern Metaheuristics. Studies in Computational Intelligence*; Akan, T., Anter, A.M., Etaner-Uyar, A.Ş., Oliva, D., Eds.; Springer: Cham, Switzerland, 2023; Volume 1069. [CrossRef]
42. Kiani, F.; Seyyedabbasi, A.; Nematzadeh, S. Improving the performance of hierarchical wireless sensor networks using the metaheuristic algorithms: Efficient cluster head selection. *Sens. Rev.* **2021**, *41*, 368–381. [CrossRef]
43. Babaeinesami, A.; Tohidi, H.; Ghasemi, P.; Goodarzian, F.; Tirkolaee, E.B. A closed-loop supply chain configuration considering environmental impacts: A self-adaptive NSGA-II algorithm. *Appl. Intell.* **2022**, *52*, 13478–13496. [CrossRef]
44. Wang, G.G.; Deb, S.; Cui, Z. Monarch butterfly optimization. *Neural Comput. Appl.* **2019**, *31*, 1995–2014. [CrossRef]
45. MiarNaeimi, F.; Azizyan, G.; Rashki, M. Multi-level cross entropy optimizer (MCEO): An evolutionary optimization algorithm for engineering problems. *Eng. Comput.* **2018**, *34*, 719–739. [CrossRef]
46. Ghaemi, M.; Feizi-Derakhshi, M. Forest Optimization Algorithm. *Expert Syst. Appl.* **2014**, *41*, 6676–6687. [CrossRef]
47. Guo, C. A survey of bacterial foraging optimization. *Neurocomputing* **2021**, *452*, 728–746. [CrossRef]
48. Kaveh, A.; Bakhshpoori, T. Big Bang-Big Crunch Algorithm. In *Metaheuristics: Outlines, MATLAB Codes and Examples*; Springer: Cham, Switzerland, 2019. [CrossRef]
49. Pereira, J.L.J.; Francisco, M.B.; Diniz, C.A.; Oliver, G.A.; Cunha, S.S., Jr.; Gomes, G.F. Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. *Expert Syst. Appl.* **2021**, *170*, 114522. [CrossRef]
50. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **2019**, *101*, 646–667. [CrossRef]
51. Mohamed, A.; Emam, A.; Zoheir, B. SAM-HIT: A Simulated Annealing Multispectral to Hyperspectral Imagery Data Transformation. *Remote Sens.* **2023**, *15*, 1154. [CrossRef]
52. Mohamed, A.W.; Hadi, A.A.; Mohamed, A.K. Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. *Int. J. Mach. Learn. Cyber.* **2020**, *11*, 1501–1529. [CrossRef]
53. Tu, J.; Chen, H.; Wang, M.; Gandomi, A.H. The Colony Predation Algorithm. *J. Bionic. Eng.* **2021**, *18*, 674–710. [CrossRef]
54. Zhang YJin, Z. Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. *Expert Syst. Appl.* **2020**, *148*, 113246. [CrossRef]
55. Yang, Y. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* **2021**, *177*, 114864. [CrossRef]
56. Li, S.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]
57. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; pp. 1942–1948.
58. Gharehchopogh, F.S.; Namazi, M.; Ebrahimi, L.; Abdollahzadeh, B. Advances in Sparrow Search Algorithm: A Comprehensive Survey. *Arch. Comput. Methods Eng.* **2023**, *30*, 427–455. [CrossRef]
59. Khishe, M.; Mosavi, M.R. Chimp optimization algorithm. *Expert Syst. Appl.* **2020**, *149*, 113338. [CrossRef]
60. Mohammed, H.; Rashid, T. A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design. *Neural Comput. Appl.* **2020**, *32*, 14701–14718. [CrossRef]

61. Patel, S.K.; Pandey, A.K.; Roshan, R.; Singh, U.K. Application of PSO and GSA Hybrid Optimization Method for 1-D Inversion of Magnetotelluric Data. In Proceedings of the International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Odisha, India, 3–5 October 2016; pp. 1908–1911.
62. Wang, W.; Liu, F.; Wang, W.; Cheng, M. The Chaotic Time Series Prediction Method Based on Sparrow Search Algorithm Optimization. In Proceeding of the 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Shenyang, China, 17–19 December 2021.
63. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
64. Mirjalili, S.; Gandomi, A.H. Chaotic gravitational constants for the gravitational search algorithm. *Appl. Soft Comput.* **2017**, *53*, 407–419. [CrossRef]
65. Kaur, G.; Arora, S. (2018). Chaotic whale optimization algorithm. *J. Comput. Des. Eng.* **2018**, *5*, 275–284.
66. Yang, D.X.; Li, G.; Cheng, G.D. On the efficiency of chaos optimization algorithms for global optimization. *Chaos Solitons Fractals* **2007**, *34*, 1366–1375. [CrossRef]
67. Secui, D.C. A modified Symbiotic Organisms Search algorithm for large scale economic dispatch problem with valve-point effects. *Energy* **2016**, *113*, 366–384. [CrossRef]
68. Rezaee Jordehi, A. A chaotic-based big bang–big crunch algorithm for solving global optimisation problems. *Neural Comput. Appl.* **2014**, *25*, 1329–1335. [CrossRef]
69. Erol, O.K.; Eksin, I. A new optimization method: Big bang-big crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [CrossRef]
70. Wang, G.G.; Deb, S.; Gandomi, A.H.; Zhang, Z.; Alavi, A.H. Chaotic cuckoo search. *Soft Comput.* **2016**, *20*, 3349–3362. [CrossRef]
71. Wang GGGandomi, A.H.; Alavi, A.H. A chaotic particle-swarm krill herd algorithm for global numerical optimization. *Kybernetes* **2013**, *42*, 962–978. [CrossRef]
72. Saremi, S.; Mirjalili, S.M.; Mirjalili, S. Chaotic krill herd optimization algorithm. *Procedia Technol.* **2014**, *12*, 180–185. [CrossRef]
73. Cheng, C.T.; Wang, W.C.; Xu, D.M.; Chau, K.W. Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos. *Water Resour. Manag.* **2008**, *22*, 895–909. [CrossRef]
74. Qiao, W.; Yang, Z. Modified dolphin swarm algorithm based on chaotic maps for solving high-dimensional function optimization problems. *IEEE Access* **2019**, *7*, 110472–110486. [CrossRef]
75. Wu, T.Q.; Yao, M.; Yang, J.H. Dolphin swarm algorithm. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 717–729. [CrossRef]
76. Tian, Y.; Jiang, P. Optimization of Tool Motion Trajectories for Pocket Milling Using a Chaos and Colony Algorithm. In Proceeding of the 10th IEEE International Conference on Computer-Aided Design and Computer Graphics, Beijing, China, 15–18 October 2007; pp. 389–394.
77. Karaboga, D.; Gorkemli, B.; Ozturk, C.; Karaboga, N. A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* **2014**, *42*, 21–57. [CrossRef]
78. Yang, D.; Liu, Z.; Zhou, J. Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 1229–1246. [CrossRef]
79. Ait-Saadi, A.; Meraihi, Y.; Soukane, A.; Ramdane-Cherif, A.; Gabis, A.B. A novel hybrid Chaotic Aquila Optimization algorithm with Simulated Annealing for Unmanned Aerial Vehicles path planning. *Comput. Electr. Eng.* **2022**, *104*, 108461. [CrossRef]
80. Yıldız, B.S.; Mehta, P.; Panagant, N.; Mirjalili, S.; Yildiz, A.R. A novel chaotic Runge Kutta optimization algorithm for solving constrained engineering problems. *J. Comput. Des. Eng.* **2022**, *9*, 2452–2465. [CrossRef]
81. He, Y.Y.; Zhou, J.Z.; Xiang, X.Q.; Chen, H.; Qin, H. Comparison of different chaotic maps in particle swarm optimization algorithm for long-term cascaded hydroelectric system scheduling. *Chaos Solitons Fractals* **2009**, *42*, 3169–3176. [CrossRef]
82. Wang, P.; Zhang, Y.; Yang, H. Research on economic optimization of microgrid cluster based on chaos sparrow search algorithm. *Comput. Intell. Neurosci.* **2021**, *2021*, 5556780. [CrossRef]
83. Xiong, H.; Wu, Z.; Fan, H.; Li, G.; Jiang, G. Quantum rotation gate in quantum-inspired evolutionary algorithm: A review, analysis and comparison study. *Swarm Evol. Comput.* **2018**, *42*, 43–57. [CrossRef]
84. Liu, Q.; Zhang, Y.; Li, M.; Zhang, Z.; Cao, N.; Shang, J. MultiUAV path planning based on fusion of sparrow search algorithm and improved bioinspired neural network. *IEEE Access* **2021**, *9*, 124670–124681. [CrossRef]
85. Elaziz, M.; Yousri, D.; Mirjalili, S. A hybrid Harris hawks-moth-flame optimization algorithm including fractional-order chaos maps and evolutionary population dynamics. *Adv. Eng. Softw.* **2021**, *154*, 102973. [CrossRef]
86. Aydemir, S.B. A novel arithmetic optimization algorithm based on chaotic maps for global optimization. *Evol. Intell.* **2022**, *2022*, 1–16. [CrossRef]
87. Liang, J.J.; Qu, B.Y.; Suganthan, P.N.; Chen, Q. *Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-based Real-Parameter Single Objective Optimization*; Technical Report 201411A; Computational Intelligence Laboratory, Zhengzhou University: Zhengzhou, China; Nanyang Technological University: Singapore, 2014; pp. 625–640.
88. Yang, M.; Guan, J.; Li, C. Differential Evolution with Auto-Enhanced Population Diversity: The Experiments on the CEC'2016 Competition. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 4785–4789.
89. Price, K.V.; Awad, N.H.; Ali, M.Z.; Suganthan, P.N. *The 100-Digit Challenge: Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization*; Nanyang Technological University: Singapore, 2018.

90. Song, M.; Jia, H.; Abualigah, L.; Liu, Q.; Lin, Z.; Wu, D.; Altalhi, M. Modified Harris Hawks Optimization Algorithm with Exploration Factor and Random Walk Strategy. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–23. [CrossRef]
91. Kumar, A.; Wu, G.; Ali, M.Z.; Mallipeddi, R.; Suganthan, P.N.; Das, S. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm Evol. Comput.* **2019**, *2019*, 1–15. [CrossRef]
92. Nekoo, S.R.; Acosta, J.Á.; Ollero, A. A search algorithm for constrained engineering optimization and tuning the gains of controllers. *Expert Syst. Appl.* **2022**, *2022*, 117866. [CrossRef]
93. Duran-Mateluna, C.; Ales, Z.; Elloumi, S. An efficient benders decomposition for the p-median problem. *Eur. J. Oper. Res.* **2022**, *308*, 84–96. [CrossRef]
94. Li, Y.; Wang, G. Sand Cat Swarm Optimization Based on Stochastic Variation with Elite Collaboration. *IEEE Access* **2022**, *10*, 89989–90003. [CrossRef]
95. Jovanovic, D.; Marjanovic, M.; Antonijevic, M.; Zivkovic, M.; Budimirovic, N.; Bacanin, N. Feature Selection by Improved Sand Cat Swarm Optimizer for Intrusion Detection. In Proceeding of the International Conference on Artificial Intelligence in Everything (AIE), Lefkosa, Cyprus, 2–4 August 2022; pp. 685–690.
96. Wu, D.; Rao, H.; Wen, C.; Jia, H.; Liu, Q.; Abualigah, L. Modified Sand Cat Swarm Optimization Algorithm for Solving Constrained Engineering Optimization Problems. *Mathematics* **2022**, *10*, 4350. [CrossRef]
97. Rahman, C.M.; Rashid, T.A. A new evolutionary algorithm: Learner performance-based behavior algorithm. *Egypt. Inform. J.* **2021**, *22*, 213–223. [CrossRef]
98. Seyyedabbasi, A.; Aliyev, R.; Kiani, F.; Gulle, M.U.; Basyildiz, H.; Shah, M.A. Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems. *Knowl.-Based Syst.* **2021**, *223*, 107044. [CrossRef]
99. Van den Bergh, F.; Engelbrecht, A.P. A study of particle swarm optimization particle trajectories. *Inf. Sci.* **2006**, *176*, 937–971. [CrossRef]
100. Olorunda, O.; Engelbrecht, A.P. Measuring Exploration/Exploitation in Particle Swarms Using Swarm Diversity. In Proceeding of the IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 1128–1134.
101. Qtaish, A.; Albashish, D.; Braik, M.; Alshammari, M.T.; Alreshidi, A.; Alreshidi, E.J. Memory-based Sand Cat Swarm Optimization for Feature Selection in Medical Diagnosis. *Electronics* **2023**, *12*, 2042. [CrossRef]
102. Chattopadhyay, S. *Pressure Vessels: Design and Practice*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2004. [CrossRef]
103. Yokota, T.; Taguchi, T.; Gen, M. A solution method for optimal weight design problem of the gear using genetic algorithms. *Comput. Ind. Eng.* **1998**, *35*, 523–526. [CrossRef]
104. Coello, C.A.C. Use of a self-adaptive penalty approach for engineering optimization problems. *Comput. Ind.* **2000**, *41*, 113–127. [CrossRef]
105. Bayzidi, H.; Talatahari, S.; Saraee, M.; Lamarche, C.P. Social Network Search for Solving Engineering Optimization Problems. *Comput. Intell. Neurosci.* **2021**, *2021*, 1–32. [CrossRef] [PubMed]
106. Nowcki, H. Optimization in pre-contract ship design. In *Computer Applications in the Automation of Shipyard Operation and Ship Design*; Fujita, Y., Lind, K., Williams, T.J., Eds.; Elsevier: Amsterdam, The Netherlands, 1974; Volume 2, pp. 327–338.
107. Parsopoulos, K.E.; Vrahatis, M.N. Unified particle swarm optimization for solving constrained engineering optimization problems. *Lect. Notes Comput. Sci.* **2005**, *3612*, 582.
108. Kariv, O.; Hakimi, S.L. An algorithmic approach to network location problems: Part 2, The p-medians. *SIAM J. Appl. Math.* **1979**, *37*, 539–560. [CrossRef]
109. Osman, A.; Erkut, E.; Drezner, Z. An Efficient Genetic Algorithm for the p-Median Problem. *Ann. Oper. Res.* **2003**, *122*, 21–42.
110. Marianov, V.; Sara, D. *p-Median Models in Public Sector, Facility Location: Applications and Theory*; Horst, W., Hamacher, Z.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 119–143.
111. Sadeghi, A.H.; Sun, Z.; Sahebi-Fakhrabad, A.; Arzani, H.; Handfield, R. A Mixed-Integer Linear Formulation for a Dynamic Modified Stochastic p-Median Problem in a Competitive Supply Chain Network Design. *Logistics* **2023**, *7*, 14. [CrossRef]
112. Hoffmann, R.; Désérable, D.; Seredyński, F. Cellular automata rules solving the wireless sensor network coverage problem. *Nat. Comput.* **2022**, *21*, 417–447. [CrossRef]
113. Bongartz, I.; Calami, P.H.; Conn, A.R. A Projection Method for Lp Norm Location Allocation Problems. *Math. Program.* **1994**, *66*, 283–312. [CrossRef]
114. Beasley, J.E. OR-Library: Distributing test problems by electronic mail. *J. Oper. Res. Soc.* **1990**, *41*, 1069–1072. [CrossRef]
115. Kiani, F.; Randazzo, G.; Yelmen, I.; Seyyedabbasi, A.; Nematzadeh, S.; Anka, F.A.; Erenel, F.; Zontul, M.; Lanza, S.; Muzirafuti, A. A Smart and Mechanized Agricultural Application: From Cultivation to Harvest. *Appl. Sci.* **2022**, *12*, 6021. [CrossRef]
116. Khangahi, F.; Kiani, F. The Role of Social Networks in the Formation of Social Lifestyle Changes Caused by the Covid-19. *Int. J. Recent Technol. Eng.* **2021**, *9*, 263–267. [CrossRef]
117. Dehghan Khangahi, F. Ecological Problems and Social Mobilization: The Case of Urmia Lake. Ph.D. Thesis, Istanbul University, Istanbul, Turkey, 2020.
118. Available online: https://www.undp.org/sustainable-development-goals (accessed on 3 February 2023).
119. Loucks, D.P.; van Beek, E. Water Resources Planning and Management: An Overview. In *Water Resource Systems Planning and Management*; Springer: Cham, Switzerland, 2017; pp. 1–33.

120. Setó-Pamies, D.; Papaoikonomou, E. Sustainable Development Goals: A Powerful Framework for Embedding Ethics, CSR, and Sustainability in Management Education. *Sustainability* **2020**, *12*, 1762. [CrossRef]
121. Eckert, E.; Kovalevska, O. Sustainability in the European Union: Analyzing the Discourse of the European Green Deal. *J. Risk Financ. Manag.* **2021**, *14*, 80. [CrossRef]
122. Bhavya, R.; Elango, L. Ant-Inspired Metaheuristic Algorithms for Combinatorial Optimization Problems in Water Resources Management. *Water* **2023**, *15*, 1712. [CrossRef]
123. Kumar, V.; Yadav, S.M. A state-of-the-Art review of heuristic and metaheuristic optimization techniques for the management of water resources. *Water Supply* **2022**, *22*, 3702–3728. [CrossRef]
124. Torkomany, M.R.; Hassan, H.S.; Shoukry, A.; Abdelrazek, A.M.; Elkholy, M. An enhanced multi-objective particle swarm optimization in water distribution systems design. *Water* **2021**, *13*, 1334. [CrossRef]
125. Maier, H.R.; Kapelan, Z.; Kasprzyk, J.; Kollat, J.; Matott, L.S.; Cunha, M.C.; Dandy, G.C.; Gibbs, M.S.; Keedwell, E.; Marchi, A.; et al. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environ. Model. Softw.* **2022**, *62*, 271–299. [CrossRef]
126. Kayhomayoon, Z.; Babaeian, F.; Ghordoyee Milan, S.; Arya Azar, N.; Berndtsson, R. A Combination of Metaheuristic Optimization Algorithms and Machine Learning Methods Improves the Prediction of Groundwater Level. *Water* **2022**, *14*, 751. [CrossRef]
127. Mi, N.; Hou, J.; Mi, W.; Song, N. Optimal spatial land-use allocation for limited development ecological zones based on the geographic information system and a genetic ant colony algorithm. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 2174–2193. [CrossRef]
128. Ming, B.; Chang, J.X.; Huang, Q.; Wang, Y.M.; Huang, S.Z. Optimal Operation of Multi-Reservoir System Based-On Cuckoo Search Algorithm. *Water Resour. Manag.* **2015**, *29*, 5671–5687. [CrossRef]