Mathematical and Computational Applications

MDPI

Article

Chaotic Artificial Rabbits Optimization for Minimax Problems

Amira A. Allam [1,2], Mohamed A. Tawhid [3] and Mahmoud Owais [4,5,*]

[1] Mathematics Department, Faculty of Science, Assiut University, Assiut 71515, Egypt; amirahm@science.aun.edu.eg
[2] Basic Science Department, Faculty of Engineering, Sphinx University, New Assiut 71684, Egypt
[3] Department of Mathematics and Statistics, Faculty of Science, Thompson Rivers University, Kamloops, BC V2C 0C8, Canada; mtawhid@tru.ca
[4] Civil Engineering Department, Faculty of Engineering, Assiut University, Assiut 71515, Egypt
[5] Civil Engineering Department, Faculty of Engineering, Sphinx University, New Assiut 71684, Egypt
* Correspondence: maowais@aun.edu.eg or mahdmohd@eng.aun.edu.eg

Abstract

Numerous engineering problems can be represented as minimax optimization problems, including machine learning, classification, robust optimal control, signal processing, game theory, and more. Typically, minimax problems are considered challenging, especially constrained ones. The recently introduced artificial rabbits optimization (ARO) is inspired by the natural behaviour of rabbits. ARO exhibits robust effectiveness in tackling optimization challenges. Despite its advantages, ARO converges early to local optima, especially in complex or multi-modal optimization problems, and it struggles to balance exploration and exploitation, often leading to premature convergence and reduced accuracy. In this paper, we present a chaotic ARO that employs five maps exhibiting randomization behaviour to refresh candidate solutions. We assess the performance of the suggested CARO by applying it to 46 benchmark functions (25 unconstrained and 21 non-smooth minimax) and 15 constrained test functions with diverse characteristics. We evaluate its performance against six swarm intelligence algorithms. Also, we employ the chaotic maps to ARO and the six compared algorithms, and we perform a non-parametric statistical test, the Friedman test, on all outcomes. The findings show that the proposed algorithm can solve both unconstrained and constrained minimax problems more effectively and efficiently than other swarm intelligence methods.

Keywords: artificial rabbits optimization; chaos; metaheuristic; particle swarm optimization; unconstrained and constrained minimax problems

Academic Editor: Guillermo Valencia-Palomo

Received: 4 April 2026
Revised: 10 May 2026
Accepted: 14 May 2026
Published: 17 May 2026

1. Introduction

Minimax optimization problems are prevalent in diverse fields, including structural design [1], classification [2], game theory [3], machine learning, signal processing, finance [4], discrete optimization [5], engineering design [6], robust combinatorial optimization [7], and control theory [8,9]. Traditional deterministic methods for solving minimax problems include algorithms based on smoothing techniques [10–12] and gradient-based approaches like Sequential Quadratic Programming (SQP) [13–15]. These approaches have limitations: smoothing methods can become ill-conditioned due to rapidly decreasing parameters; SQP methods require exact second derivatives, making efficient implementation difficult; the quadratic programming component can be non-convex, complicating solution procedures [16].

Artificial rabbits optimization (ARO) [17] is a novel metaheuristic algorithm introduced in 2022, inspired by the survival strategies of rabbits in their natural environment. ARO is designed and used to discover approximate solutions for a wide range of optimization and search problems across domains such as engineering, finance, and machine learning. The core concept of ARO is to emulate rabbits' behaviours in their quest for survival, primarily through foraging and hiding strategies. These behaviours are translated into specific algorithm operators that manipulate and evolve a population of candidate solutions. The solutions, which represent potential answers to the target problem, are typically encoded as strings of numbers, characters, or more intricate data structures depending on the problem's nature. The ARO algorithm progressively improves these solutions across multiple generations. Each generation entails applying the foraging strategy, which explores the search space by generating new solutions based on the current best-known solutions and a degree of randomness. This replicates a rabbit's search for food. Simultaneously, the hiding strategy, which protects the best solutions from potential threats (such as local optima), is implemented. This can involve techniques such as solution diversification or local search around promising areas. The algorithm continues evolving the population of solutions through these operators, generation after generation, until a solution that meets the preset criteria (e.g., a minimum acceptable solution quality or a maximum number of iterations) is found. However, the algorithm has serious issues that make it difficult to balance exploration and exploitation when handling complex, multimodal challenges. The iterative process of ARO is carried out by the rabbits choosing a burrow at random, which can speed up convergence but also reduce solution diversity and confine the rabbits in local optima. This is one of the main problems. The quality of the final answer may also suffer since ARO lacks a guidance technique to steer towards favourable regions during the iterative search. There is a great opportunity to improve by overcoming the challenge of ensuring that the increased population reaches potential regions. Basically, the old-fashioned ARO algorithm has to be made more effective.

The ARO algorithm has been successfully used in a number of fields, such as industrial engineering [18,19], sentiment analysis [20], allocation of solar photovoltaic (PV) systems [21], predicting water productivity [22], finding the optimum values of uncertain parameters for the proton exchange membrane fuel cell model [23], engineering problems [24], multi-level thresholding color image segmentation [25], feature selection in high-dimensional data [26], and various applications and problems [27]. However, the original ARO has several problems, including an imbalance between exploration and exploitation, low population diversity, and premature convergence [28]. ARO is a newly developed optimization algorithm with many benefits, such as simplicity, a limited number of parameters, and being an intuitive idea, despite its disadvantages.

Multiple studies have proposed methods to enhance the performance of artificial rabbits optimization (ARO). For instance, Mazloumi et al. (2023) [29] introduced enhanced artificial rabbits optimization (EARO), which employs a nonlinear inertia-weight reduction technique to mitigate premature convergence. Cao et al. (2023) [30] developed an improved ARO variant, termed IARO, which accelerates convergence by adjusting the inertia weight during the later algorithmic stages. Awadallah et al. (2023) [31] incorporated position-based learning and the Lévy flight to enhance population diversity and achieve a balance between exploration and exploitation; their approach demonstrated effectiveness in feature selection for medical diagnostic data. Vellingiri et al. (2023) [32] proposed ARSCA, an ARO-based method that integrates a sine–cosine optimizer to optimize host capacity control parameters more effectively.

In order to reach equilibrium between exploration and exploitation, Kumar et al. (2023) [33] suggested an enhanced form of ARO called LFCARO, which employs a tent chaos random distribution and a Lévy flight random walk. This method was effectively used for resource scheduling in home energy systems. CHAOARO, an improved ARO, was proposed by Wang et al. in 2022 [18]. They improved ARO's exploring capabilities by combining it with the Aquila optimizer. The adaptive switching mechanism and chaotic opposition-based learning technique were used to balance its exploration and exploitation capabilities. Alamir et al. (2023) [34] proposed QARO, an improved ARO that successfully applies quantum mechanics based on the Monte Carlo method to improve convergence accuracy in energy management optimization in microgrids. Wang et al. (2022) [19] proposed an improved ARO called LARO. The Lévy flying technique is used during the exploitation phase to improve the accuracy of ARO's convergence. Using the selection–opposition strategy improves the algorithm's ability to evade local optima. To greatly enhance the search capabilities and convergence accuracy of ARO, He et al. (2025) [24] proposed an enhanced multi-strategy algorithm, termed MARO, by combining several methods, including elite opposition-based learning, Levy flights, random walks, and adaptive weight factors. In order to solve the QoS-aware web service composition challenge, Sharif et al. (2025) [35] introduced an upgraded chaotic-based approach, named ECARDO, a hybrid artificial rabbit optimization and dandelion optimizer based on Singer and Intermittency maps.

These ARO improvement studies offer valuable references for further improvement investigations. There are still certain issues, even though the aforementioned researchers have helped ARO operate more effectively. The loss of population variety, for example, has not been resolved despite ARSCA increasing the pace of ARO convergence. The poor convergence performance of EARO, QARO, and IARO is due to their disregard for population diversity and the proper balance between exploration and exploitation.

The aim of this study is to balance the exploration and exploration capabilities of the ARO algorithm while keeping the benefits of its straightforward principles and strong operability. This is done while taking the aforementioned reason into account. To accomplish this, this article proposes a variant of ARO, called CARO, and employs chaotic map-based randomization to enhance the exploration and exploitation behaviour of the original ARO algorithm.

The following is a summary of this paper's main contributions:

- Introduce a new version of ARO, called CARO, by combining CRO with five types of chaotic maps: Chebyshev, circle, Gauss, logistic, and tent.
- Test the original ARO, Slime Mold Algorithm (SMA) [36], Egret Swarm Optimization Algorithm (ESOA) [37], Hunger Games Search (HGS) [38], Marine Predators Algorithm (MPA) [39], crystal structure algorithm (CSA) [40], and particle swarm optimization (PSO) [41] by solving 46 benchmark unconstrained minimax problems (including 25 smooth and 21 non-smooth minimax functions) and 15 constrained minimax optimization problems, demonstrating the effectiveness of the introduced method.
- Combine each chaotic map with ARO and with each compared algorithm (SMA, ESOA, HGS, MPA, CSA, and PSO) to ensure fairness in comparisons.
- Conduct a non-parametric statistical test, the Friedman test, on the outcomes of the proposed algorithm and the compared algorithms. The results show that the proposed algorithm is more effective and faster than the compared algorithms for both constrained and unconstrained minimax problems.

The rest of this paper follows this structure. The main structure and methods for the original ARO are highlighted in Section 2, along with the design structures of the proposed approaches. In Section 3, algorithmic implementations of the proposed methods and numerical experiments are discussed, and six different algorithms for unconstrained

https://doi.org/10.3390/mca31030083

and restricted minimax optimization problems are compared. The work's results and suggestions for additional research are presented in Section 4.

2. Enhanced Artificial Rabbits Optimization

The theory underlying artificial rabbits optimization (ARO) is first introduced in this section. Next, we present the performance and behaviour of five distinct chaotic maps. Lastly, we offer a thorough description of the suggested technique, known as Chaotic Artificial Rabbits Optimization (CARO).

2.1. Artificial Rabbits Optimization (ARO)

Since rabbits are herbivores that primarily consume grass, forbs, and leafy weeds, ARO was inspired by their natural survival strategies. Rabbits must adapt to survive, just like all other organisms throughout our evolutionary history [42]. Rabbits always look for food away from their nests, avoiding the grass close to their burrows to keep predators from discovering their nest. Because of their exceptionally broad range of vision and the fact that the majority of them are used for overhead scanning, rabbits can locate food over a wide region with ease [43,44]. This is seen as the process of exploration.

Another survival strategy of rabbits is random concealment. Because they are skilled at digging tunnels for their nests, rabbits will build numerous burrows around their nest and select one at random to protect it from predators [45]. Rabbits run more quickly because of their powerful muscles and tendons, short forelegs, and long hind legs. Rabbits can also run in a zigzag pattern to avoid pursuit, stop abruptly while running, or turn sharply [46]. By using this survival tactic, rabbits can effectively raise their chances of surviving by confusing their adversaries and evading detection.

2.2. Chaotic Map for ARO

Table 1 displays the five chaotic maps in their mathematical form. As seen in Figure 1, the 10 chaotic maps exhibit random behaviour despite the absence of random variables. In order to boost the algorithm's unpredictable behaviour and help prevent local optima from stagnating, we investigate how the five chaotic maps behave when they are invoked in the suggested algorithm. A random integer between [0, 1] serves as the starting point for each of the five chaotic maps. Because it was chosen in [47], we start with $x_0 = 0{:}7$.

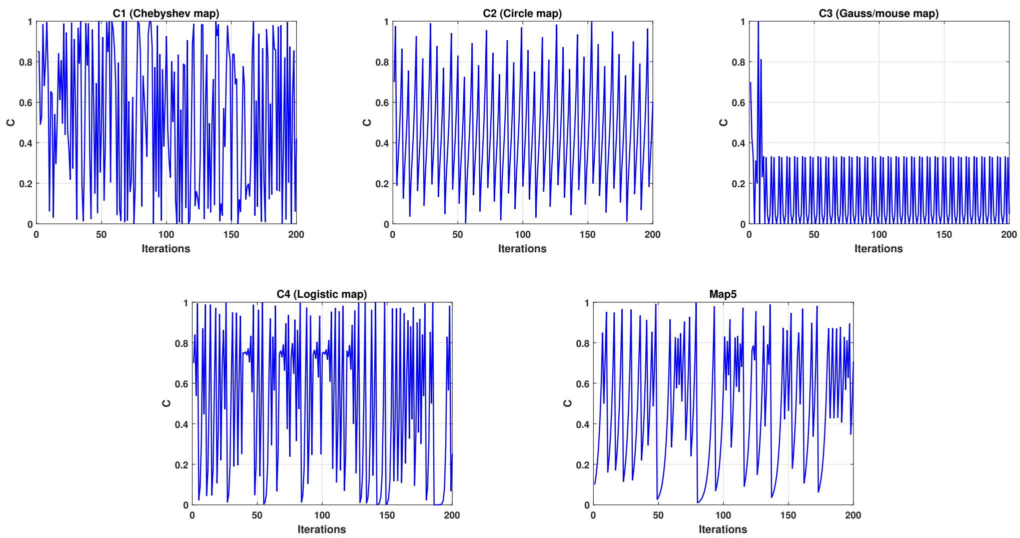

Figure 1. Visualization of chaotic maps.

Table 1. Chaotic maps.

No.	Name	Chaotic Map	Range
C1	Chebyshev [48]	$x_{i+1} = \cos\left(i \cos^{-1}(x_i)\right)$	$(-1, 1)$
C2	Circle [49]	$x_{i+1} = \mathrm{mod}\left(x_i + b - \left(\frac{a}{2\pi}\right)\sin(2\pi x_i), 1\right), \ a = 0.5, \ b = 0.2$	$(0, 1)$
C3	Gauss/mouse [50]	$x_{i+1} = \begin{cases} 1, & x_i = 0, \\ \frac{1}{x_i}\,\mathrm{mod}(ax_i, 1), & \text{otherwise} \end{cases}$	$(0, 1)$
C4	Logistic [51]	$x_{i+1} = ax_i(1 - x_i), \ a = 4$	$(0, 1)$
C5	Tent [52]	$x_{i+1} = \begin{cases} \frac{x_i}{0.7}, & x_i < 0.7, \\ 3(1 - x_i), & x_i \geq 0.7 \end{cases}$	$(0, 1)$

2.3. Chaotic Maps for Increasing Randomization Behaviour in ARO

The fundamental behaviour of rabbits, such as guided detour foraging (exploration), random hiding (exploitation), and energy shrink (transition from exploration to exploitation), is the basis of the ARO algorithm. Through the use of strategies demonstrated in Equations (3), (9) and (16), N_1 in the detour foraging strategy is a guided factor in the ARO algorithm. The algorithm produces a variety of solutions. They are subject to the normal distribution $N(0, 1)$ with mean = 0 and standard deviation = 1.

But in certain optimization situations, especially complex and high-dimensional ones, the ARO method may become stuck in local optima, leading to an inmproper exploration–exploitation balance. Search space solutions are not well covered by the algorithm since each solution modifies its position according to the prior parameters. In order to overcome these constraints, we have created a new iteration of the ARO algorithm that is more diverse by replacing these parameters with others that rely on chaotic maps. The great majority of values will fall roughly inside the range of $(-3, 3)$, even if N_1 is subject to the normal distribution of n $N(0, 1)$. The suggested approach converts the chaotic maps based on alternative random values to the same range $(-3, 3)$, as illustrated in Equation (1):

$$C(no) = \begin{cases} 3 \cdot \eta(no) & no = 1 \\ -6 \cdot \eta(i) + 3 & no = 2, 3, 4, 5 \end{cases} \tag{1}$$

where η is the chaotic map, and no is the number of the selected chaotic map in Table 1.

As demonstrated in Equation (2), two random strategies $N(0, 1)$ and the transformed chaos map $C(no)$ are combined to improve randomness and diversity in the search patterns, thereby increasing the randomization behaviour of the interaction between the solutions and the movement toward the overall best solution in the swarm:

$$CR = \begin{cases} C(no) & \rho \leq 0.5 \\ N(0, 1) & \text{otherwise} \end{cases} \tag{2}$$

The complex and variable trajectories produced by the combination are clearly shown in Figure 2, highlighting the algorithm's potential to increase search efficiency and robustness when tackling challenging optimization issues. A more comprehensive investigation of the search space is made possible by this improved chaotic behaviour, which successfully avoids convergence.

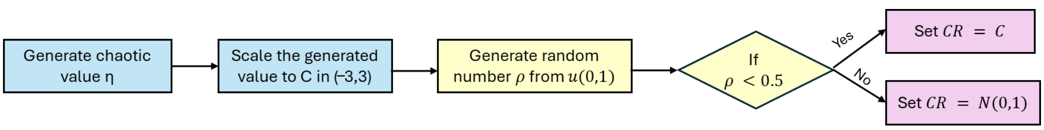

Figure 2. Randomization strategy used in CARO algorithm.

2.4. The Proposed Algorithm CARO

The ARO algorithm has limitations similar to those of other swarm intelligence systems, such as sluggish convergence and trapping in local minima, even though it is composed of three primary strategies: detour foraging (exploration), random hiding (exploitation), and energy shrink (transition from exploration to exploitation). To get over these two problems, we provide a swarm intelligence method in this study, CARO, or Chaotic Artificial Rabbits Optimization, which is the name of the suggested algorithm. During the exploration and exploitation phases of CARO, we replace a few random parameters with the converted chaotic maps $C(\text{no})$. The following is a description of the mathematical models of the three strategies:

2.4.1. Detour Foraging (Exploration)

ARO's detour foraging behaviour indicates that each search individual tends to add a perturbation and update its position in relation to the other search individual selected at random from the swarm. The following mathematical model of rabbit detour foraging is suggested:

$$\mathbf{V}_i(t+1) = \mathbf{x}_j(t) + S \cdot (\mathbf{x}_i(t) - \mathbf{x}_j(t)) + \text{round}(0.5 \cdot (0.05 + r_1)) \cdot \mathbf{N}_1, \tag{3}$$

where $i, j = 1, \ldots, Nu$ and $j \neq i$:

$$S = D \cdot \gamma \tag{4}$$

$$D = \left(e^{-e^{\left(-\frac{t-1}{T}\right)^2}} \right) \cdot \sin(2\pi r_2) \tag{5}$$

$$\gamma(k) = \begin{cases} 1, & \text{if } k = g(l) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

for $k = 1, \ldots, d$ and $l = 1, \ldots, \lfloor r_3 \cdot d \rfloor$. Here, $g = \text{randperm}(D)$ denotes a random permutation of the dimension indices $\{1, 2, \ldots, D\}$. Therefore, $\gamma(k) = 1$ is assigned to randomly selected dimensions according to the permutation vector g, while $\gamma(k) = 0$ is assigned to the remaining dimensions. Thus, the selected dimensions change randomly during the search process:

$$g = \text{randperm}(d) \tag{7}$$

$$\mathbf{N}_1 \sim \mathcal{N}(0, 1) \tag{8}$$

where V_i is the candidate position; Nu is the size of a rabbit population; d is the dimension of the problem; T is the maximal number of iterations; randperm returns a random permutation of the integers from 1 to d; r_1, r_2 and r_3 are three random numbers in $(0, 1)$, D is the running length, which represents the movement pace when performing the detour foraging, and N_1 is subject to the standard normal distribution.

2.4.2. Random Hiding (Exploitation)

In ARO, at each iteration, a rabbit always produces d burrows around it along each dimension of the search space, and it always randomly chooses one from all burrows for hiding to reduce the probability of being preyed on. The following equation is given in this regard. The jth burrow of the ith rabbit is generated by

$$\mathbf{B}_{m,j}(t) = \mathbf{x}_i(t) + \Theta \cdot \omega \cdot \mathbf{x}_i(t), \quad m = 1, \ldots, Nu, \quad j = 1, \ldots, D \tag{9}$$

$$\Theta = \frac{T - t + 1}{T} \cdot r_4 \tag{10}$$

$$\omega(i) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad = 1, \ldots, D \tag{11}$$

The D burrows are created in a rabbit's vicinity along all dimensions using Equation (9). Over the course of iterations, the hiding parameter, H, is linearly reduced from 1 to $1/T$ using a random perturbation [53]. This characteristic indicates that these burrows are first formed in a rabbit's larger neighborhood. This neighborhood also shrinks as the number of iterations rises. As previously stated, rabbits are frequently subjected to predatory chase and attack. Rabbits need to locate a secure hiding spot in order to live. They are therefore not allowed to choose a burrow at random for refuge in order to evade detection. The following equations are suggested to quantitatively model this random concealment strategy:

$$\mathbf{V}_m(t+1) = \mathbf{x}_m(t) + \Psi \cdot (r_5 \cdot \mathbf{B}_{m,q}(t) - \mathbf{x}_m(t)), \quad m = 1, \ldots, Nu \tag{12}$$

$$\gamma(n) = \begin{cases} 1, & \text{if } n = \lfloor r_6 \cdot D \rfloor \\ 0, & \text{otherwise} \end{cases} \quad n = 1, \ldots, D \tag{13}$$

$$\mathbf{B}_{m,q}(t) = \mathbf{x}_m(t) + \Theta \cdot \gamma \cdot \mathbf{x}_m(t) \tag{14}$$

$$\mathbf{x}_m(t+1) = \begin{cases} \mathbf{X}_m(t), & f(\mathbf{x}_m(t)) \le f(\mathbf{V}_m(t+1)) \\ \mathbf{V}_m(t+1), & f(\mathbf{x}_m(t)) > f(\mathbf{V}_m(t+1)) \end{cases} \tag{15}$$

This equation denotes that if the fitness of the candidate position of the th rabbit is better than that of the current one, the rabbit will abandon the current position and remain at the candidate position produced by either Equation (3) or Equation (12).

2.4.3. Energy Shrink (Switch from Exploration to Exploitation)

When it comes to ARO, rabbits usually engage in detour foraging during the early stages of iterations and random concealment during the later stages. This search mechanism is the outcome of a rabbit's energy, which gradually decreases over time. In order to simulate the transition from exploration to exploitation, an energy factor is created. The following is the definition of the energy factor in ARO:

$$K(t) = 4\left(1 - \frac{t}{T}\right) \ln \frac{1}{r} \tag{16}$$

where r is the random number in $(0,1)$. The energy factor $K(t)$ shows a declining trend towards zero during the iterations with the oscillation amplitude. A large value of the energy factor represents that a rabbit has sufficient energy and physical strength for detour foraging. In contrast, a small value of the energy factor indicates that a rabbit is less physically active, so it relies on random hiding. We can summarize the steps of the proposed algorithm (CARO) as follows. In Algorithm 1, the condition $K(t) \ge 1$ activates the exploration phase, while $K(t) < 1$ activates the exploitation phase. This switching rule follows the original ARO energy-based mechanism and allows the search process to gradually move from global exploration to local exploitation as the number of iterations increases. In Equation (3), transformed chaotic maps $C(\text{no})$ have been applied instead of N_1 to offer random interaction between solutions or progress toward the best solution. Figure 3 represents the framework of the proposed algorithm CARO and shows that CARO is avoids local minima through the solutions' random movement. The proposed algorithm CARO's primary phases are displayed in Algorithm 1.

Algorithm 1 The steps of the proposed CARO algorithm

Initialize the population $\mathbf{X}_i$, $i = 1, 2, \ldots, N_u$ randomly within the problem bounds.
Evaluate fitness $f(\mathbf{X}_i)$ for each rabbit.
Assign $\mathbf{X}_{best}$ as the current best solution.
for $t = 1$ to T **do**
  Compute energy factor from Equation (16)
  if $K(t) \geq 1$ **then** ▷ Exploration: Detour Foraging
    for $i = 1$ to N_u **do**
      Select $j \neq i$ randomly
      Compute S from Equation (4), with D in Equation (5).
      Generate random variable **RC** based on transformed map $C(\text{no})$
      Update position from Equation (3)
    end for
  else
    for $m = 1$ to N_u **do** ▷ Exploitation: Random Hiding
      Generate D burrows $\mathbf{B}_{m,q}(t)$ using Equation (14)
      Select a burrow randomly
      Compute candidate as in Equation (12)
      Apply greedy selection:
      if $f(\mathbf{V}_m(t+1)) < f(\mathbf{x}_m(t))$ **then**
        $\mathbf{x}_m(t+1) = \mathbf{V}_m(t+1)$
      else
        $\mathbf{x}_m(t+1) = \mathbf{x}_m(t)$
      end if
    end for
  end if
  Enforce bounds on all updated positions
  Update $\mathbf{X}_{best}$ if a better solution is found
end for
return $\mathbf{X}_{best}$

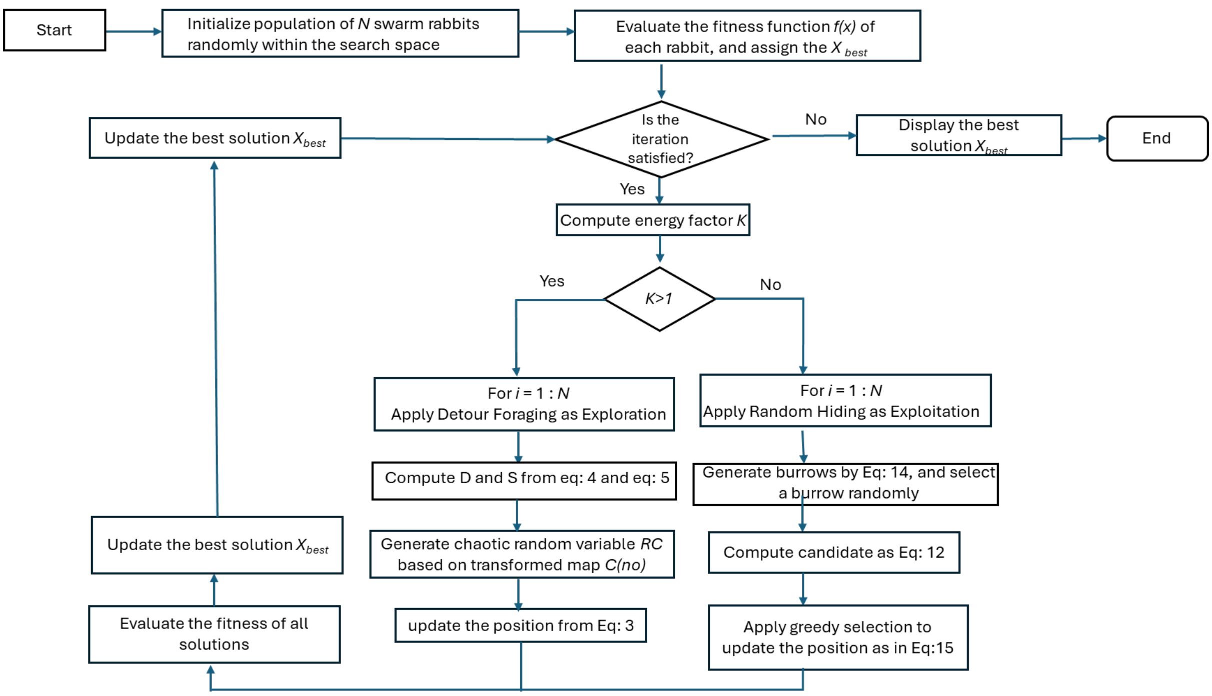

Figure 3. Flowchart of the proposed Chaotic Artificial Rabbits Optimization (CARO).

Although formal convergence analysis is valuable, the present study follows the common evaluation framework used in metaheuristic optimization research, where newly proposed variants are mainly assessed through benchmark functions, convergence behaviour, statistical tests, and comparative performance against established algorithms. Since CARO is designed as a derivative-free population-based optimizer for nonlinear, non-smooth, and constrained minimax problems, its effectiveness is evaluated empirically using extensive benchmark experiments. To claim a closed-form convergence guarantee, the numerical results, convergence curves, statistical analysis, and computational time comparison are used to demonstrate its practical optimization performance. The proposed CARO algorithm preserves the bounded search-space structure of the original ARO algorithm. At each iteration, candidate solutions are restricted within the predefined lower and upper bounds. Moreover, the best-so-far solution is retained during the optimization process. Therefore, for a minimization problem, the best objective value obtained up to iteration $t+1$ is not worse than the best objective value obtained up to iteration t. This produces a non-increasing best-so-far objective sequence during the search process.

3. Numerical Experiments

This section examines the performance of the proposed algorithm on both unconstrained and constrained benchmark functions. The proposed algorithm was implemented in MATLAB R2015a and executed on a personal computer equipped with an Intel(R) Core(TM) i5 processor, running the Windows 11 operating system.

The proposed algorithm was compared with six swarm intelligence algorithms. The parameter settings of the proposed algorithm and the comparative results against the other swarm intelligence algorithms are presented in the following subsections.

The numerical results are organized according to the type of benchmark problem. In each table, the lowest rank indicates the best-performing method.

3.1. Unconstrained Minimax Optimization Test Problems

In this subsection, we report the unconstrained minimax functions F_1–F_{25} and nonsmooth unconstrained minimax functions F_{26}–F_{46}, as shown in Tables 2 and 3, respectively.

Table 2. Benchmark unconstrained minimax optimization test problems, their dimensions (N), and known optimum objective values.

Problem	Function	N	Optimum
F_1	CB2 [54]	2	1.9522245
F_2	WF [55]	2	0
F_3	SPIRAL [55]	2	0
F_4	EVD52 [56]	6	3.5997193
F_5	Rosen–Suzuki [57]	4	-44
F_6	Polak 6 [58]	4	-44
F_7	PBC3 [59]	5	$0.42021427 \times 10^{-2}$
F_8	Bard [60]	3	$0.50816327 \times 10^{-1}$
F_9	Kowalik–Osborne [60]	4	$0.80843684 \times 10^{-2}$
F_{10}	Davidon 2 [60]	4	115.70644
F_{11}	OET5 [61]	4	0.2635735×10^{-2}
F_{12}	OET6 [61]	4	$0.20160753 \times 10^{-2}$
F_{13}	GAMMA [62]	4	$0.12041887 \times 10^{-2}$
F_{14}	EXP [63]	5	$0.12237125 \times 10^{-1}$
F_{15}	PBC1 [59]	6	$0.22340496 \times 10^{-1}$
F_{16}	EVD61 [56]	5	$0.34904926 \times 10^{-1}$
F_{17}	Filter [64]	9	$0.61852848 \times 10^{-2}$
F_{18}	Wong 1 [65]	7	680.63006
F_{19}	Wong 2 [65]	10	24.306209
F_{20}	Wong 3 [65]	20	133.72828
F_{21}	Rosen-Suzuki [12]	4	-44
F_{22}	U_1 [12]	2	2
F_{23}	Polak 3 [58]	11	261.08258
F_{24}	Watson [60]	20	$0.14743027 \times 10^{-7}$
F_{25}	Osborne 2 [60]	11	$0.48027401 \times 10^{-1}$

Table 3. Nonsmooth unconstrained minimax optimization test problems, their dimensions (N), and known optimum objective values.

Problem	Function	N	Optimum
F_{26}	QL [66]	2	7.20
F_{27}	Crescent [66]	2	0
F_{28}	CB2 [66]	2	1.9522245
F_{29}	CB3 [66]	2	2
F_{30}	DEM [66]	2	−3
F_{31}	LQ [66]	2	−1.4142136
F_{32}	Mifflin 1 [66]	2	−1
F_{33}	Mifflin 2 [66]	2	−1
F_{34}	Shor [66]	5	22.600162
F_{35}	Rosen–Suzuki [66]	4	−44
F_{36}	HS78 [67]	5	−2.9197004
F_{37}	El-Attar [56]	6	0.5598131
F_{38}	Gill [68]	10	9.7857721
F_{39}	Maxq [66]	20	0
F_{40}	Maxl [66]	20	0
F_{41}	MAXHILB [68]	50	0
F_{42}	Function 42 [12]	7	247
F_{43}	L1HILB [68]	50	0
F_{44}	Function 44 [69]	2	0
F_{45}	Function 45 [69]	10	0
F_{46}	Function 46 [62]	2	0

3.2. Parameter Settings

In Table 4, we give the parameter settings of the proposed algorithm. We select the number of population size according to the experimental test.

Table 4. Parameter settings of the proposed algorithm.

Parameter	Description	Value
N	Population size	30
f_{count}	Maximum function evaluations	24,000
Max_{itr}	Maximum number of iterations	800
ε	Accepted error tolerance	10^{-6}

3.3. The Performance of the Chaotic Maps for ARO Algorithm on Minimax Problems

In the proposed CARO algorithm, selected random variables in the original ARO structure are replaced with the transformed chaotic parameter $RC(no)$, which is generated from chaotic maps. This modification aims to improve the randomness and distribution of the search process. To identify the most suitable chaotic map for the proposed algorithm, the performance of five different chaotic maps was compared.

Using the five chaotic maps in the ARO algorithm, Figure 4 shows the average error curves over 30 independent runs for representative minimax test problems. The selected functions F_1, F_3, F_{12}, F_{14}, F_{29}, F_{41}, F_{44}, and F_{46} were chosen because they represent different optimization characteristics and allow the behaviour of the chaotic maps to be visually compared under different search landscapes. The results show that the C5 map generally provides faster convergence during the early iterations and reaches low error values with stable convergence behaviour. Although C1 performs competitively on some functions, such as F_{41} and F_{46}, C5 shows a more consistent balance between fast convergence, low error values, and smooth decline across the selected problems. This suggests that the C5 map can improve the search behaviour of ARO and is a suitable candidate for the proposed CARO algorithm.

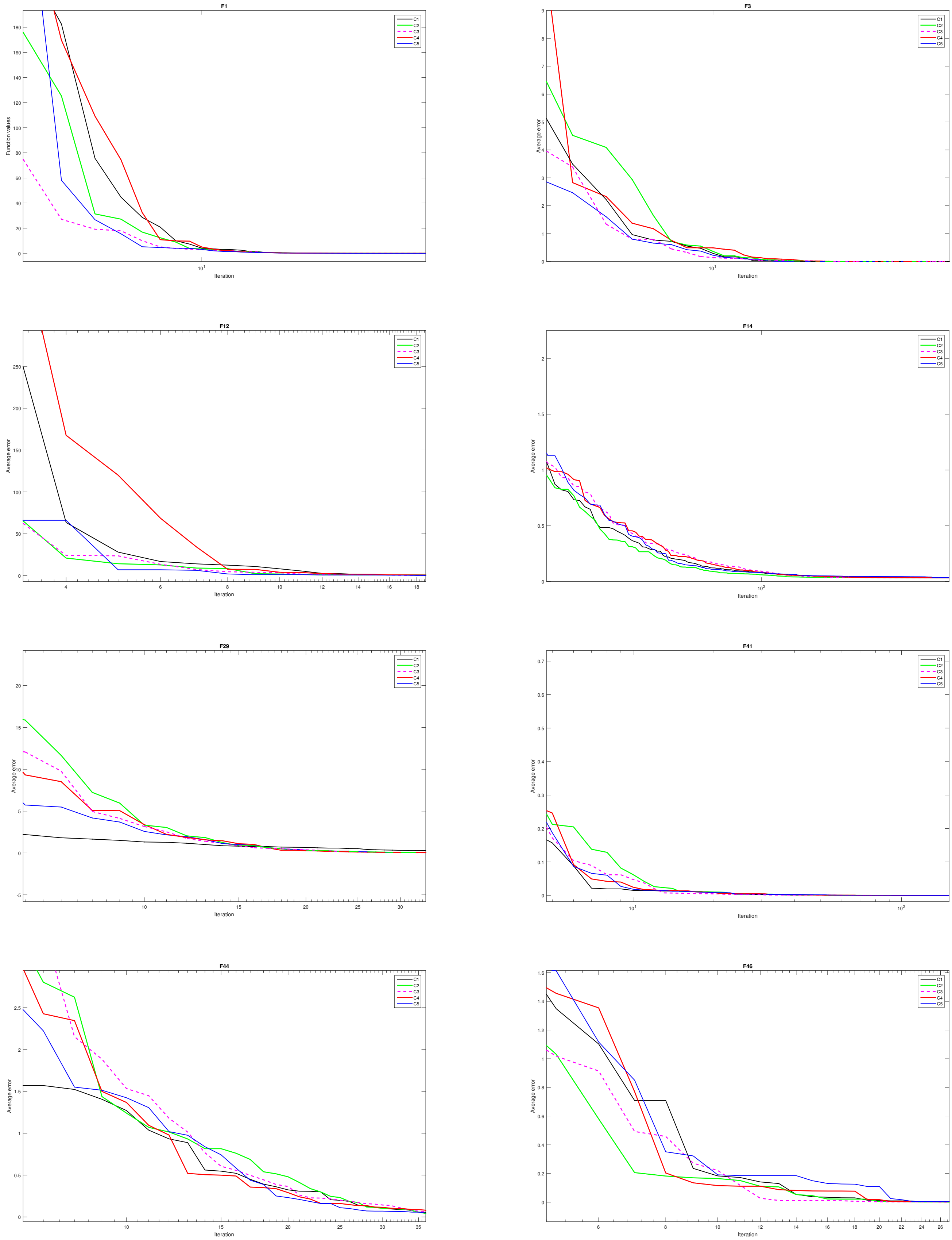

Figure 4. Performance of the five chaotic maps on the min–max test problems.

The normal random variable $N(0,1)$ was combined with the C5 chaotic map to form the combined randomization parameter $CR5$. This hybrid structure was used to combine the stochastic behaviour of the original random parameter with the deterministic irregularity of the chaotic sequence as shown in Figure 5. The aim is to improve the diversity of search patterns and reduce the probability of premature convergence in both smooth and non-smooth minimax optimization problems.

Figure 6 compares the average error curves of three randomization structures: the original random parameter R, the chaotic parameter $C5$, and the combined parameter $CR5$. The results show that $CR5$ generally achieves lower error values and faster convergence than using R or $C5$ alone. This indicates that combining the normal random variable with the Tent chaotic map provides a more effective randomization structure for the proposed algorithm. Therefore, $CR5$ was adopted in the final CARO framework.

The convergence behaviour shown in Figures 4 and 6 provides further insight into the search behaviour of the proposed CARO variants. Compared with the original randomization strategy, the chaotic-map-based variants generally show faster reduction in the average error and more stable convergence trends. This behaviour indicates that the use of chaotic maps improves the distribution of candidate solutions during the early search stage, enhances exploration, and reduces the probability of premature convergence. As the number of iterations increases, the search gradually shifts toward exploitation, allowing the algorithm to refine the best solutions more effectively. Therefore, the convergence comparison supports the role of chaotic maps in improving the balance between exploration and exploitation without changing the original structure of ARO.

For a fair computational comparison, all algorithms were executed under the same experimental settings, including the same population size, dimensionality, number of independent runs, and maximum number of iterations. No early stopping criterion was applied during CPU time measurement. Table 5 reports the average CPU time and relative overhead of the original ARO and the proposed CARO variants over both unconstrained and constrained benchmark functions.

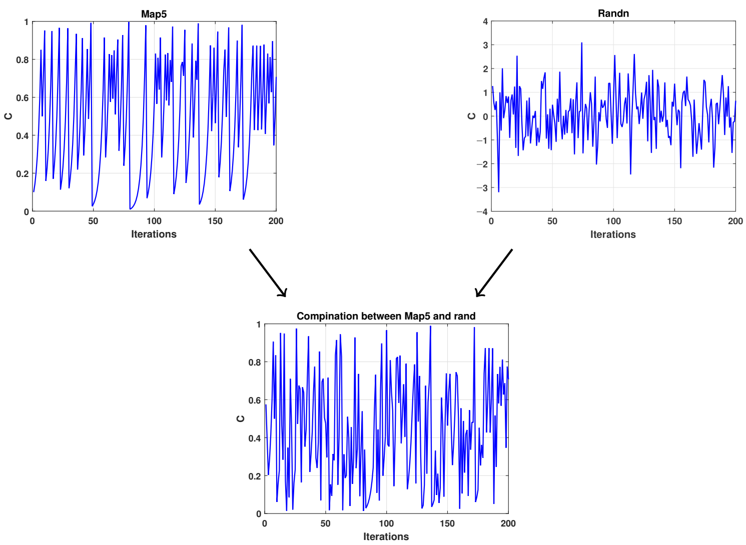

Figure 5. Visualization of chaotic maps and their combination. The third figure is the result of combining the first two figures.

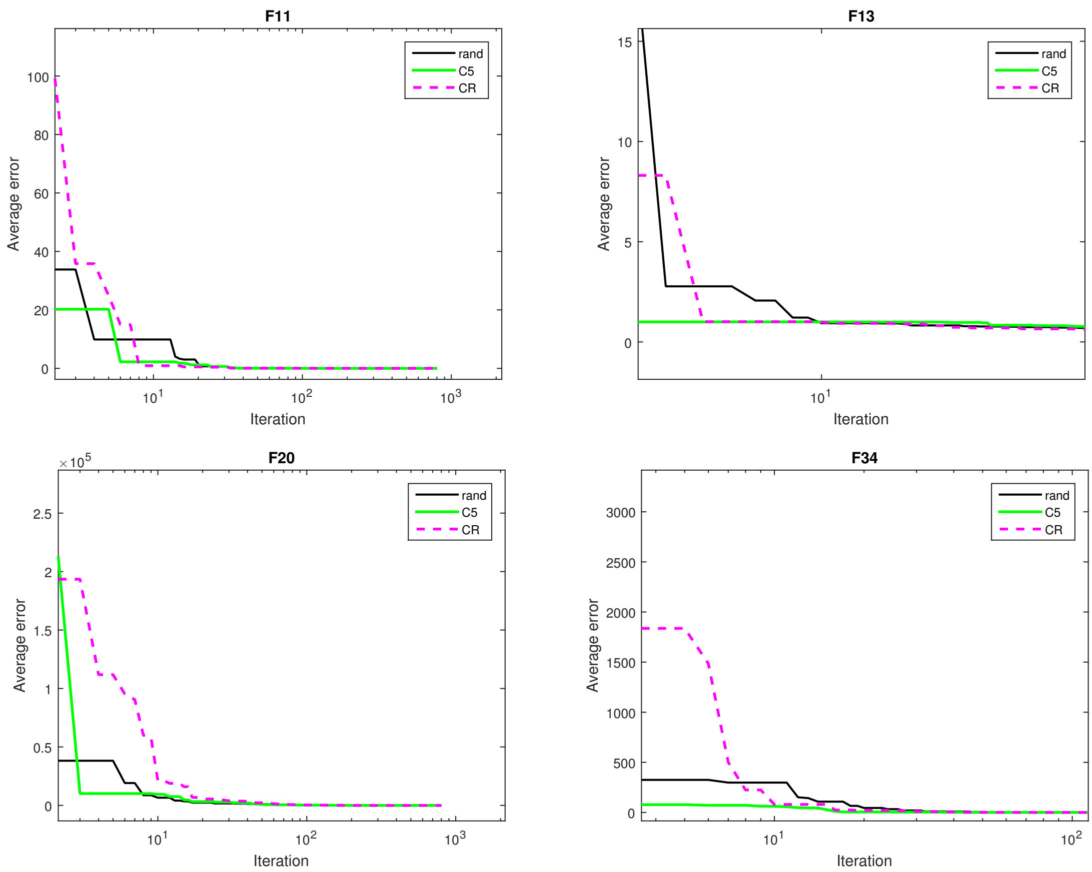

Figure 6. Comparison between C5, R, and the combined randomization parameter CR5.

Table 5. Computational time comparison between ARO and CARO variants over unconstrained and constrained benchmark functions under the fixed maximum-iteration condition.

Algorithm	Unconstrained CPU Time (s)	Unconstrained Overhead (%)	Constrained CPU Time (s)	Constrained Overhead (%)
ARO	2.1241 ± 1.8600	0.00	27.1146 ± 91.9227	0.00
ARO+C1	2.4735 ± 2.0914	16.45	28.8847 ± 96.7304	6.53
ARO+C2	3.2783 ± 3.5206	54.34	33.1468 ± 111.1213	22.25
ARO+C3	2.0639 ± 1.8234	-2.83	26.7053 ± 90.7947	-1.51
ARO+C4	2.0621 ± 1.8027	-2.92	39.3167 ± 133.4694	45.00
ARO+C5	2.1200 ± 1.8507	-0.19	32.8537 ± 110.8008	21.17

Note. CPU time is reported as mean ± standard deviation. The overhead was calculated relative to the original ARO algorithm. All algorithms preserve the same computational complexity order, $O(N \times T \times D)$, where N is the population size, T is the maximum number of iterations, and D is the problem dimension.

As shown in Table 5, all CARO variants preserve the same computational complexity order as the original ARO algorithm, namely $O(N \times T \times D)$, where N is the population size, T is the maximum number of iterations, and D is the problem dimension. For the unconstrained benchmark functions, the mean CPU time of the CARO variants remains close to that of ARO, with some variants showing slightly lower average CPU time. For the constrained benchmark functions, the CPU time values are higher because of the increased difficulty and constraint-handling operations associated with these problems. However, the chaotic-map-based modifications do not change the overall complexity class of the algorithm. The functions F_1, F_3, F_{12}, F_{14}, F_{29}, F_{41}, F_{44}, and F_{46} were selected as representative benchmark problems with different characteristics to analyze the behaviour of the tested chaotic maps under different optimization landscapes. These functions were not used as the only basis for selecting the final chaotic map. Instead, they were used to provide a focused behavioural comparison, while the average results over all benchmark functions were also considered to evaluate the overall performance of each chaotic map. Based on the representative behavioural analysis and the average results over all benchmark functions, the Tent map (C5) achieved the best overall performance among the tested chaotic maps under the adopted experimental setting. This behaviour can be attributed to its simple piecewise linear chaotic structure and its ability to generate well-distributed values within the interval $(0, 1)$. Such distribution improves the spread of candidate solutions during the early search stage, enhances exploration, and reduces the probability of premature convergence. As the iterations progress, this balanced perturbation behaviour supports exploitation and helps refine the best solutions. Therefore, C5 was selected as the main chaotic map in the proposed CARO algorithm. However, this conclusion is limited to the experimental setting of the present study and does not imply that the Tent map is universally superior for all optimization problems.

3.4. The Performance of CARO Accelerated Artificial Rabbits Optimization on Unconstrained Minimax Problems

Basic artificial rabbits optimization (ARO) [17], six Slime Mold Algorithms (SMA) [36], the Egret Swarm Optimization Algorithm (ESOA) [37], Hunger Games Search (HGS) [38], the Marine Predators Algorithm (MPA) [39], and the crystal structure algorithm (CSA) [40] were all compared against the suggested algorithm. All algorithms employ the same population size of 20 for fair comparison, and all algorithms are terminated when the error gap is less than 10^{-6} or the maximum function evaluation is reached. We provided the standard deviation (std), the best solution (fbest), the worst solution (fworst), and the average errors (mean) for each algorithm over a 30-run period. Each test problem's final row in Table 6 reports the Friedman rank test (Rank). To evaluate the statistical significance of the variations in performance among the six algorithms, the Friedman rank test is employed. The empirical results indicate CARO's efficiency when compared to other existing algorithms and show that it can get the optimum solutions for most test problems, and this demonstrates its efficiency in comparison with other existing algorithms.

The results shown in Table 6 illustrates that for smooth unconstrained min–max problems in Table 2, the average ranking of CARO is 2.16, ranking first, and it provides the optimal value among all algorithms on 15 functions. For nonsmooth mim–max problems, CARO ranks the first with average rank 2.0952 and has the optimal value among all algorithms on 15 functions of 21 test functions, as illustrated in Table 7. Therefore, it can be shown that CARO is a reliable optimization method in terms of performance when dealing with unconstrained min–max problems, especially with respect to non-smoothing problems.

For fair estimation, we applied the same chaotic map structure CR(5) all over the compared hybrid algorithms instead of the random parameter in each algorithm. The results of the average error over 30 runs for the algorithms are shown in Table 8. Table 9 shows that there is a considerable improvement for some algorithms' behaviour after applying the same randomization structure; however, the proposed algorithm is the most promising algorithm among them.

Table 6. Statistical results of the compared algorithms on unconstrained min–max optimization problems.

Function		ARO	CARO	MPA	HGS	CSA	ESOA	SMA	PSO
F_1	mean	8.0855×10^{-7}	5.38751×10^{-7}	5.19541×10^{-7}	0.00022733	7.39264×10^{-5}	0.005149456	7.39264×10^{-5}	8.97719×10^{-5}
	fbest	1.952224699	1.952225099	1.952224625	1.952228915	1.952225099	1.953548093	1.952225099	1.952225284
	fworst	1.952225454	1.952225340	1.952225417	1.953308216	1.952659552	1.966085192	1.952659552	1.952649277
	std	2.55427×10^{-7}	1.01954×10^{-7}	2.77524×10^{-7}	0.000478865	0.000131524	0.00362169	0.000131524	0.000187381
	Rank	2.5	2	1.75	7	4.625	8	4.625	5.5
F_2	mean	5.93513×10^{-7}	4.21184×10^{-7}	7.82511×10^{-7}	2.88309×10^{-7}	4.50329×10^{-7}	3.88173×10^{-7}	4.50329×10^{-7}	5.86334×10^{-7}
	fbest	1.43824×10^{-7}	4.60148×10^{-8}	3.92098×10^{-7}	0.211826×10^{-7}	2.42291×10^{-8}	1.41656×10^{-8}	9.51924×10^{-8}	2.3325×10^{-7}
	fworst	8.94337×10^{-7}	8.79379×10^{-7}	9.92412×10^{-7}	7.01652×10^{-7}	8.56045×10^{-7}	8.5993×10^{-7}	8.56045×10^{-7}	9.1445×10^{-7}
	std	2.69279×10^{-7}	2.66477×10^{-7}	2.25697×10^{-7}	3.06056×10^{-7}	2.6366×10^{-7}	2.67484×10^{-7}	2.6366×10^{-7}	3.11501×10^{-7}
	Rank	6.25	4	6.25	2.75	3.125	3	3.625	7
F_3	mean	5.25708×10^{-7}	2.92663×10^{-7}	7.25917×10^{-7}	3.10245×10^{-7}	3.12896×10^{-7}	3.56331×10^{-7}	3.12896×10^{-7}	6.46915×10^{-7}
	fbest	9.4757×10^{-8}	2.692×10^{-9}	3.75586×10^{-8}	1.86309×10^{-9}	6.60388×10^{-10}	2.83835×10^{-8}	6.60388×10^{-10}	4.07789×10^{-7}
	fworst	8.36415×10^{-7}	6.88592×10^{-7}	9.98653×10^{-7}	7.76567×10^{-7}	8.65113×10^{-7}	9.81527×10^{-7}	8.65113×10^{-7}	8.48433×10^{-7}
	std	3.01546×10^{-7}	2.43728×10^{-7}	3.10853×10^{-7}	4.13074×10^{-7}	3.1363×10^{-7}	2.8212×10^{-7}	3.1363×10^{-7}	1.69479×10^{-7}
	Rank	5	2	6.75	3.75	4.25	5	4.25	5
F_4	mean	5.66543×10^{-7}	7.08664×10^{-7}	6.75506×10^{-7}	0.000100758	3.03551×10^{-5}	0.008897717	3.03551×10^{-5}	0.000342901
	fbest	3.599719577	3.599719436	3.59971955	3.599725734	3.599720537	3.60318208	3.599720537	3.599720306
	fworst	3.599720274	3.59972029	3.599720259	3.599985726	3.59978753	3.622820699	3.59978753	3.600873015
	std	2.588×10^{-7}	2.4103×10^{-7}	2.36139×10^{-7}	9.9438×10^{-5}	1.97927×10^{-5}	0.00451313	1.97927×10^{-5}	0.000475793
	Rank	2.25	2.25	1.5	6.25	4.75	8	4.75	6.25
F_5	mean	2.84354×10^{-5}	6.11881×10^{-5}	44.01083689	0.092863516	0.037876946	0.686962216	0.037876946	0.246850998
	fbest	−43.99999709	−43.99999919	−0.005539902	−43.98412221	−43.99771703	−43.7118434	−43.99771703	−43.99106257
	fworst	−43.99989241	−43.99966169	0.038538478	−43.794324	−43.78805096	−42.95857636	−43.78805096	−43.34610102
	std	3.4061×10^{-5}	0.00010622	0.012007125	0.070948486	0.06431682	0.214071033	0.06431682	0.29802614
	Rank	1.25	1.75	6.75	5	4	7	4	6.25
F_6	mean	0.000467423	3.34697×10^{-5}	43.99664906	0.608845349	0.019973117	0.565142462	0.019973117	11.14941939
	fbest	−43.99999981	−43.9999939	−0.026049879	−43.99309655	−43.99800987	−43.7507561	−43.99800987	−43.93667706
	fworst	−43.99629001	−43.99987486	0.010589015	−41.21362029	−43.94163075	−43.13386969	−43.94163075	−1.53872448
	std	0.001148063	3.55632×10^{-5}	0.010876145	1.219046127	0.020492185	0.158257878	0.020492185	17.94699498
	Rank	1.75	1.25	6.75	6	3.75	5.75	3.75	7
F_7	mean	0.000524089	0.000366949	0.000856512	0.013916059	0.003128819	0.001842749	0.003128819	0.001150918
	fbest	0.004202969	0.004202666	0.004279669	0.006904545	0.004334541	0.004737592	0.004334541	0.004251341
	fworst	0.005847265	0.006189797	0.006074155	0.036288561	0.026676251	0.006584101	0.026676251	0.006052085
	std	0.000637871	0.000644253	0.000654694	0.013667752	0.006829752	0.000461722	0.006829752	0.000791611
	Rank	1.75	2.25	3.5	8	6.25	4.5	6.25	3.5
F_8	mean	0.023627036	0.027140917	0.023648876	1.002177477	0.613706671	0.036338868	0.613706671	0.034904975
	fbest	0.074443363	0.074443363	0.074443605	0.390210687	0.172767094	0.07460135	0.172767094	0.074443363
	fworst	0.074443363	0.10263802	0.074504195	2.064429305	1.278317118	0.102751202	1.278317118	0.102638402
	std	1.14671×10^{-14}	0.008942324	2.3341×10^{-5}	0.721844959	0.371983885	0.010327491	0.371983885	0.015442955
	Rank	1.25	2.75	2.5	8	6.5	4.75	6.5	3.75
F_9	mean	0.000706953	0.000876941	0.000416255	0.000543265	0.006819883	0.001103917	0.006819883	0.009629077
	fbest	0.008285484	0.008308769	0.008037142	0.008057455	0.00955956	0.008096601	0.00955956	0.00808464
	fworst	0.010821479	0.006368778	0.007189838	0.009710012	0.024751304	0.010408981	0.024751304	0.045994056
	std	0.001023709	0.000929729	0.000374968	0.000767586	0.004575277	0.000489695	0.004575277	0.016008229
	Rank	4.5	3.75	1.25	2.5	6.75	3.75	6.75	6.75
F_{10}	mean	2.57104×10^{-5}	1.70716×10^{-5}	5.82817×10^{-7}	0.046864604	0.02000145	6.165494454	0.02000145	0.097213686
	fbest	115.7064406	115.70644	115.7064399	115.7065202	115.7071287	117.1504288	115.7071287	115.7081525
	fworst	115.7065763	115.7065909	115.706441	115.7825102	115.7559549	127.6368529	115.7559549	116.1325601
	std	4.86475×10^{-5}	4.71155×10^{-5}	3.19314×10^{-7}	0.031130837	0.019263777	2.693184366	0.019263777	0.184062803
	Rank	2.75	2.25	1	5.5	4.75	8	4.75	7
F_{11}	mean	0.000770177	0.001072313	0.000378692	0.022311593	0.024782397	0.031676438	0.024782397	0.24341757
	fbest	0.002740591	0.002801439	0.002679888	0.012273871	0.005123741	0.015677809	0.005123741	0.084818393
	fworst	0.004891358	0.004960521	0.003418991	0.031118762	0.05869735	0.058900765	0.05869735	0.65111889
	std	0.000816341	0.000848247	0.000316636	0.007916981	0.016636829	0.011546551	0.016636829	0.234520701
	Rank	2	3	1	4.5	5.5	6.5	5.5	8
F_{12}	mean	0.010202463	0.013178559	0.009400735	0.036534667	0.030794146	0.047094253	0.030794146	0.07283459
	fbest	0.003457999	0.003409187	0.003189151	0.006711648	0.006924319	0.02857183	0.006924319	0.021220019
	fworst	0.019447058	0.024343092	0.020518672	0.113822153	0.089720831	0.083496387	0.089720831	0.090227872
	std	0.005573241	0.005730991	0.005277917	0.043833341	0.030418219	0.013910166	0.030418219	0.030034654
	Rank	2	2.75	1.25	6.5	5.5	5.75	5.5	6.75
F_{13}	mean	0.64787039	0.633565012	0.635195848	35.6381957	0.901679548	0.679111367	0.901679548	0.705750052
	fbest	0.587307512	0.587307423	0.58892685	0.748568383	0.711952687	0.641529337	0.711952687	0.650932915
	fworst	0.705504033	0.707279799	0.705699894	173.7755599	1.410756403	0.739611102	1.410756403	0.747678924
	std	0.060011372	0.061262999	0.050051465	77.22096158	0.273584838	0.028467684	0.273584838	0.034933013
	Rank	2.5	2.5	2.5	8	6.5	3.25	6.5	4.25

Table 6. *Cont.*

Function		ARO	CARO	MPA	HGS	CSA	ESOA	SMA	PSO
F_{14}	mean	0.034351426	0.034492406	0.032049482	0.037148309	0.036003423	0.053581048	0.036003423	0.149166561
	fbest	0.034265319	0.034269076	0.01336734	0.034426482	0.032385823	0.04215037	0.032385823	0.03472876
	fworst	0.034706928	0.034787896	0.034672893	0.045740372	0.039559366	0.156278197	0.039559366	0.386628162
	std	0.000194703	0.000174903	0.006610273	0.004778618	0.00178304	0.025156651	0.00178304	0.161745341
	Rank	2.5	3	2.25	5.75	3.75	7.25	3.75	7.75
F_{15}	mean	0.31826946	0.193309964	0.129328268	0.467707887	0.146348366	0.371384253	0.146348366	0.654159657
	fbest	0.010724931	0.029711643	0.023446562	0.02722811	0.028868546	0.140922208	0.028868546	0.570512541
	fworst	0.641671148	0.641302496	0.638828522	0.729087603	0.718982647	0.567145296	0.718982647	0.725923998
	std	0.327305339	0.303284193	0.258918582	0.315394034	0.288119401	0.114096417	0.288119401	0.062943343
	Rank	4.5	4.75	2	6.25	4.25	4	4.25	6
F_{16}	mean	0.049180922	0.119490784	0.061093815	0.241632529	0.322118837	0.321186396	0.322118837	0.56850712
	fbest	0.034933143	0.038156093	0.055792394	0.053854421	0.238152249	0.146480384	0.238152249	0.562400283
	fworst	0.26261027	0.517618099	0.145681833	0.466805579	0.410606877	0.607953639	0.410606877	0.620184129
	std	0.072790288	0.14618043	0.033235354	0.196565706	0.049181788	0.113033457	0.049181788	0.024202756
	Rank	2.25	4.5	2.25	5	5	5.75	5	6.25
F_{17}	mean	0.006057502	0.006047857	0.006068013	0.006044779	0.006039839	0.006036185	0.006039839	5.19425×10^{-7}
	fbest	0.000144359	0.000147824	0.000123671	0.000147828	0.000158996	0.000181428	0.000158996	0.00618565
	fworst	0.000114927	0.000114827	0.000114362	0.000115853	0.000115271	0.000126707	0.000115271	0.006184599
	std	1.36797×10^{-5}	1.40979×10^{-5}	2.64688×10^{-6}	1.38125×10^{-5}	1.12968×10^{-5}	1.31854×10^{-5}	1.12968×10^{-5}	4.27378×10^{-7}
	Rank	4.5	4.75	3	5.5	4.25	5.25	4.25	4.5
F_{18}	mean	0.231679861	0.260081304	0.0005936	0.654144305	0.254869706	13.70060705	0.254869706	1.180158444
	fbest	680.6489118	680.64214	680.6302058	680.8281985	680.6888561	684.3732894	680.6888561	680.9086565
	fworst	682.4575684	681.2467138	680.6321068	681.9642007	681.157529	704.0116037	681.157529	684.1382674
	std	0.561398934	0.236716774	0.000562143	0.436126583	0.177446415	4.468397683	0.177446415	1.336588362
	Rank	4.25	3.75	1	5.5	3.25	8	3.25	7
F_{19}	mean	2.698240992	1.950053824	0.515491377	4.998173212	1.303723346	81.78468332	1.303723346	6.215429479
	fbest	24.64800358	24.61014884	24.39566119	25.08685427	24.65825494	62.37761647	24.65825494	27.12776161
	fworst	32.35350843	29.99977381	25.25690616	34.50926845	27.5434298	139.9483402	27.5434298	33.76574355
	std	2.212345103	2.106921119	0.29523311	3.685373251	0.898693754	24.05342947	0.898693754	2.857244059
	Rank	4.5	3.5	1	6.5	3	8	3	6.5
F_{20}	mean	42.61315176	46.51947898	19.12187023	53.16628189	42.69034782	732.8194709	42.69034782	55.28829846
	fbest	101.1813297	93.15476406	138.5190109	104.5322223	117.4978287	523.3594364	117.4978287	133.7282801
	fworst	84.03253841	83.20077047	99.03608798	230.7751823	85.24414889	1244.696415	85.24414889	254.7987264
	std	5.762872582	3.362524602	21.57696286	57.87412626	9.447492304	206.8629965	9.447492304	44.16278239
	Rank	2	2	4.5	5.5	3.75	8	3.75	6.5
F_{21}	mean	9.91315×10^{-5}	9.03114×10^{-5}	44.00957572	0.10819175	0.008381767	0.534080349	0.008381767	0.0651652
	fbest	−43.99999914	−43.99999871	−0.004686067	−43.99990026	−43.99891819	−43.86057664	−43.99891819	−43.9992945
	fworst	−43.99957266	−43.99960237	0.047726808	−43.67418123	−43.97389947	−43.1778196	−43.97389947	−43.81914336
	std	0.00014016	0.000130695	0.015621326	0.132066869	0.008875777	0.186020974	0.008875777	0.070073203
	Rank	1.75	1.25	7.25	5.5	4	7.25	4	5
F_{22}	mean	6.49521×10^{-7}	6.64×10^{-7}	5.33376×10^{-7}	8.06471×10^{-7}	0.00010414	0.017645417	0.00010414	7.7356×10^{-7}
	fbest	2.000000334	2.000000202	2.000000088	2.000000642	2.000001051	2.003252943	2.000001051	2.000000608
	fworst	2.000000952	2.000000986	2.000000876	2.000000955	2.00029116	2.047319047	2.00029116	2.000000943
	std	2.22188×10^{-7}	2.68658×10^{-7}	2.80501×10^{-7}	1.31901×10^{-7}	9.69529×10^{-5}	0.011933494	9.69529×10^{-5}	1.52724×10^{-7}
	Rank	2.75	3.5	2	3.75	6.5	8	6.5	3
F_{23}	mean	257.377553	257.3779766	41267748565	1.0262×10^{176}	256.3919631	257.3136354	256.3919631	65535
	fbest	3.706157618	3.705642789	8109.291787	15.02643444	5.116659356	3.792206761	5.116659356	261.0825801
	fworst	3.703542661	3.703860752	3.81519×10^{11}	5.1305×10^{176}	4.498449568	3.74921382	4.498449568	65535
	std	0.000927086	0.000652002	1.19941×10^{11}	65535	0.172311078	0.013976067	0.172311078	46,155.63
	Rank	2.25	2.25	7.5	7.25	3.75	3	3.75	6.25
F_{24}	mean	0.45512054	0.452836465	0.506668437	0.466187393	0.541758734	1.804748845	0.541758734	3.091612298
	fbest	0.425637975	0.407745341	0.336879372	0.456506024	0.464358176	0.475614995	0.464358176	0.675035728
	fworst	0.485650706	0.47510754	0.699807746	0.493131983	0.774490054	3.839135048	0.774490054	7.808911191
	std	0.024142324	0.02086195	0.121491575	0.01533101	0.113899329	1.21303491	0.113899329	2.739077835
	Rank	2.5	1.5	3.75	2.75	5.25	7	5.25	8
F_{25}	mean	0.217882195	0.226574166	0.266177135	8.3388×10^{115}	0.751941073	0.300123	0.751941073	0.347914904
	fbest	0.188443045	0.240472725	0.311222882	3.355167173	0.667673618	0.299391472	0.667673618	0.309702002
	fworst	0.295724934	0.309672608	0.317940395	4.1694×10^{116}	1.603812742	0.310854252	1.603812742	0.739
	std	0.031014859	0.018427993	0.001896707	1.8646×10^{116}	0.289542599	0.0147022	0.289542599	0.191775829
	Rank	1.75	2.25	3.25	8	6.5	3	6.5	4.75

Table 7. Statistical results of the compared algorithms on non-smooth unconstrained min–max optimization problems.

Function		ARO	CARO	MPA	HGS	CSA	ESOA	SMA	PSO
F_{26}	errors	7.22353×10^{-7}	6.47088×10^{-7}	5.52722×10^{-7}	0.001020641	0.000403424	0.021264817	0.000403424	0.004141632
	fbest	7.200000141	7.200000164	7.200000215	7.200003283	7.200017845	7.200062746	7.200017845	7.200007038
	fworst	7.200000958	7.20000097	7.200000784	7.203380052	7.202942002	7.257595585	7.202942002	7.210064305
	std	2.27387×10^{-7}	2.63655×10^{-7}	1.91532×10^{-7}	0.001495594	0.000896508	0.014873704	0.000896508	0.005257071
	Rank	2	2.5	1.5	5.5	5	8	5	6.5
F_{27}	mean	5.95892×10^{-7}	5.54776×10^{-7}	8.61924×10^{-7}	4.5581×10^{-7}	5.10881×10^{-7}	4.43438×10^{-7}	5.10881×10^{-7}	5.08463×10^{-7}
	fbest	5.11635×10^{-9}	6.98972×10^{-8}	4.73853×10^{-7}	1.31184×10^{-12}	1.56735×10^{-7}	7.03791×10^{-8}	1.56735×10^{-7}	1.03173×10^{-7}
	fworst	9.95648×10^{-7}	9.88921×10^{-7}	9.99987×10^{-7}	9.77107×10^{-7}	9.9324×10^{-7}	9.96473×10^{-7}	9.9324×10^{-7}	9.44712×10^{-7}
	std	2.96575×10^{-7}	3.42353×10^{-7}	1.58669×10^{-7}	3.74411×10^{-7}	2.59535×10^{-7}	2.74746×10^{-7}	2.59535×10^{-7}	3.21436×10^{-7}
	Rank	5	4.75	6.25	3.25	4.5	4	4.5	3.75
F_{28}	mean	4.51069×10^{-7}	6.1464×10^{-7}	6.13574×10^{-7}	0.000118072	8.08233×10^{-5}	0.006737892	8.08233×10^{-5}	8.15859×10^{-5}
	fbest	1.952224551	1.952224516	1.952224645	1.952224892	1.952225961	1.95242701	1.952225961	1.952229294
	fworst	1.952225498	1.952225419	1.952225466	1.952507848	1.952500626	1.968185683	1.952500626	1.952488538
	std	3.32798×10^{-7}	2.98679×10^{-7}	2.80883×10^{-7}	0.000121893	9.63163×10^{-5}	0.004450958	9.63163×10^{-5}	0.000105279
	Rank	2.25	1.75	2	6.25	5	8	5	5.75
F_{29}	mean	8.31726×10^{-7}	6.91413×10^{-7}	6.09761×10^{-7}	5.92093×10^{-5}	0.000159181	0.009454854	0.000159181	0.000959734
	fbest	2.000000568	2.000000343	2.000000123	2.000000403	2.000020169	2.000858842	2.000020169	2.000002282
	fworst	2.000000951	2.000000908	2.000000868	2.000273628	2.000433649	2.027374838	2.000433649	2.003305128
	std	1.30625×10^{-7}	1.83203×10^{-7}	2.79345×10^{-7}	0.000120034	0.000141934	0.007752488	0.000141934	0.001331181
	Rank	2.75	2	1.5	3.75	5.75	8	5.75	6.5
F_{30}	mean	7.24925×10^{-7}	4.89659×10^{-7}	3.000247903	8.83266×10^{-7}	1.36058×10^{-6}	0.030347141	1.36058×10^{-6}	7.40942×10^{-7}
	fbest	−2.999999846	−2.999999671	−0.001319007	−2.999999166	−2.999999644	−2.993186607	−2.999999644	−2.999999588
	fworst	−2.999999031	−2.99999923	0.001343243	−2.99999905	−2.999992332	−2.937913681	−2.999992332	−2.999999023
	std	2.78305×10^{-7}	1.43259×10^{-7}	0.000668538	4.72845×10^{-8}	2.22585×10^{-6}	0.015603394	2.22585×10^{-6}	2.05768×10^{-7}
	Rank	2.5	1.5	7.75	3.25	5	7.25	5	3.75
F_{31}	mean	5.71403×10^{-7}	7.40697×10^{-7}	1.414237087	2.4424×10^{-5}	2.34573×10^{-5}	0.001324878	2.34573×10^{-5}	5.24618×10^{-5}
	fbest	−1.41421342	−1.414213195	-7.51681×10^{-5}	−1.41421155	−1.414212922	−1.413839527	−1.414212922	−1.414211433
	fworst	−1.414212701	−1.414212626	0.000152718	−1.414143736	−1.414150201	−1.411256559	−1.414150201	−1.414103002
	std	2.51464×10^{-7}	1.91315×10^{-7}	6.42071×10^{-5}	2.72122×10^{-5}	1.95591×10^{-5}	0.000798277	1.95591×10^{-5}	4.89672×10^{-5}
	Rank	1.25	1.75	7.75	5	3.5	7.25	3.5	6
F_{32}	mean	5.82597×10^{-7}	5.82776×10^{-7}	1.000029512	5.48981×10^{-7}	6.94835×10^{-7}	0.001593752	6.94835×10^{-7}	7.97848×10^{-7}
	fbest	−0.999999871	−0.999999905	−0.000231196	−0.999999914	−0.999999939	−0.999794565	−0.999999939	−0.999999441
	fworst	−0.999999058	−0.999999002	0.000247222	−0.999999013	−0.999999007	−0.996308738	−0.999999007	−0.999999
	std	3.01624×10^{-7}	3.24274×10^{-7}	0.000123742	3.24186×10^{-7}	3.43269×10^{-7}	0.0010056	3.43269×10^{-7}	2.06239×10^{-7}
	Rank	2.5	4	7.75	2.25	3.75	7.25	3.75	4.75
F_{33}	mean	5.49501×10^{-7}	7.70507×10^{-7}	1.000129343	4.54042×10^{-7}	5.33243×10^{-7}	0.001035064	5.33243×10^{-7}	5.37615×10^{-7}
	fbest	−0.999999773	−0.999999818	−0.000108694	−0.999999815	−0.999999834	−0.999757772	−0.999999834	−0.99999972
	fworst	−0.999999158	−0.999999014	0.000508045	−0.999999233	−0.999999115	−0.997105342	−0.999999115	−0.999999161
	std	2.21725×10^{-7}	2.50348×10^{-7}	0.000200878	2.11524×10^{-7}	2.4772×10^{-7}	0.000670503	2.4772×10^{-7}	2.33797×10^{-7}
	Rank	3.75	5.25	7.75	1.75	3.25	7.25	3.25	3.75
F_{34}	mean	0.003330807	0.001432885	1.20241×10^{-6}	0.066647095	0.050196119	2.013375127	0.050196119	0.064466293
	fbest	22.60020311	22.60016672	22.60016256	22.6578775	22.60389669	23.39462771	22.60389669	22.61374764
	fworst	22.61964753	22.60820978	22.60016403	22.67814948	22.79306043	25.80273822	22.79306043	22.7163607
	std	0.005847667	0.002435636	5.68901×10^{-7}	0.007267971	0.056264276	0.618335792	0.056264276	0.046766844
	Rank	3	2	1	5.5	5.5	8	5.5	5.5
F_{35}	mean	4.21367×10^{-5}	7.34346×10^{-5}	43.99842714	0.047706852	0.037749041	0.613282357	0.037749041	0.042552045
	fbest	−43.99999912	−43.99999921	−0.018998014	−43.99461474	−43.99806943	−43.66363491	−43.99806943	−43.99907598
	fworst	−43.99982255	−43.99966851	0.015361582	−43.87743314	−43.80357683	−42.89296325	−43.80357683	−43.90956489
	std	6.05429×10^{-5}	0.000117837	0.010641072	0.048591681	0.058428202	0.206726348	0.058428202	0.039630223
	Rank	1.25	1.75	6.75	5.25	5	7.25	5	3.75
F_{36}	mean	311,239,828.3	311,710,015.1	414.3068078	311,984,943.1	31,654,254.13	312,465,024.1	312,475,069.1	4.16517×10^{-7}
	fbest	−310,023,247.3	−310025090	−7.3295459	−310224910	−8,747,144.123	−312275090	−312474910	−2.919700407
	fworst	−312475090	−312475090	1903.790949	−312474910	−52,146,504.66	−312475090	−312475090	−2.919699536
	std	1,202,330.763	1,077,072.595	648.0859878	987,687.2514	16,062,787.13	44,706.61521	56.92099788	6.23569×10^{-7}
	Rank	4.625	4.375	5	4.75	5.75	3.875	3.375	4.25
F_{37}	mean	1.681421109	1.517910026	1.143559538	2.638399549	2.604558257	2.947513673	2.604558257	5.224224016
	fbest	0.565136907	0.561530716	0.784822016	1.521393183	0.642094331	1.919580303	0.642094331	4.402027013
	fworst	4.042335298	4.090233326	2.56706509	3.650584622	3.949114149	5.237287635	3.949114149	6.74294099
	std	1.407400812	1.673072793	0.548818767	0.938122555	1.142671841	0.811655281	1.142671841	1.224561856
	Rank	4.25	4.25	2	4.25	4	5.75	4	7.5
F_{38}	mean	18.25034093	18.19329333	6.22604×10^{15}	18.54753351	18.90925075	1.19859×10^{27}	18.90925075	2.94623×10^{44}
	fbest	27.87192662	27.69641836	7.49467×10^{12}	28.31762556	28.47821866	28.33766435	28.47821866	2.06201×10^{37}
	fworst	28.22616083	28.12083393	3.64735×10^{16}	28.3559473	28.99917104	2.05451×10^{28}	28.99917104	1.28717×10^{45}
	std	0.105575391	0.143972638	1.24281×10^{16}	0.015535719	0.130389432	4.59998×10^{27}	0.130389432	5.6066×10^{44}
	Rank	2	2	6.25	2.5	4.5	6.25	4.5	8

Table 7. *Cont.*

Function		ARO	CARO	MPA	HGS	CSA	ESOA	SMA	PSO
F_{39}	mean	3.19545×10^{-7}	2.83303×10^{-7}	5.56757×10^{-7}	1.50923×10^{-7}	3.25992×10^{-7}	2.175×10^{-7}	3.25992×10^{-7}	6.1177×10^{-7}
	fbest	7.81265×10^{-10}	8.03146×10^{-10}	1.70997×10^{-8}	0	1.9303×10^{-10}	1.35716×10^{-11}	1.9303×10^{-10}	1.89636×10^{-7}
	fworst	8.79728×10^{-7}	8.3641×10^{-7}	9.9833×10^{-7}	3.97501×10^{-7}	8.58568×10^{-7}	7.7184×10^{-7}	8.58568×10^{-7}	9.99195×10^{-7}
	std	2.98302×10^{-7}	3.21788×10^{-7}	4.33664×10^{-7}	2.02161×10^{-7}	3.17251×10^{-7}	2.50442×10^{-7}	3.17251×10^{-7}	3.3814×10^{-7}
	Rank	4.5	4.5	7.25	1	4.5	2	4.5	7.75
F_{40}	mean	4.2609×10^{-7}	1.80027×10^{-7}	7.69959×10^{-7}	8.782×10^{-8}	4.17723×10^{-7}	2.38552×10^{-7}	4.17723×10^{-7}	1.89386×10^{-7}
	fbest	1.34452×10^{-9}	5.66582×10^{-9}	7.59801×10^{-8}	0	1.8459×10^{-8}	5.87142×10^{-10}	1.8459×10^{-8}	5.91941×10^{-10}
	fworst	9.73788×10^{-7}	7.01077×10^{-7}	9.97939×10^{-7}	1.63959×10^{-7}	9.20547×10^{-7}	9.77222×10^{-7}	9.20547×10^{-7}	4.35852×10^{-7}
	std	3.44292×10^{-7}	2.03153×10^{-7}	3.1639×10^{-7}	7.84266×10^{-8}	3.43689×10^{-7}	2.66474×10^{-7}	3.43689×10^{-7}	1.71844×10^{-7}
	Rank	6.25	3.25	7.25	1	5.75	4.25	5.75	2.5
F_{41}	mean	1.16167×10^{-6}	5.10412×10^{-7}	5.90589×10^{-7}	1.52176×10^{-5}	2.7141×10^{-7}	1.31446×10^{-5}	2.7141×10^{-7}	6.5831×10^{-7}
	fbest	3.28339×10^{-7}	5.11802×10^{-8}	7.29828×10^{-8}	1.57578×10^{-7}	5.31558×10^{-9}	2.11677×10^{-7}	5.31558×10^{-9}	4.83157×10^{-8}
	fworst	5.6177×10^{-6}	1.03204×10^{-6}	1.00936×10^{-6}	4.20225×10^{-5}	9.76255×10^{-7}	0.000108385	9.76255×10^{-7}	9.82432×10^{-7}
	std	1.58324×10^{-6}	3.93803×10^{-7}	3.30232×10^{-7}	1.71148×10^{-5}	3.42339×10^{-7}	2.34033×10^{-5}	3.42339×10^{-7}	3.89007×10^{-7}
	Rank	6.5	4.25	3.5	7	1.75	7.5	1.75	3.75
F_{42}	mean	0.344381004	0.275970153	0.403020486	0.393685343	0.24430027	21.34319354	0.24430027	1.694505182
	fbest	247.0147548	246.9137329	246.5973326	246.9991836	246.9809291	259.8272902	246.9809291	247.6714151
	fworst	247.8045785	246.6060455	246.5968829	248.2910591	247.7905391	280.2332048	247.7905391	249.8710681
	std	0.367604136	0.104310764	0.000137456	0.520070611	0.339793835	6.812484652	0.339793835	0.977131424
	Rank	5	2.25	2.25	5.5	3	8	3	7
F_{43}	mean	3.90135×10^{-5}	5.64866×10^{-6}	8.17287×10^{-7}	0.001467884	0.000271811	0.000559692	0.000271811	4.90213×10^{-7}
	fbest	2.76287×10^{-8}	6.50721×10^{-9}	1.7129×10^{-8}	1.8144×10^{-8}	0	1.2879×10^{-6}	0	2.0196×10^{-7}
	fworst	0.000100936	1.87863×10^{-5}	9.95462×10^{-7}	0.005904487	0.001110857	0.002236417	0.001110857	9.99283×10^{-7}
	std	3.43599×10^{-5}	6.58166×10^{-6}	2.98776×10^{-7}	0.00250822	0.000446041	0.000548994	0.000446041	3.04132×10^{-7}
	Rank	4.5	3	2	7.25	4.5	7.25	4.5	3
F_{44}	mean	6.81637×10^{-7}	6.06085×10^{-7}	8.75151×10^{-7}	7.34891×10^{-7}	6.15326×10^{-5}	0.020132817	6.15326×10^{-5}	5.29415×10^{-7}
	fbest	2.63416×10^{-7}	2.28547×10^{-7}	6.13749×10^{-7}	5.55512×10^{-7}	3.26941×10^{-6}	0.002890538	3.26941×10^{-6}	2.94074×10^{-7}
	fworst	9.88172×10^{-7}	9.47416×10^{-7}	9.89203×10^{-7}	9.1327×10^{-7}	0.00013953	0.042531966	0.00013953	8.01766×10^{-7}
	std	2.40257×10^{-7}	2.64824×10^{-7}	1.22757×10^{-7}	1.27972×10^{-7}	4.57757×10^{-5}	0.010513879	4.57757×10^{-5}	1.80812×10^{-7}
	Rank	3.25	2.75	4	3	6.5	8	6.5	2
F_{45}	mean	5.59936×10^{-7}	7.13656×10^{-7}	9.58392×10^{-7}	3.30979×10^{-7}	4.89775×10^{-7}	4.93988×10^{-7}	4.89775×10^{-7}	0.058013748
	fbest	1.77596×10^{-7}	3.74503×10^{-7}	8.51176×10^{-7}	0	1.87521×10^{-7}	6.51716×10^{-8}	1.87521×10^{-7}	0.00966428
	fworst	9.64326×10^{-7}	9.47956×10^{-7}	1.00308×10^{-6}	6.84449×10^{-7}	7.04142×10^{-7}	9.96457×10^{-7}	7.04142×10^{-7}	0.112866004
	std	2.67222×10^{-7}	1.93094×10^{-7}	4.8023×10^{-8}	3.06706×10^{-7}	1.89948×10^{-7}	3.01662×10^{-7}	1.89948×10^{-7}	0.047436642
	Rank	4.5	5	5.5	2.5	3	4.5	3	8
F_{46}	mean	5.16551×10^{-7}	3.49904×10^{-7}	6.65618×10^{-7}	2.2216×10^{-7}	3.35613×10^{-7}	1.8994×10^{-7}	3.35613×10^{-7}	2.66498×10^{-7}
	fbest	3.33231×10^{-8}	2.05861×10^{-8}	2.54649×10^{-7}	0	3.38636×10^{-9}	3.55392×10^{-10}	3.38636×10^{-9}	6.09846×10^{-8}
	fworst	9.50431×10^{-7}	8.02878×10^{-7}	1.00113×10^{-6}	7.03967×10^{-7}	7.82565×10^{-7}	7.35399×10^{-7}	7.82565×10^{-7}	6.26746×10^{-7}
	std	3.41788×10^{-7}	2.60125×10^{-7}	2.69665×10^{-7}	3.1866×10^{-7}	2.31254×10^{-7}	1.78967×10^{-7}	2.31254×10^{-7}	2.20094×10^{-7}
	Rank	7	5.5	7.5	3	4	1.75	4	3.25

Table 8. Statistical results of the compared chaotic algorithms for unconstrained min–max optimization problems.

Function		CARO	CMPA	CHGS	CCSA	CESOA	CSMA	CPSO
F_1	mean	5.38751×10^{-7}	7.14283×10^{-7}	0.00014583	0.005649003	0.002798702	3.69435×10^{-5}	7.41892×10^{-5}
	fbest	1.952225099	1.952224899	1.952224808	1.954095308	1.953080935	1.952225066	1.952232943
	fworst	1.95222534	1.952225469	1.953530969	1.966467446	1.957355916	1.952448857	1.952486434
	std	1.01954×10^{-7}	1.91375×10^{-7}	0.000305008	0.003944794	0.001548225	6.82603×10^{-5}	0.000106723
	Rank	1.75	2	4	7	6	3	4.25
F_2	mean	4.21184×10^{-7}	7.07151×10^{-7}	9.98728×10^{-8}	4.61258×10^{-7}	4.53225×10^{-7}	4.3136×10^{-7}	7.35835×10^{-7}
	fbest	4.60148×10^{-8}	3.00346×10^{-8}	0	8.92756×10^{-8}	2.70018×10^{-7}	1.04511×10^{-7}	5.48846×10^{-7}
	fworst	8.79379×10^{-7}	9.98012×10^{-7}	7.85057×10^{-7}	8.62543×10^{-7}	7.79579×10^{-7}	8.67232×10^{-7}	9.73994×10^{-7}
	std	2.66477×10^{-7}	3.44201×10^{-7}	2.47386×10^{-7}	2.92343×10^{-7}	2.04009×10^{-7}	2.6957×10^{-7}	1.71185×10^{-7}
	Rank	3.5	5.5	1.75	4.5	3.25	4.25	5.25
F_3	mean	2.96263×10^{-7}	7.3808×10^{-7}	2.84325×10^{-7}	2.09132×10^{-7}	2.81391×10^{-7}	3.10262×10^{-7}	7.8592×10^{-7}
	fbest	2.692×10^{-9}	1.84288×10^{-7}	0	6.24957×10^{-9}	5.15856×10^{-8}	9.36534×10^{-9}	4.51295×10^{-7}
	fworst	6.88592×10^{-7}	1.00077×10^{-6}	8.61155×10^{-7}	5.31538×10^{-7}	6.8472×10^{-7}	8.37202×10^{-7}	9.15766×10^{-7}
	std	2.43728×10^{-7}	3.14387×10^{-7}	2.7601×10^{-7}	1.91117×10^{-7}	2.61045×10^{-7}	3.41844×10^{-7}	1.95601×10^{-7}
	Rank	3	6.25	3.5	1.5	3.25	5	5.5

Table 8. *Cont.*

Function		CARO	CMPA	CHGS	CCSA	CESOA	CSMA	CPSO
F_4	mean	7.08664×10^{-7}	5.81653×10^{-7}	0.00026437	0.002238101	0.012789142	8.56765×10^{-5}	0.000465268
	fbest	3.599719436	3.599719527	3.599720418	3.600301602	3.602290061	3.599720576	3.599723856
	fworst	3.59972029	3.599720234	3.600890995	3.604426612	3.626374584	3.600028952	3.601528338
	std	2.4103×10^{-7}	2.36042×10^{-7}	0.000354421	0.001416606	0.010057317	0.000125651	0.000759196
	Rank	1.75	1.25	3.75	6	7	3.25	5
F_5	mean	6.11881×10^{-5}	43.9987266	0.723446874	1.009613359	0.275321921	0.039488444	10.98895281
	fbest	−43.99999919	−0.027464345	−43.99818002	−43.58648329	−43.79489323	−43.99783521	−43.99929257
	fworst	−43.99966169	0.008391071	−40.88914458	−42.05145559	−43.66174489	−43.90219905	10.72783696
	std	0.00010622	0.01214846	1.103804358	0.495257818	0.053956812	0.03379996	24.45088754
	Rank	1	5.5	4.5	5	3.75	2.75	5.5
F_6	errors	3.34697×10^{-5}	44.00041366	1.040475632	1.201437041	0.300665315	0.018438807	0.042579072
	fbest	−43.9999939	−0.042089004	−43.98973734	−43.59391219	−43.84299799	−43.99828116	−43.99740751
	fworst	−43.99987486	0.054868868	−40.77008285	−40.36506661	−43.46794567	−43.9550762	−43.88975082
	std	3.55632×10^{-5}	0.023930331	1.338941631	0.964676472	0.154743894	0.015995196	0.050989076
	Rank	1	6	5.25	6	4.5	2	3.25
F_7	mean	0.000366949	0.001301682	0.022015578	0.002206502	0.001853398	0.00157594	0.001356416
	fbest	0.004202666	0.004378779	0.00449053	0.006049129	0.005192437	0.004235513	0.004512023
	fworst	0.006189797	0.006074538	0.058584114	0.006834181	0.006809059	0.007051301	0.006108759
	std	0.000644253	0.000686579	0.02065215	0.000271013	0.000709622	0.00085501	0.000706946
	Rank	1.75	2.25	6.25	4.75	5	4.5	3.5
F_8	mean	0.027140917	0.023629799	0.828885069	0.023649138	0.052027981	0.621547624	0.046182762
	fbest	0.074443363	0.074443396	0.074453685	0.074446895	0.102773215	0.193535029	0.074443363
	fworst	0.10263802	0.074455821	3.277071077	0.074490916	0.102921348	1.408419115	0.10263802
	std	0.008942324	3.70591×10^{-6}	0.930228193	1.5923×10^{-5}	5.64019×10^{-5}	0.400607168	0.012609034
	Rank	3	1.5	6.5	2.5	4.75	6.25	3.5
F_9	mean	0.000876941	0.000496538	0.004958025	0.002340727	0.001336771	0.003867294	0.011042038
	fbest	0.008308769	0.008085209	0.007984701	0.00879402	0.008891923	0.008365966	0.008084259
	fworst	0.006368778	0.007102396	0.023126362	0.012691622	0.009853089	0.022634829	0.032978979
	std	0.000929729	0.000440792	0.004879288	0.001293908	0.000441087	0.004483383	0.01063425
	Rank	2.5	1.75	4.75	4.5	3.75	5	5.75
F_{10}	mean	1.70716×10^{-5}	5.94021×10^{-7}	23.88299471	7.748322334	5.821704888	0.036767322	0.067649095
	fbest	115.70644	115.70644	115.7315772	119.3715856	118.3841492	115.7070119	115.7219403
	fworst	115.7065909	115.706441	188.1576578	129.5353189	129.3452568	115.7976902	115.8211833
	std	4.71155×10^{-5}	4.23456×10^{-7}	30.6474368	3.101678423	4.511421387	0.02921009	0.040317492
	Rank	1.875	1.125	6.5	6	5.5	3	4
F_{11}	mean	0.001072313	0.000583758	0.022673024	0.029401473	0.024294557	0.015191323	0.295758047
	fbest	0.002801439	0.002793727	0.004866354	0.006224864	0.021496618	0.002992509	0.00899142
	fworst	0.004960521	0.004258267	0.139422368	0.068196684	0.032359633	0.079082487	0.836230379
	std	0.000848247	0.000452432	0.029200358	0.019204716	0.004806382	0.022709036	0.334300107
	Rank	2	1	5	4.75	4.5	4	6.75
F_{12}	mean	0.013178559	0.011148483	0.085512165	0.082796006	0.05211516	0.04496574	0.093133472
	fbest	0.003409187	0.002845389	0.014529718	0.060600424	0.035858108	0.008069687	0.088149228
	fworst	0.024343092	0.024750703	0.294323316	0.108108709	0.066743185	0.103622614	0.112594014
	std	0.005730991	0.006177717	0.066035212	0.017179291	0.015297903	0.034093585	0.010596333
	Rank	1.5	1.5	6	5.25	4	4	5.75
F_{13}	mean	0.633565012	0.648999649	2.244269155	0.714344783	0.715353578	0.972842471	0.703007413
	fbest	0.587307423	0.588446272	0.721344764	0.705669572	0.672610432	0.678302447	0.595221119
	fworst	0.707279799	0.705516102	13.7251546	0.74867196	0.741696098	1.749353879	0.747809432
	std	0.061262999	0.055841667	3.039735533	0.012912105	0.027186279	0.361853243	0.063322768
	Rank	2	2	7	4	3.5	5.75	3.75
F_{14}	mean	0.034492406	0.034164522	0.141119357	0.043354092	0.043991346	0.035757107	0.187015814
	fbest	0.034269076	0.034164454	0.014856903	0.038664402	0.0401041	0.035255522	0.035804469
	fworst	0.034787896	0.03466126	0.769172967	0.049540134	0.047481949	0.037218048	0.779130696
	std	0.000174903	0.00013559	0.176485729	0.003511265	0.00346276	0.000588845	0.330944093
	Rank	2.25	1.25	4.75	5	5	3.25	6.5

Table 8. *Cont.*

Function		CARO	CMPA	CHGS	CCSA	CESOA	CSMA	CPSO
F_{15}	mean	0.193309964	0.192477763	0.265837295	0.36339102	0.014602557	0.136925702	0.612090153
	fbest	0.029711643	0.01833368	0.023867629	0.208025062	0.025589039	0.023085935	0.579232516
	fworst	0.641302496	0.641348377	0.739212217	0.814022224	0.053975488	0.731815646	0.694741518
	std	0.303284193	0.303388762	0.315452973	0.17713241	0.011141499	0.282875763	0.051420646
	Rank	4	3.25	5.25	5.5	1.75	3.25	5
F_{16}	mean	0.119490784	0.106553778	0.318992222	0.454434765	0.120644735	0.260561798	0.367226214
	fbest	0.038156093	0.074927343	0.074316171	0.257367287	0.093011891	0.052599824	0.300447029
	fworst	0.517618099	0.270077938	0.455195344	0.653393123	0.239636455	0.432644833	0.554617822
	std	0.14618043	0.051251482	0.123607675	0.119067535	0.053999427	0.136612101	0.126734707
	Rank	3.75	2	4	5.75	2.75	3.75	6
F_{17}	mean	0.006047857	0.00605994	0.006041734	0.005863621	0.006057043	0.006050718	5.60101×10^{-7}
	fbest	0.000147824	0.0001451	0.000228241	0.000476521	0.398	0.000149727	0.006185521
	fworst	0.000114827	0.000115611	0.000116372	0.00018763	0.000117886	0.000115327	0.006184443
	std	1.40979×10^{-5}	1.21148×10^{-5}	2.3362×10^{-5}	9.99269×10^{-5}	1.43603×10^{-5}	1.63058×10^{-5}	6.68041×10^{-7}
	Rank	2.5	3.25	4.25	5	5.5	3.75	3.75
F_{18}	mean	0.260081304	0.00086906	3.260897562	15.10546115	7.071571488	0.249887758	1.633530323
	fbest	680.64214	680.6303727	681.6498183	689.8876404	685.5582458	680.7136705	680.8659307
	fworst	681.2467138	680.6321409	692.2780644	702.0596913	692.9732558	681.0654712	684.9808858
	std	0.236716774	0.000516235	2.631972375	3.82324189	2.989872372	0.117490386	1.609756171
	Rank	2.75	1	5	7	6	2.25	4
F_{19}	mean	1.950053824	0.542249781	20.23085655	86.46552851	40.08636741	2.488919481	4.961931188
	fbest	24.61014884	24.62081357	31.91599252	61.9551592	57.47457172	24.5855617	26.80375224
	fworst	29.99977381	25.24380259	74.22473554	165.8240901	68.21112568	33.44901381	32.96361346
	std	2.106921119	0.20342394	11.98229152	4.202877169	34.85104351	4.101987464	2.389477826
	Rank	2	1.5	5.5	6.5	6	3	3.5
F_{20}	mean	46.51947898	45.37035571	96.6204909	1468.212967	214.1504802	41.34856379	91.5158585
	fbest	93.15476406	138.8184348	141.3350537	934.8617857	272.4836011	100.8808826	133.7282793
	fworst	83.20077047	446.1472412	455.3378816	2559.989204	458.7265957	84.71707608	350.4200741
	std	3.362524602	103.4854883	96.81403257	547.2640359	74.278359	6.051923449	84.15404052
	Rank	1.5	4	5	7	5.25	1.75	3.5
F_{21}	mean	9.03114×10^{-5}	44.00005745	0.150876129	0.41742391	0.425319587	0.015846543	0.045389712
	fbest	−43.99999871	−0.006756419	−43.99949083	−43.9157033	−43.76040654	−43.99909667	−43.99445824
	fworst	−43.99960237	0.008868704	−43.50143059	−42.96718915	−43.42052845	−43.94049744	−43.88649032
	std	0.000130695	0.004861794	0.135085282	0.247839017	0.149382988	0.019767284	0.046475682
	Rank	1	5.75	3.75	5.75	5.75	2.5	3.5
F_{22}	mean	6.64×10^{-7}	6.97525×10^{-7}	7.70227×10^{-7}	0.011551652	0.009235555	1.5821×10^{-5}	8.42845×10^{-7}
	fbest	2.000000202	2.000000163	2.000000204	2.00480827	2.003938197	2.00000141	2.000000513
	fworst	2.000000986	2.000000978	2.000001734	2.01834648	2.016000357	2.000069456	2.000001104
	std	2.68658×10^{-7}	2.42496×10^{-7}	3.27536×10^{-7}	0.005189873	0.004366704	1.99123×10^{-5}	2.11743×10^{-7}
	Rank	2	1.5	3.5	7	6	5	3
F_{23}	mean	257.3779766	2.80891×10^{20}	65535	257.2947968	257.29042	256.5330153	58.91590467
	fbest	3.705642789	591,770.7624	276.7384484	3.938646547	3.84565514	4.726467484	261.0825799
	fworst	3.703860752	2.18419×10^{21}	65535	3.729880599	3.761265462	4.098143983	555.6621009
	std	0.000652002	6.83118×10^{20}	46,144.56	0.058239558	0.032711248	0.212182586	131.7399668
	Rank	2	7	6	3	2.5	3.5	4
F_{24}	mean	0.452836465	0.435819038	0.484261213	0.416926452	0.501044608	0.472403389	6.307357298
	fbest	0.407745341	0.282805885	0.422661578	0.370087408	0.493716377	0.457042632	3.140838783
	fworst	0.47510754	0.629746901	0.637938449	0.462300307	0.515892349	0.496457679	10.11379523
	std	0.02086195	0.139200642	0.052780563	0.028689999	0.008873783	0.012578066	2.638017308
	Rank	2.75	3.5	5	2	4.25	3.5	7
F_{25}	mean	0.226574166	0.267078093	5.46985×10^{97}	0.26374434	0.237679522	0.630637508	0.262079639
	fbest	0.240472725	0.308045096	3.036096143	0.310461493	0.218702674	0.667343819	0.309672695
	fworst	0.309672608	0.329669618	1.0808×10^{99}	0.31559779	0.310281641	0.688193957	0.310495052
	std	0.018427993	0.007137032	2.41532×10^{98}	0.001457676	0.0380889	0.008102636	0.000394696
	Rank	2.25	4	7	3.75	2.75	5.5	2.75

Table 9. Statistical results of the compared chaotic algorithms on non-smooth unconstrained min–max optimization problems.

Function		CARO	CMPA	CHGS	CCSA	CESOA	CSMA	CPSO
F_{26}	mean	6.47088×10^{-7}	6.23911×10^{-7}	0.001934836	0.013353594	0.020828376	0.000242566	0.000287633
	fbest	7.200000164	7.20000028	7.200013672	7.203348616	7.201970083	7.20000328	7.200014551
	fworst	7.20000097	7.200000964	7.206869938	7.22898782	7.247407882	7.200896444	7.201178359
	std	2.63655×10^{-7}	2.52335×10^{-7}	0.00209114	0.007827795	0.017060392	0.000313359	0.00050095
	Rank	1.75	1.25	4.75	6.25	6.75	3	4.25
F_{27}	mean	5.54776×10^{-7}	7.62391×10^{-7}	2.15265×10^{-7}	4.24982×10^{-7}	3.87476×10^{-7}	3.32862×10^{-7}	6.26946×10^{-7}
	fbest	6.98972×10^{-8}	1.08588×10^{-7}	0	5.7641×10^{-8}	8.63694×10^{-8}	4.41895×10^{-8}	3.8369×10^{-7}
	fworst	9.88921×10^{-7}	9.98728×10^{-7}	9.6025×10^{-7}	9.87077×10^{-7}	7.1248×10^{-7}	8.79398×10^{-7}	9.83671×10^{-7}
	std	3.42353×10^{-7}	3.13849×10^{-7}	3.51214×10^{-7}	2.70439×10^{-7}	2.96562×10^{-7}	2.81892×10^{-7}	2.80734×10^{-7}
	Rank	5.25	6.25	3	3.25	3.25	2.25	4.75
F_{28}	mean	6.1464×10^{-7}	6.46256×10^{-7}	0.000176153	0.004340216	0.004529972	1.98237×10^{-5}	2.85269×10^{-5}
	fbest	1.952224516	1.952224871	1.952225967	1.952384589	1.953750812	1.952225019	1.952225485
	fworst	1.952225419	1.952225357	1.953943524	1.965041974	1.958775706	1.952270506	1.952298284
	std	2.98679×10^{-7}	1.41134×10^{-7}	0.000385209	0.003888722	0.002195026	1.89856×10^{-5}	3.32384×10^{-5}
	Rank	1.5	1.5	5	6.5	6.5	3	4
F_{29}	mean	6.91413×10^{-7}	5.57204×10^{-7}	9.80587×10^{-5}	0.005071849	0.006384286	0.000138949	0.000372607
	fbest	2.000000343	2.000000056	2.000000567	2.00366318	2.001609606	2.000015677	2.000005201
	fworst	2.000000908	2.000000939	2.001021653	2.007283928	2.010903981	2.000542849	2.001484421
	std	1.83203×10^{-7}	3.11106×10^{-7}	0.000225164	0.001387831	0.004127302	0.000172965	0.000628802
	Rank	1.5	1.5	3.5	6.25	6.75	3.75	4.75
F_{30}	mean	4.89659×10^{-7}	2.999996663	8.52883×10^{-7}	0.014504372	0.054385296	7.41946×10^{-7}	5.25516×10^{-7}
	fbest	−2.999999671	−0.00145289	−2.999999737	−2.996019411	−2.971296931	−2.999999968	−2.999999719
	fworst	−2.99999923	0.001701064	−2.999999	−2.967146891	−2.918601976	−2.999998038	−2.99999912
	std	1.43259×10^{-7}	0.000873473	1.65061×10^{-7}	0.010318818	0.0248602	6.25078×10^{-7}	2.19364×10^{-7}
	Rank	1.75	6.5	2.75	5.25	6.25	3	2.5
F_{31}	mean	7.40697×10^{-7}	1.41424644	3.32317×10^{-5}	0.000947636	0.001185311	1.55038×10^{-5}	1.32119×10^{-5}
	fbest	−1.414213195	−0.000498014	−1.414213192	−1.414070255	−1.413901679	−1.414213346	−1.41421307
	fworst	−1.414212626	0.000708297	−1.413907771	−1.412351476	−1.412292175	−1.414163159	−1.414167284
	std	1.91315×10^{-7}	0.000291703	7.28001×10^{-5}	0.000580011	0.000582965	1.38154×10^{-5}	1.90849×10^{-5}
	Rank	1.25	6.5	3.75	5.25	6.25	2.25	2.75
F_{32}	mean	5.82776×10^{-7}	0.999966079	6.01752×10^{-7}	0.001337265	0.002038009	5.64455×10^{-7}	6.89122×10^{-7}
	fbest	−0.999999905	−0.000188457	−0.999999828	−0.999699987	−0.999225221	−0.999999979	−0.999999729
	fworst	−0.999999002	0.000130463	−0.999999074	−0.996903377	−0.994917727	−0.999999089	−0.999999002
	std	3.24274×10^{-7}	9.69537×10^{-5}	2.48517×10^{-7}	0.001014753	0.001761718	3.17077×10^{-7}	3.2836×10^{-7}
	Rank	2.62	6.5	2.25	5.25	6.25	1.25	3.88
F_{33}	mean	7.70507×10^{-7}	1.000110279	6.39149×10^{-7}	0.000568466	0.001603569	4.475×10^{-7}	7.37197×10^{-7}
	fbest	−0.999999818	-9.28219×10^{-5}	−0.999999973	−0.99984269	−0.999426898	−0.999999894	−0.999999672
	fworst	−0.999999014	0.000439811	−0.999999005	−0.998954999	−0.996905428	−0.999999203	−0.999999047
	std	2.50348×10^{-7}	0.000150816	2.96961×10^{-7}	0.000227815	0.000937756	2.52226×10^{-7}	2.53938×10^{-7}
	Rank	2.75	6.5	2.75	5.25	6.25	1.5	3
F_{34}	mean	0.001432885	2.6884×10^{-6}	0.140513929	0.613484974	1.150166768	0.037955718	0.044683793
	fbest	22.60016672	22.60016294	22.60572684	22.90469351	23.21537566	22.60831318	22.60961181
	fworst	22.60820978	22.60016825	22.94116498	23.95208561	23.95859998	22.68706991	22.69628715
	std	0.002435636	1.5884×10^{-6}	0.086520015	0.339750006	0.303561282	0.025510415	0.034932318
	Rank	2	1	4.5	6.25	6.75	3.25	4.25
F_{35}	mean	7.34346×10^{-5}	43.99707748	0.176848532	0.307466997	0.323129862	0.012339789	0.039445656
	fbest	−43.99999921	−0.031085637	−43.98829423	−43.87992372	−43.78079017	−43.99808195	−43.98240169
	fworst	−43.99966851	0.014268312	−43.51179777	−43.41727907	−43.5489826	−43.96564957	−43.91788667
	std	0.000117837	0.013090079	0.132436485	0.14581702	0.09335846	0.010450427	0.025597398
	Rank	1	6	4.5	5.75	5.25	2	3.5
F_{36}	mean	311,710,015.1	244.7918374	311,494,934.1	312,455,087.1	312,475,087.1	312,475,087.1	3.69689×10^{-7}
	fbest	−310025090	44.38210432	−310024910	−312275090	−312475090	−312475090	−2.919700344
	fworst	−312475090	792.6594792	−312475090	−312475090	−312475090	−312475090	−2.919699726
	std	1,077,072.595	352.2861283	1,231,434.262	63,245.5532	0	0	4.28523×10^{-7}
	Rank	4.25	5	4.5	4	3.12	3.12	4

Table 9. *Cont.*

Function		CARO	CMPA	CHGS	CCSA	CESOA	CSMA	CPSO
F_{37}	mean	1.517910026	1.050321898	2.592931024	4.253931433	1.380571833	3.201080816	4.685452784
	fbest	0.561530716	1.122262675	0.577715832	4.044930163	1.658188686	3.480930621	4.045841032
	fworst	4.090233326	2.170755881	3.806903694	5.642668828	2.292305058	5.053951508	6.558539344
	std	1.673072793	0.296284643	1.136689334	0.54812948	0.249956672	0.461356952	0.970592343
	Rank	3.75	1.75	3.75	5.5	2.25	4.5	6.5
F_{38}	mean	18.19329333	3.05091×10^{18}	18.50764031	19,277,403.22	18.75806319	18.85680045	6.47241×10^{48}
	fbest	27.69641836	8.32775×10^{13}	28.20843365	28.62930994	28.46159327	28.47299172	1.53728×10^{40}
	fworst	28.12083393	2.99645×10^{19}	28.44975585	183,895,487.4	28.68874088	28.78725035	2.82629×10^{49}
	std	0.143972638	9.45677×10^{18}	0.054499359	57,907,075.62	0.097131404	0.089384687	1.23099×10^{49}
	Rank	1.75	6	1.75	5	3	3.5	7
F_{39}	mean	2.83303×10^{-7}	5.22836×10^{-7}	1.46556×10^{-7}	2.36084×10^{-7}	2.45193×10^{-8}	2.51866×10^{-7}	2.25516×10^{-7}
	fbest	8.03146×10^{-10}	1.65617×10^{-7}	0	3.69846×10^{-10}	4.59804×10^{-10}	1.8171×10^{-9}	5.50329×10^{-8}
	fworst	8.3641×10^{-7}	8.58848×10^{-7}	9.81303×10^{-7}	8.41524×10^{-7}	9.36972×10^{-8}	9.98795×10^{-7}	4.26705×10^{-7}
	std	3.21788×10^{-7}	2.54909×10^{-7}	2.78541×10^{-7}	3.08376×10^{-7}	4.02632×10^{-8}	3.16427×10^{-7}	1.69395×10^{-7}
	Rank	5	5.5	3.25	3.75	1.5	5.75	3.25
F_{40}	mean	1.80027×10^{-7}	4.71893×10^{-7}	1.2095×10^{-7}	3.5801×10^{-7}	1.29101×10^{-7}	2.89406×10^{-7}	2.42735×10^{-7}
	fbest	5.66582×10^{-9}	1.2081×10^{-8}	0	2.68079×10^{-9}	1.53874×10^{-8}	8.08425×10^{-10}	2.46291×10^{-8}
	fworst	7.01077×10^{-7}	9.42115×10^{-7}	6.84703×10^{-7}	9.35584×10^{-7}	2.34688×10^{-7}	9.70364×10^{-7}	7.80151×10^{-7}
	std	2.03153×10^{-7}	3.39324×10^{-7}	2.28669×10^{-7}	3.74736×10^{-7}	9.17163×10^{-8}	3.34827×10^{-7}	3.17512×10^{-7}
	Rank	3	6	1.75	5.25	2.5	4.75	4.75
F_{41}	mean	5.10412×10^{-7}	6.56022×10^{-7}	6.52745×10^{-5}	4.06073×10^{-7}	2.74857×10^{-6}	2.74074×10^{-7}	5.99909×10^{-7}
	fbest	5.11802×10^{-8}	9.16828×10^{-8}	0	4.51995×10^{-8}	5.41607×10^{-7}	0	3.85153×10^{-7}
	fworst	1.03204×10^{-6}	9.61697×10^{-7}	0.00028935	9.63601×10^{-7}	1.1081×10^{-5}	9.78759×10^{-7}	7.28576×10^{-7}
	std	3.93803×10^{-7}	2.79597×10^{-7}	8.21579×10^{-5}	3.54756×10^{-7}	4.66161×10^{-6}	3.8039×10^{-7}	1.34814×10^{-7}
	Rank	4.25	3.5	5.62	2.75	6.25	2.62	3
F_{42}	mean	0.275970153	0.402891659	1.793192459	14.10972312	10.24504008	0.204343683	1.396572884
	fbest	246.9137329	246.5977216	246.6742216	252.9489705	253.2395912	246.9227597	247.3900276
	fworst	246.6060455	246.5968935	249.8639597	270.7641538	261.6365347	247.6145404	249.589436
	std	0.104310764	0.000260016	0.768819176	5.856191661	3.195656626	0.262771829	0.917562125
	Rank	2.25	1.5	4	6.75	6.25	2.75	4.5
F_{43}	mean	5.64866×10^{-6}	6.56235×10^{-7}	0.002134379	5.5857×10^{-7}	5.36748×10^{-7}	0.000952038	5.1299×10^{-7}
	fbest	6.50721×10^{-9}	1.97451×10^{-7}	0	5.23315×10^{-8}	1.38316×10^{-7}	0	1.5417×10^{-7}
	fworst	1.87863×10^{-5}	9.91841×10^{-7}	0.009010185	9.51743×10^{-7}	8.84661×10^{-7}	0.003242083	9.31857×10^{-7}
	std	6.58166×10^{-6}	2.87604×10^{-7}	0.003032703	3.22874×10^{-7}	3.63181×10^{-7}	0.00125296	3.48233×10^{-7}
	Rank	4.5	4	5.62	3	3	4.88	3
F_{44}	mean	6.06085×10^{-7}	8.86442×10^{-7}	9.61428×10^{-7}	0.016011738	0.009181877	4.59186×10^{-5}	4.13237×10^{-7}
	fbest	2.28547×10^{-7}	6.42361×10^{-7}	2.89548×10^{-7}	0.006060033	0.002538855	6.13806×10^{-6}	2.33176×10^{-7}
	fworst	9.47416×10^{-7}	9.99026×10^{-7}	5.05836×10^{-6}	0.028861152	0.015404274	0.00015747	5.31904×10^{-7}
	std	2.64824×10^{-7}	1.13672×10^{-7}	9.90092×10^{-7}	0.007410015	0.00527114	4.26656×10^{-5}	1.11049×10^{-7}
	Rank	2	3	3.75	7	6	5	1.25
F_{45}	mean	7.13656×10^{-7}	9.41708×10^{-7}	2.59867×10^{-7}	4.41072×10^{-7}	6.10703×10^{-7}	5.50888×10^{-7}	0.131155634
	fbest	3.74503×10^{-7}	7.3109×10^{-7}	0	5.66149×10^{-8}	2.60821×10^{-7}	8.8986×10^{-8}	0.040369401
	fworst	9.47956×10^{-7}	9.99578×10^{-7}	9.97411×10^{-7}	9.53101×10^{-7}	9.81058×10^{-7}	8.81149×10^{-7}	0.234388914
	std	1.93094×10^{-7}	8.30647×10^{-8}	3.10552×10^{-7}	3.0713×10^{-7}	3.25908×10^{-7}	2.57606×10^{-7}	0.070249757
	Rank	3.5	4.75	3	2.75	4.5	2.5	7
F_{46}	mean	3.49904×10^{-7}	6.24637×10^{-7}	2.09105×10^{-7}	3.073×10^{-7}	5.25883×10^{-7}	2.22306×10^{-7}	5.16971×10^{-7}
	fbest	2.05861×10^{-8}	8.42913×10^{-10}	0	4.8668×10^{-9}	2.29211×10^{-7}	6.3638×10^{-9}	2.97039×10^{-7}
	fworst	8.02878×10^{-7}	9.3962×10^{-7}	8.3656×10^{-7}	8.53049×10^{-7}	7.9883×10^{-7}	8.22996×10^{-7}	8.72759×10^{-7}
	std	2.60125×10^{-7}	3.56919×10^{-7}	2.90306×10^{-7}	2.86256×10^{-7}	2.57354×10^{-7}	2.84232×10^{-7}	2.17614×10^{-7}
	Rank	3.5	5.75	3	4	3.75	3.25	4.75

3.5. Constrained Minimax Optimization Test Problems

The aim of this experiment is to evaluate the performance of the proposed CARO algorithm with constrained min–max test problems. We report the constrained minimax functions F_1–F_{15}. Table 10 contains problem dimensions and optimum function values. For the constrained minimax test problems, the same constraint-handling mechanism was applied to CARO and all compared algorithms to ensure a fair comparison. First, boundary control was performed after each position update. If any candidate solution exceeded the predefined lower or upper bounds, it was corrected according to the corresponding bound. This ensured that all search agents remained within the variable domain.

For the problem constraints, a penalty-based constraint handling strategy was used. Any violation of inequality or equality constraints was added to the objective function as a penalty term. Therefore, infeasible solutions received worse fitness values, while feasible solutions were evaluated according to the original objective function. The results of the proposed algorithm were compared with six algorithms before and after applying the accelerating term. Table 11 displays the experimental results of the proposed algorithm (CARO) and six algorithms including the original artificial rabbits optimization (ARO) over constrained min–max optimization problems. We gave the average errors (mean), the best solution (fbest), the worst solution (fworst), and standard deviation (std) for all algorithms over 30 runs. The Friedman rank test (Rank) is reported in the last row of each test problem in Table 11. The bold black content in the tables indicates the best mean after comparing the six methods. The details of the constrained testing functions that were used in our experiments are as follows:

Table 10. Problem dimensions and optimum function values.

No.	Problem	n	Optimum Value
F_1	MAD1	2	−0.38965952
F_2	MAD2	2	−0.33035714
F_3	MAD4	2	−0.44891079
F_4	MAD5	2	−0.42928061
F_5	PENTAGON	7	−1.85961870
F_6	MAD6	7	0.10183089
F_7	EQUIL	10	0
F_8	Wong 2	10	24.306209
F_9	Wong 3	10	133.78228
F_{10}	MAD8	20	0.50694799
F_{11}	BP filter	20	$0.27607734 \times 10^{-3}$
F_{12}	HS114	9	−1768.8070
F_{13}	Dembo 3	7	1227.2260
F_{14}	Dembo 5	14	7049.2480
F_{15}	Dembo 7	16	174.78699

Table 12 presents the average error from 30 independent runs for the chaotic map structure CR(5) across the compared hybrid algorithms instead of the random parameter in each algorithm. Table 12 shows that there is a considered improvement for some algorithms' behaviour after applying the same randomization structure; however, the proposed algorithm is the most promising algorithms among them. The results of the Friedman rank test indicate that CARO is the most effective algorithm.

Table 11. Statistical results of the compared algorithms for constrained min–max optimization problems.

Function		ARO	CARO	MPA	HGS	CSA	ESOA	SMA	PSO
F_1	mean	6.8158×10^{-7}	6.19464×10^{-7}	0.3897	3.327	0.5556	0.0512	3.81×10^{-6}	6.38965952
	fbest	−0.389659218	−0.3896592	-1.10×10^{-4}	0.4457	0.1659	−0.3516	−0.3897	6
	fworst	−0.389658583	−0.389658578	1.65×10^{-4}	6	0.1659	−0.3189	−0.3897	6
	std	1.79481×10^{-7}	2.23359×10^{-7}	1.01×10^{-4}	2.8434	1.70×10^{-5}	0.0139	2.16×10^{-6}	0
	rank	2	2.5	5.25	7.375	5.75	4.75	2.25	6.125
F_2	mean	5.60752×10^{-7}	7.17308×10^{-7}	0.3304	3.523	0.2234	0.0045	4.02×10^{-5}	6.33035714
	fbest	−0.330357003	−0.33035711	-2.45×10^{-4}	−0.3304	−0.1071	−0.3289	−0.3303	6
	fworst	−0.330356232	−0.330356177	1.39×10^{-4}	6	−0.1065	−0.3197	−0.3303	6
	std	3.09972×10^{-7}	3.23569×10^{-7}	1.48×10^{-4}	2.967	2.11×10^{-4}	0.0037	3.92×10^{-5}	0
	rank	1.75	2.25	6	5.875	5.5	5	3.5	6.125
F_3	mean	2.68812×10^{43}	7.3235×10^{-7}	0.5941	3.3315	2.69×10^{43}	2.69×10^{43}	0.5949	4.054080976
	fbest	-2.68812×10^{43}	−0.448910466	−0.1374011	-2.69×10^{43}	-2.69×10^{43}	−0.2552	3.605170186	
	fworst	-2.68812×10^{43}	−0.448909795	0.6973	3.1701	-2.69×10^{43}	-2.69×10^{43}	−1.2375	3.605170186
	std	5.21961×10^{27}	2.05308×10^{-7}	0.3473	0.3017	0	0	0.7108	0
	rank	4.5	3.5	5	5.75	3.375	3.375	4.75	5.75
F_4	mean	6.75369×10^{-7}	6.04828×10^{-7}	0.4293	0.411	0.0708	0.0037	0.0934	0.410964971
	fbest	−0.429280241	−0.429280603	-4.09×10^{-24}	−0.0183	−0.3586	−0.4286	−0.3558	−0.018315639
	fworst	−0.429279745	−0.429279677	-4.09×10^{-24}	−0.0183	−0.3584	−0.4224	−0.3048	−0.018315639
	std	1.79476×10^{-7}	3.09226×10^{-7}	1.88×10^{-24}	0	5.70×10^{-5}	0.0025	0.0203	3.65712×10^{-18}
	rank	2.25	2.25	6.5	5.5	4.5	4	5.75	5.25
F_5	mean	1.8596187	1.8596187	1.8596	1.8596	1.8596	0.313	0.0291	1.8596187
	fbest	0	0	0	0	0	−1.5837	−1.8587	0
	fworst	0	0	0	0	0	−1.4837	−1.7601	0
	std	0	0	0	0	0	0.0423	0.0426	0
	rank	5.375	5.375	4.625	4.625	4.625	3.25	2.75	5.375
F_6	mean	0.061246613	5.57472×10^{-7}	0.0531	0.0321	0.0321	0.0321	0.0248	0.032116739
	fbest	0.042195218	0.101830727	0.0592	0.1339	0.1339	0.1339	0.1207	0.133947629
	fworst	0.040149616	0.101829912	0.0417	0.1339	0.1339	0.1339	0.1339	0.133947629
	std	0.000650136	6.54428×10^{-7}	0.0075	0	0	0	0.0067	2.92569×10^{-17}
	rank	4	3	4.75	4.375	4.375	4.375	4.625	6.5
F_7	mean	9.787830444	9.787830444	9.7878	9.7878	9.7878	9.7878	9.6753	9.787830444
	fbest	9.787830444	9.787830444	9. 9.7878	9.7878	9.7878	9.6753	9.787830444	
	fworst	9.787830444	9.787830444	9.7878	9.7878	9.7878	9.7878	9.6753	9.787830444
	std	0	0	0	0	0	0	3.20×10^{-15}	0
	rank	6.25	6.25	3.625	3.625	3.625	3.625	2.75	6.25
F_8	mean	1.42110289	1.643180246	1.3303	728.6938	728.2372	26.1135	4.4913	728.693791
	fbest	24.31667283	24.69646964	24.9892	753	752.5434	41.7565	24.4457	753
	fworst	28.39550387	27.9660513	26.8698	753	752.5435	60.8446	39.353	753
	std	1.506413943	1.223091629	0.7598	0	6.79×10^{-6}	7.85	6.1063	0
	rank	3	3.25	2.5	6.125	5.25	5.75	4.25	5.875
F_9	mean	32.56339963	1.83767×10^{-6}	28.9638	767.2717	763.711	171.0299	0.0037	767.27172
	fbest	108.631473	133.7282801	138.3536	901	897.4332	255.6545	133.7272	901
	fworst	95.65198934	133.7282934	197.7002	901	897.4454	385.2074	133.735	901
	std	4.560730911	4.2502×10^{-6}	38.8932	0	0.0053	53.6482	0.0048	0
	rank	3	2.25	4.5	5.875	5.75	5.75	2.75	6.125
F_{10}	mean	0.710778044	0.772661158	0.5609	1.63×10^{3}	0.612	1.3482	0.6638	116.7404008
	fbest	0.790525187	0.768111264	1.0573	9.6045	1.1186	1.5173	1.1443	42.08355303
	fworst	2.048216758	1.634441429	1.0851	3450	1.1196	2.1917	1.1945	270.1711314
	std	0.417046972	0.243556748	0.0105	1.72×10^{3}	4.37×10^{-4}	0.2741	0.0231	77.67965763
	rank	4.25	3.5	1.75	7.75	2.25	5.75	3.5	7.25
F_{11}	mean	2985.025646	1.514171134	1.5809	108.6787	1.53×10^{3}	1.03×10^{29}	1.5197	108.6787126
	fbest	−2984.998919	1.508950881	1.5145	−107.9495	-1.50×10^{3}	-4.20×10^{20}	1.5091	−107.9494757
	fworst	−2984.998919	1.522736834	1.7686	−107.9495	-1.57×10^{3}	-2.52×10^{29}	1.5307	−107.9494757
	std	0	0.005373986	0.1057	1.59×10^{-14}	32.5815	1.09×10^{29}	0.0215	2.99591×10^{-14}
	rank	3	4.25	6.25	3.5	4.75	4.5	5.25	4.5
F_{12}	mean	21,929.94388	27,931.59612	1.06×10^{5}	1.94×10^{19}	9.80×10^{18}	1.94×10^{19}	7.50×10^{4}	1.936×10^{19}
	fbest	0.053096805	5.744874061	443.7118	1.94×10^{19}	3.44×10^{15}	1.94×10^{19}	4.34×10^{4}	1.936×10^{19}
	fworst	40,304.23797	40,235.51617	4.56×10^{5}	1.94×10^{19}	1.91×10^{19}	1.94×10^{19}	1.30×10^{5}	1.936×10^{19}
	std	20,742.72405	18,055.26692	1.97×10^{5}	0	9.50×10^{18}	0	3.56×10^{4}	0
	rank	2.25	2.25	4.5	5.375	5.75	5.375	4	6.5
F_{13}	mean	10,634.66008	10,631.44642	1.06×10^{4}	4.83×10^{5}	7.91×10^{5}	1.19×10^{4}	1.11×10^{4}	804,296.7668
	fbest	11,845.73396	11,848.40236	1.18×10^{4}	2.48×10^{3}	7.92×10^{5}	1.28×10^{4}	1.18×10^{4}	805,523.9928
	fworst	11,889.23667	11,888.89681	1.18×10^{4}	8.06×10^{5}	7.92×10^{5}	1.38×10^{4}	1.40×10^{4}	805,523.9928
	std	13.97960937	11.49060873	0.1061	4.40×10^{5}	111.7257	394.9386	942.763	0
	rank	3.25	3	1.5	5.5	6.25	5.25	5	6.25

Table 11. *Cont.*

Function		ARO	CARO	MPA	HGS	CSA	ESOA	SMA	PSO
F_{14}	mean	238.6789691	415.3398428	100.311	1.35×10^4	1.35×10^4	3.66×10^3	454.0792	13,506.30756
	fbest	7109.290454	7204.467868	7.09×10^3	2.06×10^4	2.05×10^4	1.03×10^4	7.10×10^3	20,555.55556
	fworst	7633.103567	7664.703847	7.23×10^3	2.06×10^4	2.05×10^4	1.11×10^4	8.20×10^3	20,555.55556
	std	171.4456262	148.0226817	55.1803	0	1.94×10^{-5}	269.1566	504.6608	0
	rank	3.25	3.75	1.75	6.125	5.25	5.5	4.5	5.875
F_{15}	mean	40.96139301	35.9834694	7.0656	165.901	250.8601	96.5226	12.7394	253.3351123
	fbest	181.6805665	186.0202521	175.492	318.8294	425.5349	253.4697	176.4632	428.1221023
	fworst	297.098196	244.6687348	191.0398	428.1221	425.8848	287.925	209.0361	428.1221023
	std	36.73343097	17.83682441	8.0221	48.8772	0.1374	12.7296	13.1497	5.99182×10^{-14}
	rank	4.75	4	1.5	6.75	5.5	4.5	2.75	6.25
	Average Rank	3.525	3.425	4	5.608333333	4.833333333	4.716666667	3.891666667	6

Table 12. Statistical results of the compared chaotic algorithms for constrained min–max optimization problems.

Function		CARO	CMPA	CHGS	CCSA	CESOA	CSMA	CPSO
F_1	mean	6.19464×10^{-7}	0.3897	3.779	0.5556	0.0473	4.76×10^{-6}	6.38965952
	fbest	−0.3896592	-1.44×10^{-4}	0.5273	0.1659	−0.3677	−0.3897	6
	fworst	−0.389658578	1.24×10^{-4}	6	0.1659	−0.3026	−0.3897	6
	std	2.23359×10^{-7}	1.12×10^{-4}	2.4857	1.67×10^{-5}	0.0249	2.82×10^{-6}	0
	rank	1.75	4.25	6.375	4.75	3.75	1.75	5.375
F_2	mean	7.17308×10^{-7}	0.3303	3.4207	0.2235	0.0069	4.31×10^{-5}	6.33035714
	fbest	−0.33035711	-1.74×10^{-4}	−0.3303	−0.1071	−0.3295	−0.3304	6
	fworst	−0.330356177	2.40×10^{-4}	6	−0.1065	−0.3206	−0.3302	6
	std	3.23569×10^{-7}	1.68×10^{-4}	2.2966	2.71×10^{-4}	0.0035	5.55×10^{-5}	0
	rank	1.5	5	5.625	4.5	4	2	5.375
F_3	mean	7.3235×10^{-7}	0.5587	5.38×10^{42}	2.69×10^{43}	2.69×10^{43}	0.3184	4.054080976
	fbest	−0.448910466	−0.1499	−0.0058	-2.69×10^{43}	-2.69×10^{43}	−0.3344	3.605170186
	fworst	−0.448909795	0.4471	-2.69×10^{43}	-2.69×10^{43}	-2.69×10^{43}	0.1757	3.605170186
	std	2.05308×10^{-7}	0.2621	1.20×10^{43}	0	0	0.2474	0
	rank	3	5	5	3	3	4	5
F_4	mean	6.04828×10^{-7}	0.4293	0.411	0.0708	0.0032	0.088	0.410964971
	fbest	−0.429280603	-3.97×10^{-30}	−0.0183	−0.3586	−0.4287	−0.3655	−0.018315639
	fworst	−0.429279677	-3.97×10^{-30}	−0.0183	−0.3583	−0.4247	−0.3244	−0.018315639
	std	3.09226×10^{-7}	4.19×10^{-23}	0	9.04×10^{-5}	0.0017	0.0182	3.65712×10^{-18}
	rank	1.75	5.75	4.75	3.75	3	4.5	4.5
F_5	mean	1.8596187	1.8596	1.8596	1.8596	0.2832	0.0301	1.8596187
	fbest	0	0	0	0	−1.6656	−1.8505	0
	fworst	0	0	0	0	−1.4822	−1.8186	0
	std	0	0	0	0	0.0766	0.0132	0
	rank	4.875	4.25	4.25	4.25	3.25	2.25	4.875
F_6	mean	5.57472×10^{-7}	0.0523	0.0321	0.0321	0.0321	0.0256	0.032116739
	fbest	0.101830727	0.0653	0.1339	0.1339	0.1339	0.1144	0.133947629
	fworst	0.101829912	0.0436	0.1339	0.1339	0.1339	0.1339	0.133947629
	std	6.54428×10^{-7}	0.0089	0	0	0	0.0078	2.92569×10^{-17}
	rank	2.5	4	3.875	3.875	3.875	3.875	6
F_7	mean	9.787830444	9.7878	9.7878	9.7878	9.7878	9.6753	9.787830444
	fbest	9.787830444	9.7878	9.7878	9.7878	9.7878	9.6753	9.787830444
	fworst	9.787830444	9.7878	9.7878	9.7878	9.7878	9.6753	9.787830444
	std	0	0	0	0	0	8.88×10^{-16}	0
	rank	5.75	3.5	3.5	3.5	3.5	2.5	5.75
F_8	mean	1.643180246	0.6473	728.6938	728.2372	33.4299	5.0104	728.693791
	fbest	24.69646964	24.6643	753	752.5434	49.1917	28.0629	753
	fworst	27.9660513	25.3003	753	752.5435	61.9575	30.7746	753
	std	1.223091629	0.258	0	7.56×10^{-6}	5.0214	1.3024	0
	rank	2.75	1.75	5.375	4.5	4.75	3.75	5.125

Table 12. *Cont.*

Function		CARO	CMPA	CHGS	CCSA	CESOA	CSMA	CPSO
F_9	mean	1.83767×10^{-6}	20.6945	767.2717	763.7078	176.5539	0.0096	767.27172
	fbest	133.7282801	134.1623	901	897.4315	260.0043	133.7281	901
	fworst	133.7282934	186.773	901	897.4434	356.6973	133.7729	901
	std	4.2502×10^{-6}	29.957	0	0.0051	36.5012	0.0202	0
	rank	1.75	3.75	5.125	4.75	4.75	2.5	5.375
F_{10}	mean	0.772661158	0.585	2.79×10^3	0.6126	1.1907	0.6652	290.1101596
	fbest	0.768111264	1.0704	135.0962	1.1187	1.4133	1.1372	53.53434848
	fworst	1.634441429	1.1077	3450	1.1211	2.1031	1.2015	370.3368243
	std	0.243556748	0.0138	1.48×10^3	9.41×10^{-4}	0.282	0.0274	105.9438534
	rank	3.25	1.5	7	2	5	3.25	6
F_{11}	mean	1.514171134	1.6775	108.6787	1.40×10^3	4.76×10^{28}	1.5358	108.6787126
	fbest	1.508950881	1.5503	−107.9495	-1.37×10^3	-9.26×10^{19}	1.5171	−107.9494757
	fworst	1.522736834	2.0263	−107.9495	-1.42×10^3	-2.28×10^{29}	1.5484	−107.9494757
	std	0.005373986	0.1963	1.59×10^{-14}	17.9063	1.01×10^{29}	0.0365	2.99591×10^{-14}
	rank	3.5	5.5	2.75	4	4.5	4.5	3.75
F_{12}	mean	27,931.59612	1.85×10^5	1.94×10^{19}	2.70×10^{18}	1.94×10^{19}	2.86×10^7	1.936×10^{19}
	fbest	5.744874061	1.13×10^3	1.94×10^{19}	7.15×10^{16}	1.94×10^{19}	4.52×10^4	1.936×10^{19}
	fworst	40,235.51617	7.95×10^5	1.94×10^{19}	7.10×10^{18}	1.94×10^{19}	1.41×10^8	1.936×10^{19}
	std	18,055.26692	3.43×10^5	0	2.95×10^{18}	0	6.30×10^7	0
	rank	1.75	2.75	4.625	4.75	4.625	3.75	5.75
F_{13}	mean	10,631.44642	1.06×10^4	1.62×10^5	7.91×10^5	1.18×10^4	1.13×10^4	804,296.7668
	fbest	11,848.40236	1.18×10^4	2.48×10^3	7.92×10^5	1.30×10^4	1.18×10^4	805,523.9928
	fworst	11,888.89681	1.18×10^4	8.06×10^5	7.93×10^5	1.31×10^4	1.41×10^4	805,523.9928
	std	11.49060873	0.0885	3.59×10^5	164.3454	48.2991	975.447	0
	rank	2.75	1.625	4.75	5.5	4	3.875	5.5
F_{14}	mean	415.3398428	101.5633	1.35×10^4	1.35×10^4	4.86×10^3	480.9608	13,506.30756
	fbest	7204.467868	7.08×10^3	2.06×10^4	2.05×10^4	1.06×10^4	7.10×10^3	20,555.55556
	fworst	7664.703847	7.23×10^3	2.06×10^4	2.05×10^4	1.44×10^4	8.13×10^3	20,555.55556
	std	148.0226817	60.5302	0	8.07×10^{-5}	1.69×10^3	374.1022	0
	rank	3	1.75	5.375	4.5	4.75	3.5	5.125
F_{15}	mean	35.9834694	4.3006	187.7595	250.8515	80.6844	9.9617	253.3351123
	fbest	186.0202521	176.9216	318.8294	425.4479	236.9263	180.0427	428.1221023
	fworst	244.6687348	181.6842	428.1221	425.8082	270.2795	191.0587	428.1221023
	std	17.83682441	2.0392	59.8621	0.159	12.378	4.6278	5.99182×10^{-14}
	rank	3.75	1.5	5.75	4.75	4.25	2.5	5.5
	Average Rank	2.908333333	3.458333333	4.941666667	4.158333333	4.033333333	3.233333333	5.266666667

4. Conclusions

This work introduced a chaotic variant of artificial rabbits optimization (CARO) for solving minimax optimization problems. By embedding five chaotic maps into the randomization components of ARO, CARO strengthens the exploration–exploitation balance and mitigates premature convergence. Comprehensive experiments on 46 benchmark functions (25 unconstrained smooth and 21 nonsmooth) and 15 constrained minimax tests—each evaluated over 30 independent runs—demonstrate that CARO attains the best overall performance. For the unconstrained suite, CARO achieves the top average ranks and reaches the optimal value on the majority of functions; similar gains persist on the nonsmooth set. On constrained problems, CARO outperforms six competitors (including the original ARO) in terms of mean error, with Friedman ranks consistently favoring CARO. A fair-comparison study, where all hybrid baselines adopt the same chaotic CR(5) scheme, confirms that the benefits are not due to the added randomness but due to the proposed design.

In summary, CARO is a reliable and efficient optimizer for both unconstrained and constrained minimax problems, providing improved accuracy and faster convergence relative to state-of-the-art swarm methods. Future work will examine (i) sensitivity and self-adaptation of chaos parameters, (ii) scalability to very high dimensions and large-scale constraints, (iii) hybridization with local solvers for tightened optimality on nonsmooth landscapes, and (iv) theoretical analysis of convergence under chaotic perturbations.

Although the present study focuses mainly on mathematical benchmark optimization problems, the proposed CARO algorithm may also be useful for practical engineering optimization tasks that require robust parameter tuning. Examples include controller parameter optimization, system design optimization, scheduling, and constrained engineering design problems. In such applications, the improved balance between exploration and exploitation may help the algorithm search more effectively in nonlinear and complex solution spaces. Future work may investigate the application of CARO to real-world engineering systems, including robotic control and other physical AI-related optimization problems.

Problem 1 (MAD1 [70]).

$$F(x) = \max_{1\le i\le 2} f_i(x),$$

$$f_1(x) = x_1^2 + x_2^2 + x_1x_2 - 1,$$

$$f_2(x) = \sin x_1,$$

Problem 2 (MAD2 [70]).

$$F(x) = \max_{1\le i\le 2} f_i(x),$$

$$f_3(x) = -\cos x_2,$$

$$c_1(x) = x_1 + x_2 - 0.5 \ge 0$$

$$\bar{x}_1 = 1, \quad \bar{x}_2 = 2.$$

Problem 3 (MAD4 [70]).

$$F(x) = \max_{1\le i\le 3} f_i(x),$$

$$f_1(x) = -\exp(x_1 - x_2),$$

$$f_2(x) = \sinh(1 - x_1) - 1,$$

$$f_3(x) = -\log(x_2),$$

$$c_1(x) = 0.05x_1 - x_2 + 0.5 \ge 0,$$

$$\bar{x}_1 = -1.00, \quad \bar{x}_2 = 0.01.$$

Problem 4 (MAD5 [70]).

$$F(x) = \max_{1\le i\le 3} f_i(x),$$

$$f_i(x) \textit{ as in Problem 3},$$

$$c_1(x) = -0.9x_1 + x_2 - 1 \ge 0,$$

$$\bar{x}_1 = -1, \quad \bar{x}_2 = 3.$$

Problem 5 (PENTAGON [71]).

$$F(x) = \max_{1\le i\le 3} f_i(x),$$

$$f_1(x) = -\sqrt{(x_1 - x_3)^2 + (x_2 - x_4)^2},$$

$$f_2(x) = -\sqrt{(x_3 - x_5)^2 + (x_4 - x_6)^2},$$

$$f_3(x) = -\sqrt{(x_5 - x_1)^2 + (x_6 - x_2)^2},$$

$$x_i \cos\left(\frac{2\pi j}{5}\right) + x_{i+1} \sin\left(\frac{2\pi j}{5}\right) \leq 1,$$
$$i = 1,3,5, \quad j = 0,1,2,3,4.$$

Problem 6 (MAD6 [70]).

$$F(x) = \max_{1 \leq i \leq 163} f_i(x),$$

$$f_i(x) = \frac{1}{15} + \frac{2}{15} \sum_{j=1}^{7} \cos(2\pi x_j \sin \vartheta_i), \quad 1 \leq i \leq 163,$$

$$\vartheta_i = \frac{\pi}{180}(8.5 + i0.5), \quad 1 \leq i \leq 163,$$

$$\begin{aligned}
c_1(x) &= x_1 \geq 0.4, \\
c_2(x) &= -x_1 + x_2 \geq 0.4, \\
c_3(x) &= -x_2 + x_3 \geq 0.4, \\
c_4(x) &= -x_3 + x_4 \geq 0.4, \\
c_5(x) &= -x_4 + x_5 \geq 0.4, \\
c_6(x) &= -x_5 + x_6 \geq 0.4, \\
c_7(x) &= -x_6 + x_7 \geq 0.4, \\
c_8(x) &= -x_4 + x_6 = 1.0, \\
c_9(x) &= x_7 = 3.5,
\end{aligned}$$

$$\begin{aligned}
&\bar{x}_1 = 0.5, \quad \bar{x}_2 = 1.0, \quad \bar{x}_3 = 1.5, \quad \bar{x}_4 = 2.0, \\
&\bar{x}_5 = 2.5, \quad \bar{x}_6 = 3.0, \quad \bar{x}_7 = 3.5.
\end{aligned}$$

Problem 7 (EQUIL [72]).

$$F(x) = \max_{1 \leq i \leq 8} f_i(x),$$

$$f_i(x) = \sum_{j=1}^{5} \left(\frac{a_{ji} \sum_{k=1}^{8} w_{jk} x_k}{x_i^{b_j} \sum_{k=1}^{8} a_{jk} x_k^{1-b_j}} - w_{ji} \right),$$

$$c_1(x) = \sum_{i=1}^{8} x_i = 1, \quad x_i \geq 0, \quad 1 \leq i \leq 8,$$

$$W = [w_{jk}] = \begin{bmatrix} 3 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 6 \\ 0.1 & 10 & 0.1 & 0.1 & 5 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 10 & 0.1 & 4 & 0.1 & 7 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 3 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.1 & 11 \end{bmatrix},$$

$$A = [a_{ij}] = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1.2 & 0.8 & 1 & 1.2 & 1.2 & 0.6 & 0.1 & 1 \\ 2 & 1.2 & 0.8 & 1.2 & 1.2 & 0.1 & 1 & 2 \end{bmatrix},$$

$$b = [b_j] = [0.5 \quad 1.2 \quad 0.8 \quad 2.0 \quad 1.5],$$

$$\bar{x}_i = 0.125, \quad 1 \leq i \leq 8.$$

Problem 8 (Wong 2 [65]).

$$F(x) = \max_{1 \le i \le 6} f_i(x),$$

$$\begin{aligned}
f_1(x) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\
&\quad + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45, \\
f_2(x) &= f_1(x) + 10(3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120), \\
f_3(x) &= f_1(x) + 10(5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40), \\
f_4(x) &= f_1(x) + 10(0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30), \\
f_5(x) &= f_1(x) + 10(x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6), \\
f_6(x) &= f_1(x) + 10(-3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10}), \\
c_1(x) &= 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 105, \\
c_2(x) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0, \\
c_3(x) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} \le 12,
\end{aligned}$$

$$\bar{x}_1 = 2, \quad \bar{x}_2 = 3, \quad \bar{x}_3 = 5, \quad \bar{x}_4 = 5, \quad \bar{x}_5 = 1, \quad \bar{x}_6 = 2,$$

$$\bar{x}_7 = 7, \quad \bar{x}_8 = 3, \quad \bar{x}_9 = 6, \quad \bar{x}_{10} = 10.$$

Problem 9 (Wong 3 [65]).

$$F(x) = \max_{1 \le i \le 11} f_i(x),$$

$$\begin{aligned}
f_1(x) &= x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\
&\quad + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + (x_{11} - 9)^2 \\
&\quad + 10(x_{12} - 1)^2 + 5(x_{13} - 7)^2 + 4(x_{14} - 14)^2 + 3(x_{15} - 1)^2 + x_{16}^4 + (x_{17} - 2)^4 \\
&\quad + 13(x_{18} - 2)^2 + f_2(x) + 95, \\
f_2(x) &= f_1(x) + 10(3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120), \\
f_3(x) &= f_1(x) + 10(5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40), \\
f_4(x) &= f_1(x) + 10(0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 - x_6 - 30), \\
f_5(x) &= f_1(x) + 10(x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6), \\
f_6(x) &= f_1(x) + 10(-3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10}), \\
f_7(x) &= f_1(x) + 10(4x_1 + 5x_2 - 3x_7 + 9x_8 - 105), \\
f_8(x) &= f_1(x) + 10(x_1 + 2x_2 + 4x_{11} - 8x_9 + 30), \\
f_9(x) &= f_1(x) + 10(6x_1^2 + 5x_2 + 3x_{12} - 19), \\
f_{10}(x) &= f_1(x) + 10(5x_1^2 + 5x_2 + 7x_9 - 30x_{10}), \\
f_{11}(x) &= f_1(x) + 10(11x_5^2 + 11x_{15} - 6x_{16} - 54), \\
f_{12}(x) &= f_1(x) + 10(5x_1^2 + 2x_2 + 9x_4 - x_8 - 68), \\
f_{13}(x) &= f_1(x) + 10(4x_1^2 - 19 + 2x_7 - 20x_9 + 19), \\
f_{14}(x) &= f_1(x) + 10(7x_4^2 + 5x_2 + 7x_9 - 30x_{20}), \\
c_1(x) &= 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 105, \\
c_2(x) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0, \\
c_3(x) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} \le 12, \\
c_4(x) &= x_1 + 2x_4 + 2x_{11} - 2x_{12} \le 0,
\end{aligned}$$

$$\begin{aligned}
&\bar{x}_1 = 2, \quad \bar{x}_2 = 3, \quad \bar{x}_3 = 5, \quad \bar{x}_4 = 5, \quad \bar{x}_5 = 1, \quad \bar{x}_6 = 2, \\
&\bar{x}_7 = 7, \quad \bar{x}_8 = 3, \quad \bar{x}_9 = 6, \quad \bar{x}_{10} = 10, \quad \bar{x}_{11} = 2, \quad \bar{x}_{12} = 2, \\
&\bar{x}_{13} = 6, \quad \bar{x}_{14} = 15, \quad \bar{x}_{15} = 1, \quad \bar{x}_{16} = 2, \quad \bar{x}_{17} = 3, \quad \bar{x}_{18} = 2, \\
&\bar{x}_{19} = 1, \quad \bar{x}_{20} = 3.
\end{aligned}$$

Problem 10 (MAD8 [70])**.**

$$F(x) = \max_{1 \le i \le 38} |f_i(x)|,$$

$$\begin{aligned}
f_1(x) &= -1 + x_1^2 + \sum_{j=2}^{20} x_j, \\
f_i(x) &= -1 + c_i x_k^2 + \sum_{\substack{j=1 \\ j \neq k}}^{20} x_j, \quad 1 < i < 38, \\
f_{38}(x) &= -1 + x_{20}^2 + \sum_{j=1}^{19} x_j,
\end{aligned}$$

$$\begin{aligned}
k = \frac{i+2}{2}, \quad c_i = 1, \quad i = 2, 4, \ldots, 36, \\
k = \frac{i+1}{2}, \quad c_i = 2, \quad i = 3, 5, \ldots, 37, \\
x_j \ge 0.5, \quad 1 \le j \le 10, \\
\bar{x}_j = 100, \quad 1 \le j \le 20.
\end{aligned}$$

Problem 11 (BP filter)**.**

$$F(x) = \max_{1 \le i \le 124} f_i(x),$$

$$\begin{aligned}
f_1(x) &= \varphi(x, t_0) - x_9 + a, \\
f_{2j}(x) &= \varphi(x, t_j) - x_9, \quad 1 \le j \le 61, \\
f_{2j+1}(x) &= x_9 - \varphi(x, t_j), \quad 1 \le j \le 61, \\
f_{124}(x) &= \varphi(x, t_{62}) - x_9 + a,
\end{aligned}$$

$$\varphi(x, t) = \frac{1}{2} \sum_{i=1}^{4} \log\Big((x_{i+4}^2 - t^2)^2 + t^2 x_i^2 \Big) - 4 \log(x), \quad a = 3.0164,$$

$$\begin{aligned}
t_0 &= 0.967320, & t_1 &= 0.987423, \\
t_j &= t_1 + (j-1)(t_{61} - t_1), \quad 1 \le j \le 61, \\
t_{61} &= 1.012577, & t_{62} &= 1.032680,
\end{aligned}$$

$$c_i(x) = x_{i+4} - 10000 x_i \le 0, \quad 1 \le i \le 4,$$

$$\begin{aligned}
&\bar{x}_1 = 0.398 \cdot 10^{-1}, \quad \bar{x}_2 = 0.968 \cdot 10^{-4}, \quad \bar{x}_3 = 0.103 \cdot 10^{-3}, \\
&\bar{x}_4 = 0.389 \cdot 10^{-1}, \quad \bar{x}_5 = 1.010, \quad \bar{x}_6 = 0.968, \\
&\bar{x}_7 = 1.030, \quad \bar{x}_8 = 0.981, \quad \bar{x}_9 = -11.6.
\end{aligned}$$

Problem 12 (HS114 [67]).

$$F(x) = \max_{1\le i\le 9} f_i(x),$$

$$\begin{aligned}
f_1(x) &= 5.04x_1 + 0.035x_2 + 10x_3 + 3.36x_5 - 0.063x_4x_7,\\
f_2(x) &= f_1(x) + 500\left(1.12x_1 + 0.13167x_1x_8 - 0.00667x_1x_8^2 - \frac{1}{a}x_4\right),\\
f_3(x) &= f_1(x) - 500\left(1.12x_1 + 0.13167x_1x_8 - 0.00667x_1x_8^2 - ax_4\right),\\
f_4(x) &= f_1(x) + 500\left(1.098x_8 - 0.038x_8^2 + 0.325x_6 - \frac{1}{a}x_7 + 57.425\right),\\
f_5(x) &= f_1(x) - 500\left(1.098x_8 - 0.038x_8^2 + 0.325x_6 - ax_7 + 57.425\right),\\
f_6(x) &= f_1(x) + 500\left(\frac{98000x_3}{x_4x_9 + 10000x_3} - x_6\right),\\
f_7(x) &= f_1(x) - 500\left(\frac{98000x_3}{x_4x_9 + 10000x_3} - x_6\right),\\
f_8(x) &= f_1(x) + 500\left(\frac{x_2 + x_5}{x_1} - x_8\right),\\
f_9(x) &= f_1(x) - 500\left(\frac{x_2 + x_5}{x_1} - x_8\right),\\
c_1(x) &= 0.222x_{10} + bx_9 \le 35.82,\\
c_2(x) &= 0.222x_{10} + \frac{1}{b}x_9 \ge 35.82,\\
c_3(x) &= 3x_7 - ax_{10} \ge 133,\\
c_4(x) &= 3x_7 - \frac{1}{a}x_{10} \le 133,\\
c_5(x) &= 1.22x_4 - x_1 - x_5 = 0,
\end{aligned}$$

$$a = 0.99, \quad b = 0.90,$$

$$10^{-5} \le x_1 \le 2000, \quad \bar{x}_1 = 1745,$$
$$10^{-5} \le x_2 \le 16000, \quad \bar{x}_2 = 12000,$$
$$10^{-5} \le x_3 \le 120, \quad \bar{x}_3 = 110,$$
$$10^{-5} \le x_4 \le 5000, \quad \bar{x}_4 = 3048,$$
$$10^{-5} \le x_5 \le 2000, \quad \bar{x}_5 = 1974,$$
$$85 \le x_6 \le 93, \quad \bar{x}_6 = 89.2,$$
$$90 \le x_7 \le 95, \quad \bar{x}_7 = 92.8,$$
$$3 \le x_8 \le 12, \quad \bar{x}_8 = 8.0,$$
$$1.2 \le x_9 \le 4, \quad \bar{x}_9 = 3.6,$$
$$145 \le x_{10} \le 162, \quad \bar{x}_{10} = 145.$$

Problem 13 (Dembo 3 [73]).

$$F(x) = \max_{1 \le i \le 13} f_i(x),$$

$$\begin{aligned}
f_1(x) &= a_1x_1 + a_2x_1x_6 + a_3x_3 + a_4x_2 + a_5 + a_6x_3x_5, \\
f_2(x) &= f_1(x) + 10^5\left(a_7x_6^2 + a_8x_1x_3^{-1}x_3 + a_9x_6 - 1\right), \\
f_3(x) &= f_1(x) + 10^5\left(a_{10}x_1x_3^{-1} + a_{11}x_1x_3^{-1}x_6 + a_{12}x_1x_3^{-1}x_6^2 - 1\right), \\
f_4(x) &= f_1(x) + 10^5\left(a_{13}x_6^2 + a_{14}x_5 + a_{15}x_4 + a_{16}x_6 - 1\right), \\
f_5(x) &= f_1(x) + 10^5\left(a_{17}x_5^{-1} + a_{18}x_5^{-1}x_6 + a_{19}x_4x_5^{-1} + a_{20}x_5^{-1}x_6^2 - 1\right), \\
f_6(x) &= f_1(x) + 10^5\left(a_{21}x_7 + a_{22}x_3x_3^{-1}x_4 + a_{23}x_2x_3^{-1}x_7 - 1\right), \\
f_7(x) &= f_1(x) + 10^5\left(a_{24}x_7^{-1} + a_{25}x_2x_3^{-1}x_7^{-1} + a_{26}x_2x_3^{-1}x_7^{-1} - 1\right), \\
f_8(x) &= f_1(x) + 10^5\left(a_{27}x_5 + a_{28}x_7^{-1} - 1\right), \\
f_9(x) &= f_1(x) + 10^5\left(a_{29}x_3^{-1} + a_{30}x_4^{-1}x_3^{-1}x_4^{-1} - 1\right), \\
f_{10}(x) &= f_1(x) + 10^5\left(a_{31}x_4^{-1}x_4^{-1}x_7^{-1} - 1\right), \\
f_{11}(x) &= f_1(x) + 10^5\left(a_{32}x_4x_3^{-1}x_7^{-1} - 1\right), \\
f_{12}(x) &= f_1(x) + 10^5(a_{33}x_1x_6 + a_{34}x_6 - 1), \\
f_{13}(x) &= f_1(x) + 10^5(a_{42}x_1x_3 + a_{43}x_1 + a_{44}x_6 - 1),
\end{aligned}$$

$$\begin{aligned}
c_1(x) &= a_{29}x_5 + a_{30}x_7 \le 1, \\
c_2(x) &= a_{31}x_3 + a_{32}x_1 \le 1,
\end{aligned}$$

i	a_i	i	a_i	i	a_i
1	1.715	16	$0.19120592 \times 10^{-1}$	31	0.000061000
2	0.035	17	0.56850750×10^{2}	32	−0.0005
3	4.0565	18	1.08702000	33	0.81967200
4	10.0	19	0.32175000	34	0.81967200
5	3000.0	20	−0.03762000	35	24500.0
6	−0.063	21	0.00619800	36	−250.0
7	$0.59553571 \times 10^{-2}$	22	0.24623121×10^{4}	37	$0.10204082 \times 10^{-1}$
8	0.88392857	23	$-0.25125634 \times 10^{2}$	38	$0.12244898 \times 10^{-4}$
9	−0.11756250	24	0.16118996×10^{3}	39	0.00006250
10	1.10880000	25	5000.0	40	0.00006250
11	0.13035330	26	$-0.48951000 \times 10^{6}$	41	−0.00007625
12	−0.00660330	27	0.14433333×10^{2}	42	1.22
13	$0.66173269 \times 10^{-3}$	28	0.33000000	43	1.0
14	$0.17239878 \times 10^{-1}$	29	0.02256000	44	−1.0
15	$-0.56595559 \times 10^{-2}$	30	−0.00759500		

$$
\begin{aligned}
1 \le x_1 &\le 2000, & \bar{x}_1 &= 1745,\\
1 \le x_2 &\le 120, & \bar{x}_2 &= 110,\\
1 \le x_3 &\le 5000, & \bar{x}_3 &= 3048,\\
85 \le x_4 &\le 93, & \bar{x}_4 &= 89,\\
90 \le x_5 &\le 95, & \bar{x}_5 &= 92,\\
3 \le x_6 &\le 12, & \bar{x}_6 &= 8,\\
145 \le x_7 &\le 162, & \bar{x}_7 &= 145,
\end{aligned}
$$

Problem 14 (Dembo 5 [73]).

$$F(x) = \max_{1\le i\le 4} f_i(x),$$

$$
\begin{aligned}
f_1(x) &= x_1 + x_2 + x_3,\\
f_2(x) &= f_1(x) + 10^5\left(ax_1^{-1}x_4x_6^{-1} + 100x_6^{-1} + bx_1^{-1}x_6^{-1} - 1\right),\\
f_3(x) &= f_1(x) + 10^5\left(x_4x_7^{-1} + 1250(x_5 - x_4)x_2^{-1}x_7^{-1} - 1\right),\\
f_4(x) &= f_1(x) + 10^5\left(cx_3^{-1}x_8^{-1} + x_5x_8^{-1} - 2500x_3x_5x_8^{-1} - 1\right),
\end{aligned}
$$

$$
\begin{aligned}
c_1(x) &= 0.0025(x_4 + x_6) \le 1,\\
c_2(x) &= 0.0025(x_5 + x_7 - x_4) \le 1,\\
c_3(x) &= 0.01(x_8 - x_5) \le 1,
\end{aligned}
$$

$$a = 833.33252, \quad b = -83333.333, \quad c = 1250000.0.$$

$$
\begin{aligned}
100 \le x_1 &\le 10000, & \bar{x}_1 &= 5000,\\
1000 \le x_2 &\le 10000, & \bar{x}_2 &= 5000,\\
1000 \le x_3 &\le 10000, & \bar{x}_3 &= 5000,\\
10 \le x_4 &\le 1000, & \bar{x}_4 &= 200,\\
10 \le x_5 &\le 1000, & \bar{x}_5 &= 350,\\
10 \le x_6 &\le 1000, & \bar{x}_6 &= 150,\\
10 \le x_7 &\le 1000, & \bar{x}_7 &= 225,\\
10 \le x_8 &\le 1000, & \bar{x}_8 &= 425.
\end{aligned}
$$

Problem 15 (Dembo 7 [73]).

$$F(x) = \max_{1\le i\le 19} f_i(x),$$

$$
\begin{aligned}
f_1(x) &= a(x_{12} + x_{13} + x_{14} + x_{15} + x_{16}) + b(x_1x_{12} + x_2x_{13} + x_3x_{14} + x_4x_{15} + x_5x_{16}),\\
f_2(x) &= f_1(x) + 10^3\left(cx_1x_6^{-1} + 100dx_1 - dx_1^2x_6^{-1} - 1\right),\\
f_3(x) &= f_1(x) + 10^3\left(cx_2x_7^{-1} + 100dx_2 - dx_2^2x_7^{-1} - 1\right),\\
f_4(x) &= f_1(x) + 10^3\left(cx_3x_8^{-1} + 100dx_3 - dx_3^2x_8^{-1} - 1\right),
\end{aligned}
$$

$$
\begin{aligned}
f_5(x) &= f_1(x) + 10^3\left(cx_4x_9^{-1} + 100dx_4 - dx_4^2x_9^{-1} - 1\right),\\
f_6(x) &= f_1(x) + 10^3\left(cx_5x_{10}^{-1} + 100dx_5 - dx_5^2x_{10}^{-1} - 1\right),\\
f_7(x) &= f_1(x) + 10^3\left(x_6x_7^{-1} + x_1x_7^{-1}x_{11}x_{12}^{-1} - x_6x_7^{-1}x_{11}x_{12}^{-1} - 1\right),\\
f_8(x) &= f_1(x) + 10^3\left(x_7x_8^{-1} + 0.002(x_7 - x_1)x_{12}^{-1} + 0.002(x_2x_8^{-1} - 1)x_{13} - 1\right),\\
f_9(x) &= f_1(x) + 10^3(x_8 + 0.002(x_8 - x_2)x_{13} + 0.002(x_3 - x_9)x_{14} + x_9 - 1),\\
f_{10}(x) &= f_1(x) + 10^3\left(x_3^{-1}x_9 + (x_4 - x_8)x_3^{-1}x_{14}x_{15} + 500(x_{10} - x_9)x_3^{-1}x_{14}^{-1} - 1\right),\\
f_{11}(x) &= f_1(x) + 10^3\left((x_1^{-1}x_5 - 1)x_{16}x_{14}^{-1}x_{10} + 500(1 - x_1^{-1}x_{10})x_{15}^{-1}x_{14} - 1\right),\\
f_{12}(x) &= f_1(x) + 10^3\left(0.9x_1^{-1}x_4 + 0.002(1 - x_1^{-1}x_5)x_{16} - 1\right),\\
f_{13}(x) &= f_1(x) + 10^3\left(x_{11}x_{12}^{-1} - 1\right),\\
f_{14}(x) &= f_1(x) + 10^3\left(x_4x_5^{-1} - 1\right),\\
f_{15}(x) &= f_1(x) + 10^3\left(x_3x_4^{-1} - 1\right),\\
f_{16}(x) &= f_1(x) + 10^3\left(x_2x_3^{-1} - 1\right),\\
f_{17}(x) &= f_1(x) + 10^3\left(x_1x_2^{-1} - 1\right),\\
f_{18}(x) &= f_1(x) + 10^3\left(x_9x_{10}^{-1} - 1\right),\\
f_{19}(x) &= f_1(x) + 10^3\left(x_8x_9^{-1} - 1\right),
\end{aligned}
$$

$$c_1(x) = 0.002(x_{11} - x_{12}) \leq 1,$$

$$a = 1.262626, \quad b = -1.231060, \quad c = 0.034750, \quad d = 0.009750.$$

$$
\begin{aligned}
0.1 \leq x_1 \leq 0.9, &\quad \bar{x}_1 = 0.80,\\
0.1 \leq x_2 \leq 0.9, &\quad \bar{x}_2 = 0.83,\\
0.1 \leq x_3 \leq 0.9, &\quad \bar{x}_3 = 0.85,\\
0.1 \leq x_4 \leq 0.9, &\quad \bar{x}_4 = 0.87,\\
0.9 \leq x_5 \leq 1.0, &\quad \bar{x}_5 = 0.90,\\
10^{-4} \leq x_6 \leq 0.1, &\quad \bar{x}_6 = 0.10,\\
0.1 \leq x_7 \leq 0.9, &\quad \bar{x}_7 = 0.12,\\
0.1 \leq x_8 \leq 0.9, &\quad \bar{x}_8 = 0.19,\\
0.1 \leq x_9 \leq 0.9, &\quad \bar{x}_9 = 0.25,\\
0.1 \leq x_{10} \leq 0.9, &\quad \bar{x}_{10} = 0.29,\\
1 \leq x_{11} \leq 1000, &\quad \bar{x}_{11} = 512,\\
10^{-6} \leq x_{12} \leq 500, &\quad \bar{x}_{12} = 13.1,\\
1 \leq x_{13} \leq 500, &\quad \bar{x}_{13} = 71.8,\\
500 \leq x_{14} \leq 1000, &\quad \bar{x}_{14} = 640,\\
500 \leq x_{15} \leq 1000, &\quad \bar{x}_{15} = 650,\\
10^{-6} \leq x_{16} \leq 500, &\quad \bar{x}_{16} = 5.7.
\end{aligned}
$$

Author Contributions: Conceptualization, M.A.T.; methodology, M.A.T. and A.A.A.; software, A.A.A.; validation, M.A.T., M.O. and A.A.A.; formal analysis, M.A.T. and A.A.A.; investigation, M.A.T. and A.A.A.; resources, M.A.T. and M.O.; data curation, A.A.A.; writing—original draft preparation, A.A.A.; writing—review and editing, M.A.T. and M.O.; visualization, M.A.T. and A.A.A.; supervision, M.A.T.; project administration, M.A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on reasonable request from the corresponding author.

Acknowledgments: The research of the second author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The postdoctoral fellowship of the first author is supported by NSERC.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cherkaev, E.; Cherkaev, A. Minimax optimization problem of structural design. *Comput. Struct.* **2008**, *86*, 1426–1435. [CrossRef]
2. Lanckriet, G.R.; Ghaoui, L.E.; Bhattacharyya, C.; Jordan, M.I. A robust minimax approach to classification. *J. Mach. Learn. Res.* **2002**, *3*, 555–582.
3. Rapoport, A.; Boebel, R.B. Mixed strategies in strictly competitive games: A further test of the minimax hypothesis. *Games Econ. Behav.* **1992**, *4*, 261–283. [CrossRef]
4. Kim, S.J.; Boyd, S. A minimax theorem with applications to machine learning, signal processing, and finance. *SIAM J. Optim.* **2008**, *19*, 1344–1367. [CrossRef]
5. Yu, G. Min-max optimization of several classical discrete optimization problems. *J. Optim. Theory Appl.* **1998**, *98*, 221–242. [CrossRef]
6. Cramer, M.; Sudhoff, S.D.; Zivi, E.L. Evolutionary algorithms for minimax problems in robust design. *IEEE Trans. Evol. Comput.* **2009**, *13*, 444–453. [CrossRef]
7. Cao, F.; Du, D.Z.; Gao, B.; Wan, P.J.; Pardalos, P.M. Minimax problems in combinatorial optimization. In *Minimax and Applications*; Springer: Boston, MA, USA, 1995; pp. 269–292.
8. Vias, P.H. Quantified Real Constraint Solving Using Modal Intervals with Applications to Control. Ph.D. Thesis, Universitat de Girona, Girona, Spain, 2006.
9. Wie, B.; Bernstein, D.S. Benchmark problems for robust control design. *J. Guid. Control Dyn.* **1992**, *15*, 1057–1059. [CrossRef]
10. Liuzzi, G.; Lucidi, S.; Sciandrone, M. A derivative-free algorithm for linearly constrained finite minimax problems. *SIAM J. Optim.* **2006**, *16*, 1054–1075. [CrossRef]
11. Polak, E.; Royset, J.O.; Womersley, R.S. Algorithms with adaptive smoothing for finite minimax problems. *J. Optim. Theory Appl.* **2003**, *119*, 459–484. [CrossRef]
12. Xu, S. Smoothing method for minimax problems. *Comput. Optim. Appl.* **2001**, *20*, 267–279. [CrossRef]
13. Han, S.P. A globally convergent method for nonlinear programming. *J. Optim. Theory Appl.* **1977**, *22*, 297–309. [CrossRef]
14. Powell, M.J.D. A fast algorithm for nonlinearly constrained optimization calculations. In *Numerical Analysis*; Lecture Notes in Math; Springer: New York, NY, USA, 1978; Volume 630, pp. 144–157.
15. Fletcher, R. *Practical Method of Optimization*; John Wiley and Sons: Hoboken, NJ, USA, 1980.
16. Gill, P.E.; Wong, E. Sequential quadratic programming methods. In *Mixed Integer Nonlinear Programming*; Springer: New York, NY, USA, 2012; pp. 147–224.
17. Wang, L.; Cao, Q.; Zhang, Z.; Mirjalili, S.; Zhao, W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Knowl.-Based Syst.* **2022**, *244*, 108517. [CrossRef]
18. Wang, Y.; Xiao, Y.; Guo, Y.; Li, J. Dynamic chaotic opposition-based learning-driven hybrid Aquila Optimizer and artificial rabbits optimization algorithm: Framework and applications. *Processes* **2022**, *10*, 2703. [CrossRef]
19. Wang, Y.; Huang, L.; Zhong, J.; Hu, G. LARO: Opposition-based learning boosted artificial rabbits-inspired optimization algorithm with Lévy flight. *Symmetry* **2022**, *14*, 2282. [CrossRef]
20. Dangi, D.; Telang Chandel, S.; Kumar Dixit, D.; Sharma, S.; Bhagat, A. An efficient model for sentiment analysis using artificial rabbits optimized vector functional link network. *Expert Syst. Appl.* **2023**, *225*, 119849. [CrossRef]
21. Elshahed, M.; Tolba, M.A.; El-Rifaie, A.M.; Ginidi, A.; Shaheen, A.; Mohamed, S.A. An artificial rabbits' optimization to allocate PVSTATCOM for ancillary service provision in distribution systems. *Mathematics* **2023**, *11*, 339. [CrossRef]

22. Alsaiari, A.O.; Moustafa, E.B.; Alhumade, H.; Abulkhair, H.; Elsheikh, A. A coupled artificial neural network with artificial rabbits optimizer for predicting water productivity of different designs of solar stills. *Adv. Eng. Softw.* **2023**, *175*, 103315. [CrossRef]
23. Riad, A.J.; Hasanien, H.M.; Turky, R.A.; Yakout, A.H. Identifying the PEM fuel cell parameters using artificial rabbits optimization algorithm. *Sustainability* **2023**, *15*, 4625. [CrossRef]
24. He, N.-N.; Wang, W.-C.; Wang, J. Multi-strategy enhanced artificial rabbit optimization algorithm for solving engineering optimization problems. *Evol. Intell.* **2025**, *18*, 24. [CrossRef]
25. Jia, H.; Su, Y.; Rao, H.; Liang, M.; Abualigah, L.; Liu, C.; Chen, X. Improved artificial rabbits algorithm for global optimization and multi-level thresholding color image segmentation. *Artif. Intell. Rev.* **2024**, *58*, 55. [CrossRef]
26. Al-Daom, E.; Abed-alguni, B.H. BIARO: An improved artificial rabbit optimization algorithm for feature selection in high-dimensional data. *Clust. Comput.* **2025**, *28*, 868. [CrossRef]
27. Anka, F.; Agaoglu, N.; Nematzadeh, S.; Torkamanian-afshar, M.; Gharehchopogh, F.S. Advances in Artificial Rabbits Optimization: A Comprehensive Review: F. Anka et al. *Arch. Comput. Methods Eng.* **2025**, *32*, 2113–2148. [CrossRef]
28. Hilal, A.M.; Abdalla Hashim, A.H.; Obayya, M.; Gaddah, A.; Mohamed, A.; Yaseen, I.; Rizwanullah, M.; Zamani, A.S. Metaheuristics based energy efficient task scheduling scheme for cyber-physical systems environment. *Sustainability* **2022**, *14*, 16539. [CrossRef]
29. Mazloumi, A.; Poolad, A.; Mokhtari, M.S.; Altman, M.B.; Abdelaziz, A.Y.; Elsisi, M. Optimal sizing of a photovoltaic pumping system integrated with water storage tank considering cost/reliability assessment using enhanced artificial rabbits optimization: A case study. *Mathematics* **2023**, *11*, 463. [CrossRef]
30. Cao, Q.; Wang, L.; Zhao, W.; Yuan, Z.; Liu, A.; Gao, Y.; Ye, R. Vibration state identification of hydraulic units based on improved artificial rabbits optimization algorithm. *Biomimetics* **2023**, *8*, 243. [CrossRef]
31. Awadallah, M.A.; Braik, M.S.; Al-Betar, M.A.; Abu Doush, I. An enhanced binary artificial rabbits optimization for feature selection in medical diagnosis. *Neural Comput. Appl.* **2023**, *35*, 20013–20068. [CrossRef]
32. Vellingiri, M.; Rawa, M.; Alghamdi, S.; Alhussainy, A.A.; Ali, Z.M.; Turky, R.A.; Refaat, M.M.; Aleem, S.H.E.A. Maximum hosting capacity estimation for renewables in power grids considering energy storage and transmission lines expansion using hybrid sine cosine artificial rabbits algorithm. *Ain Shams Eng. J.* **2023**, *14*, 102092. [CrossRef]
33. Kumar, D.S.; Premkumar, M.; Kumar, C.; Muyeen, S.M. Optimal scheduling algorithm for residential building distributed energy source systems using Levy flight and chaos-assisted artificial rabbits optimizer. *Energy Rep.* **2023**, *9*, 5721–5740. [CrossRef]
34. Alamir, N.; Kamel, S.; Hassan, M.H.; Abdelkader, S.M. An effective quantum artificial rabbits optimizer for energy management in microgrid considering demand response: N. Alamir et al. *Soft Comput.* **2023**, *27*, 15741–15768. [CrossRef]
35. Sharif, R.H.; Masdari, M.; Ghaffari, A.; Gharehchopogh, F.S. A chaotic-based artificial rabbit optimization and dandelion optimizer for QoS-aware web service composition in mobile edge computing. *Neural Comput. Appl.* **2025**, *37*, 18549–18608. [CrossRef]
36. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]
37. Chen, Z.; Francis, A.; Li, S.; Liao, B.; Xiao, D.; Ha, T.T.; Li, J.; Ding, L.; Cao, X. Egret swarm optimization algorithm: An evolutionary computation approach for model free optimization. *Biomimetics* **2022**, *7*, 144. [CrossRef]
38. Yang, Y.; Chen, H.; Heidari, A.A.; Gandomi, A.H. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* **2021**, *177*, 114864. [CrossRef]
39. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [CrossRef]
40. Talatahari, S.; Azizi, M.; Tolouei, M.; Talatahari, B.; Sareh, P. Crystal structure algorithm (CryStAl): A metaheuristic optimization method. *IEEE Access* **2021**, *9*, 71244–71261. [CrossRef]
41. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95—International Conference on Neural Networks*; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948.
42. Tumová, E.; Martinec, M.; Chodová, D. Analysis of Czech rabbit genetic resources. *Sci. Agric. Bohem.* **2011**, *42*, 113–118.
43. List, G. *Animal Tales from the Caribbean*; Indiana University Press: Indiana, IL, USA, 2017.
44. Tynes, V.V. *Behavior of Exotic Pets*; John Wiley & Sons: New York, NY, USA, 2010.
45. Camp, M.J.; Rachlow, J.L.; Shipley, L.A.; Johnson, T.R.; Bockting, K.D. Grazing in sagebrush rangelands in western North America: Implications for habitat quality for a sagebrush specialist, the pygmy rabbit. *Rangel. J.* **2014**, *36*, 151–159. [CrossRef]
46. Firestone, C.Z.; Warren, W.H. Why does the rabbit escape the fox on a zig-zag path? Predator-prey dynamics and the constant bearing strategy. *J. Vis.* **2010**, *10*, 1049. [CrossRef]
47. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimization algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]
48. Wang, N.; Liu, L.; Liu, L. Genetic algorithm in chaos. *OR Trans.* **2001**, *5*, 1–10.
49. Yang, L.J.; Chen, T.L. Application of chaos in genetic algorithms. *Commun. Theor. Phys.* **2002**, *38*, 68–172. [CrossRef]

50. Jothiprakash, V.; Arunkumar, R. Optimization of hydropower reservoir using evolutionary algorithms coupled with chaos. *Water Resour. Manag.* **2013**, *27*, 1963–1979. [CrossRef]
51. Zhenyu, G.; Bo, C.; Min, Y.; Binggang, C. Self-adaptive chaos differential evolution. In *International Conference on Natural Computation*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 972–975.
52. Bhattacharya, A.; Chattopadhyay, P.K. Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Trans. Power Syst.* **2010**, *25*, 1955–1964. [CrossRef]
53. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
54. Charalambous, C.; Conn, A.R. An efficient method to solve the minimax problem directly. *SIAM J. Numer. Anal.* **1978**, *15*, 162–187. [CrossRef]
55. Polak, E.; Higgins, J.E.; Mayne, D.Q. A barrier function method for minimax problems. *Math. Program.* **1992**, *54*, 155–176. [CrossRef]
56. El-Attar, R.A.; Vidyasagar, M.; Dutta, S.R.K. An algorithm for l_1-norm minimization with application to nonlinear l_1-approximation. *SIAM J. Numer. Anal.* **1979**, *16*, 70–86. [CrossRef]
57. Rosen, J.B.; Suzuki, S. Pracniques: Construction of nonlinear programming test problems. *Commun. ACM* **1965**, *8*, 113. [CrossRef]
58. Polak, E.; Mayne, D.Q.; Higgins, J.E. Superlinearly convergent algorithm for min-max problems. *J. Optim. Theory Appl.* **1991**, *69*, 407–439. [CrossRef]
59. Popović, J.R.; Bandler, J.W.; Charalambous, C. General programmes for least pth and near minimax approximation. *Int. J. Syst. Sci.* **1974**, *5*, 907–932. [CrossRef]
60. Watson, G.A. The minimax solution of an overdetermined system of non-linear equations. *IMA J. Appl. Math.* **1979**, *23*, 167–180. [CrossRef]
61. Zhou, J.L.; Tits, A.L. An SQP algorithm for finely discretized continuous minimax problems and other minimax problems with many objective functions. *SIAM J. Optim.* **1996**, *6*, 461–487. [CrossRef]
62. Lukšan, L.; Vlcek, J. *Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization*; Technical Report No. 798; Institute of Computer Science, Academy of Sciences of the Czech Republic: Prague, Czech Republic, 2000.
63. Hald, J.; Madsen, K. Combined LP and quasi-Newton methods for minimax optimization. *Math. Program.* **1981**, *20*, 49–62. [CrossRef]
64. Charalambous, C. Acceleration of the least p th algorithm for minimax optimization with engineering applications. *Math. Program.* **1979**, *17*, 270–297. [CrossRef]
65. Asaadi, J. A computational comparison of some non-linear programs. *Math. Program.* **1973**, *4*, 144–154. [CrossRef]
66. Mäkela, M.M.; Neittaanmäki, P. *Nonsmooth Optimization*; World Scientific: Singapore, 1992.
67. Hock, W.; Schittkowski, K. Test examples for nonlinear programming codes. *J. Optim. Theory Appl.* **1980**, *30*, 127–129. [CrossRef]
68. Kiwiel, K.C. An ellipsoid trust region bundle method for nonsmooth convex minimization. *SIAM J. Control Optim.* **1989**, *27*, 737–757. [CrossRef]
69. Schwefel, H.-P. *Evolution and Optimum Seeking*; Wiley: New York, NY, USA, 1995.
70. Madsen, K.; Schjær-Jacobsen, H. Linearly constrained minimax optimization. *Math. Program.* **1978**, *14*, 208–223. [CrossRef]
71. Powel, M.J.D. *TOLMIN: A Fortran Package for Linearly Constrained Optimization Calculations*; DAMTP/NA2; Department of Applied Mathematics and Theoretical Physics, University of Cambridge: Cambridge, UK, 1989.
72. Lemaréchal, C.; Mifflin, R. (Eds.) *Nonsmooth Optimization*; Pergamon Press: Oxford, UK, 1978.
73. Dembo, R.S. A set of geometric programming test problems and their solutions. *Math. Program.* **1976**, *10*, 192–213. [CrossRef]