



ARTICLE

An Enhanced Equilibrium Optimizer for Solving Optimization Tasks

Yuting Liu¹, Hongwei Ding^{1,*}, Zongshan Wang^{1,*}, Gaurav Dhiman^{2,3,4}, Zhijun Yang¹ and Peng Hu⁵

¹School of Information Science and Engineering, Yunnan University, Kunming, 650500, China

²Department of Electrical and Computer Engineering, Lebanese American University, Byblos, 1401, Lebanon

³University Centre for Research and Development, Department of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali, 140413, India

⁴Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun, 248002, India

⁵Research and Development Department, Youbei Technology Co., Ltd., Kunming, 650500, China

*Corresponding Authors: Hongwei Ding. Email: lyt20221211@163.com; Zongshan Wang. Email: wzs694513739@163.com

Received: 22 February 2023 Accepted: 11 August 2023 Published: 29 November 2023

ABSTRACT

The equilibrium optimizer (EO) represents a new, physics-inspired metaheuristic optimization approach that draws inspiration from the principles governing the control of volume-based mixing to achieve dynamic mass equilibrium. Despite its innovative foundation, the EO exhibits certain limitations, including imbalances between exploration and exploitation, the tendency to local optima, and the susceptibility to loss of population diversity. To alleviate these drawbacks, this paper introduces an improved EO that adopts three strategies: adaptive inertia weight, Cauchy mutation, and adaptive sine cosine mechanism, called SCEO. Firstly, a new update formula is conceived by incorporating an adaptive inertia weight to reach an appropriate balance between exploration and exploitation. Next, an adaptive sine cosine mechanism is embedded to boost the global exploratory capacity. Finally, the Cauchy mutation is utilized to prevent the loss of population diversity during searching. To validate the efficacy of the proposed SCEO, a comprehensive evaluation is conducted on 15 classical benchmark functions and the CEC2017 test suite. The outcomes are subsequently benchmarked against both the conventional EO, its variants, and other cutting-edge metaheuristic techniques. The comparisons reveal that the SCEO method provides significantly superior results against the standard EO and other competitors. In addition, the developed SCEO is implemented to deal with a mobile robot path planning (MRPP) task, and compared to some classical metaheuristic approaches. The analysis results demonstrate that the SCEO approach provides the best performance and is a prospective tool for MRPP.

KEYWORDS

Metaheuristic algorithms; equilibrium optimizer; Cauchy mutation; robot path planning

1 Introduction

As the advancement of the society rapidly, a number of optimization issues are required to be resolved. These problems are previously usually tackled with traditional numerical computational



methods, but these are inefficient and difficult to deal with high-dimensional and non-linear optimization problems [1]. Consequently, researchers have pioneered the development of metaheuristic algorithms, drawing inspiration from the intricate social behaviors exhibited by biological colonies and the governing principles of natural phenomena. These algorithms, characterized by their stochastic nature and the absence of a need for priori knowledge, exhibit remarkable flexibility in their problem-solving capabilities [2].

Recently, some excellent metaheuristics have been developed and successfully employed to address optimization problems in different domains. In [3], the hunger games search algorithm (HGS) is introduced as an innovative approach. This algorithm takes its inspiration from the behaviors and behavioral selections of animals driven by the necessity to overcome hunger. The performance of HGS is verified on function cases and engineering design cases, and the results prove that HGS has excellent optimization performance. In [4], a novel metaheuristic technique based on the idea of weight means is developed, called INFO. Comparative experiments with basic and advanced algorithms prove that INFO has outstanding convergence performance. In [5], a novel algorithm, the rime optimization algorithm (RIME), is proposed. RIME simulates the growth and crossover behavior of the rime-particle population. Extensive comparative experiments justify that the RIME has promising convergence performance. In [6], the coati optimization algorithm (COA) that draws inspiration from the natural behaviors of coatis, is introduced as a solution for tackling global optimization problems. COA obtains superior performance on functional problems and some practical cases. In [7], a nature-inspired method called the nutcracker optimizer (NOA) is proposed. NOA is mimicking the search, hiding, and recovery behaviors of nutcrackers. Comprehensive experimental results verify the performance of NOA. Well-performing algorithms, including those mentioned above, provide researchers in different fields with tools to resolve optimization problems in their domains. Nature-inspired approaches are successfully applied in different fields and metaheuristic techniques are gaining popularity in fields such as engineering and science [8].

This article directs its focus towards a novel physics-based metaheuristic algorithm referred to as EO. Initially introduced by Faramarzi et al. in 2020 [9], EO presents a range of compelling advantages. These encompass a streamlined framework, straightforward operational principles, simplified implementation, a reduced parameter set, and expedited processing times. Furthermore, EO has showcased its prowess in optimization, surpassing the performance of certain other metaheuristics like PSO [10] and FA [11]. It is widely embraced as a solution for numerous intricate optimization challenges. For example, in [12], the adaptive EO (AEO) is introduced for multi-level threshold processing, which modifies the basic EO by making adaptive decisions on the dispersion of non-performance search agents. The performance of the AEO is assessed on benchmark function cases and compared with other methods. The comparison shows that AEO outperforms other competitors. Then the AEO is employed to optimize the multilevel thresholding problem and obtain the best value by minimizing the objective function. In [13], an improved EO algorithm (IEO) incorporating linear reduced discretization and local minimum elimination method is proposed. A number of comparative experiments indicate that IEO is much superior. This algorithm effectively predicts unknown parameters within the single diode model, double diode model, and triple diode model. In [14], an enhanced EO is presented to tackle the challenges of parameter identification across various photovoltaic (PV) cell models. The analysis illustrates that the improved EO has highly competitive advantages and can accurately extract PV parameters.

Despite EO's exceptional performance in tackling functional problems and engineering design cases, it does exhibit certain drawbacks, such as the exploration-exploitation imbalance, population diversity loss, and susceptibility to local optima. To address these limitations, numerous scholars have

proposed a range of enhanced EO algorithms. In [15], an adaptive EO (LWMEO) with three mechanisms is developed. The effectiveness of LWMEO is rigorously assessed using CEC2014 functions and classical multimodal functions, revealing its competitive performance. In [16], a modified EO algorithm (m-EO) is put forth, featuring Gaussian mutation and an exploitative search mechanism grounded in population segmentation and reestablishment principles. Through extensive testing on 33 benchmark problems and comparisons to the EO and other advanced approaches, the m-EO demonstrates its superiority over competitors. In [17], an enhanced iteration of EO (IEO) is introduced, incorporating a declining equilibrium pool strategy. The evaluation of IEO encompasses 29 benchmark functions, along with three engineering task, highlighting its versatility and efficacy.

Although these EO variants strengthen the overall performance of the basic EO, the shortcomings of the exploration-exploitation imbalance, the proneness to become trapped in local optimization, and the loss of population diversity have not been adequately addressed. This is primarily because most of the existing EO variants are designed to improve one of the limitations, rather than all of them. Based on the above motivation, the present work introduces a novel EO variant, referred to as SCEO, to refine the performance of the canonical EO. The key points of this study are outlined below.

- (1) A new EO version is introduced, known as SCEO, which involves three strategies, namely adaptive inertial weight, Cauchy mutation, and adaptive sine cosine mechanism. The component study in Section 4.4 examines the validity of the three mechanisms of SCEO.
- (2) To assess the performance of the newly SCEO, it undergoes rigorous testing on 15 classical benchmark functions as well as the comprehensive CEC2017 test suite. This assessment involves a comparative analysis against state-of-the-art techniques, as well as the foundational EO and its various iterations. Additionally, the gathered experimental data undergoes meticulous statistical scrutiny through the utilization of both the Friedman test and the Wilcoxon signed rank test.
- (3) To substantiate the practical viability of the SCEO algorithm, its capabilities are demonstrated through its application to an MRPP task. The algorithm can plan satisfactory paths for mobile robots in different environments.

The subsequent sections are structured as follows: Section 2 outlines the canonical EO for contextual reference. Section 3 elaborates on the three pivotal enhancement strategies and introduces the refined SCEO approach. In Section 4, an exhaustive evaluation of SCEO's effectiveness is conducted across the spectrum of 15 classical benchmark functions and the comprehensive CEC2017 test suite. Furthermore, Section 5 delves into the SCEO-driven path-planning technique. Finally, Section 6 encapsulates this study with a comprehensive summary and conclusion.

2 Basic EO Algorithm

2.1 Initialization

Similar to most metaheuristic methods, EO utilizes initialized candidate solutions to commence the optimization process. The initial concentration is described below.

$$X_i = X_{\min} + \text{randi}(X_{\max} - X_{\min}), i = 1, 2, \dots, N \quad (1)$$

where X_i is the concentration of the i -th initial particle; X_{\max} and X_{\min} denote the upper and lower boundaries of the exploration space, respectively; N signifies the scale of the population under consideration.

2.2 Equilibrium Pool

The four particles with the optimal fitness and the mean of these four particles are selected to create the equilibrium pool. X_{eq_pool} and X_{eq_ave} are expressed below, respectively.

$$X_{eq_pool} = \{X_{eq1}, X_{eq2}, X_{eq3}, X_{eq4}, X_{eq_ave}\} \quad (2)$$

$$X_{eq_ave} = \frac{X_{eq1} + X_{eq2} + X_{eq3} + X_{eq4}}{4} \quad (3)$$

2.3 Exponential Term (F)

The exponential factor F plays a crucial role in maintaining the delicate exploration-exploitation balance. This factor is determined by Eq. (4).

$$F = e^{-\lambda(t-t_0)} \quad (4)$$

where λ represents a randomly generated variable within the range of $[0,1]$; t is a non-linear computation, formulated as follows.

$$t = \left(1 - \frac{It}{T}\right)^{\left(a_2 \frac{t}{T}\right)} \quad (5)$$

where It corresponds to the current iteration count and T corresponds to the predefined maximum iteration count; a_2 stands as a constant parameter. t_0 is expressed below.

$$t_0 = \frac{1}{\lambda} \ln(-a_1 \text{sign}(r - 0.5) [1 - e^{-\lambda t}]) + t \quad (6)$$

where a_1 takes on a value of 2 and a_2 takes on a value of 1; $\text{sign}(r - 0.5)$ effectively determines the direction of development; r symbolizes a random variable confined within the range of $[0,1]$. Substituting Eq. (6) into Eq. (4), we get:

$$F = a_1 \text{sign}(r - 0.5) [1 - e^{-\lambda t}] \quad (7)$$

2.4 Generation Rate (G)

The parameter responsible for regulating the extent of exploitation is known as the generation rate G , and its definition is provided below.

$$G = G_0 e^{-\lambda(t-t_0)} = G_0 F \quad (8)$$

where

$$G_0 = GCP (X_{eq} - \lambda X) \quad (9)$$

$$GCP = \begin{cases} 0.5r_1, r_2 \geq GP \\ 0, r_2 < GP \end{cases} \quad (10)$$

where r_1 and r_2 represent random values of $[0,1]$, GCP signifies the generation rate control parameter, calculated through the generation probability GP .

Thus, the update method of EO is below.

$$X = X_{eq} + (X - X_{eq}) \cdot F + \frac{G}{\lambda V} (1 - F) \quad (11)$$

3 Structure of SCEO

In this subsection, to modify the overall performance of the basic EO, we introduce an improved EO algorithm, called SCEO. SCEO increases the performance of the basic EO by proposing new strategies while keeping the original update mechanism intact.

3.1 Adaptive Inertia Weight Mechanism

Regarding metaheuristic techniques, the exploration-exploitation balance is essential. As can be observed from Eq. (11), which F is employed to influence concentration changes to reach equilibrium, while the equilibrium candidate is a randomly chosen candidate from the equalization pool. Once it gets stuck in a local optimum, the best solution will prematurely stagnate during the search. Thus, to avoid this problem, the adaptive search for the random candidate is carried out, and we design an adaptive update formula to instead of Eq. (11). The novel updating equation is below.

$$X = \omega \times X_{eq} + (X - X_{eq}) \cdot F + \frac{G}{\lambda V} (1 - F) \quad (12)$$

where ω is an inertia weight.

Inspired by the PSO, the inertia weight mechanism is broadly employed in metaheuristic approaches. For instance, Ding et al. [18] revise the update equation of the salp swarm algorithm (SSA) by introducing an adaptive inertia weight factor. Wang et al. [19] use an inertia weight to modify the leader position of the SSA.

In this study, a novel adaptive inertial weight coefficient is designed and mathematically expressed as follows.

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \frac{1}{1 + ab^{(t-\frac{T}{2})}} \quad (13)$$

where ω_{\max} represents the upper limit of the weight; ω_{\min} signifies the lower limit of the weight; a and b are constants.

3.2 The Cauchy Mutation

To deal with the problem that EO tends to become stuck in local optimum and thus loses population diversity, this work employ the Cauchy mutation to improve the population diversity. The Cauchy distribution exhibits a narrower peak at its origin and a wider spread towards both extremes. By using the Cauchy mutation, a larger perturbation can be generated near the particle of the current variation, thereby allowing the Cauchy distribution function to search all areas of the solution space. This approach effectively leverages the mutation effects at the distribution's edges to optimize individuals, thereby achieving the global optimum while simultaneously preserving population diversity. The Cauchy distribution function is described below.

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{t}\right) \quad (14)$$

where $t > 0$.

Based on the Cauchy distribution function, we can further update the position of the particles. The update equation is below.

$$X' = X + X \times Cauchy(0, 1) \quad (15)$$

where *Cauchy* (0,1) represents a randomly generated value drawn from the Cauchy distribution, X denotes the position of the current iteration, X' signifies the position of the mutated particle.

3.3 Adaptive Sine Cosine Mechanism

In the standard EO, the update mechanism can readily become ensnared in local optima, leading to a decline in its capacity for global exploration. To address the limitations, this work introduces the position update equation of the sine cosine algorithm (SCA) [20] with inertia weight factor to perform adaptive search of the particles. The novel position update equation is below.

$$X'' = \begin{cases} X + r_3 \times \sin(r_4) \times |\omega \times r_5 \times gbest - X|, r_6 < 0.5 \\ X + r_3 \times \cos(r_4) \times |\omega \times r_5 \times gbest - X|, r_6 \geq 0.5 \end{cases} \quad (16)$$

$$r_3 = a \left(1 - \frac{It}{T} \right) \quad (17)$$

where a is set to 2; r_3 linearly decreases from 2 to 0. r_4 is derived from a uniform random distribution that encompasses the interval $[0, 2\pi]$. r_5 and r_6 signify random numbers in $[0, 1]$. $gbest$ is the global optimum solution. X signifies the position of the current iteration. X'' represents the updated position.

3.4 SCEO Algorithm

The developed SCEO algorithm updates the particle positions and does not affect the framework of the basic EO. Firstly, it injects an adaptive inertial weight to enhance the navigation of a stochastic equilibrium candidate. During iterations, the inertia weight continually adjusts the position of the equilibrium candidate and hopes to move it to a more prospective region. Next, the entire particle is updated again using the adaptive SCA to promote the search capability. The sine-cosine function decides the exploration horizon of the search individual's next position by controlling the direction of the particle's search. Once stuck in a local optimum, the search direction is changed, effectively avoiding search stagnation. Finally, population diversity is ensured by the addition of the Cauchy mutation mechanism in case of increased ability. By harnessing the distinctive characteristics of the Cauchy distribution, which exhibits a significant shift from a peak at the origin to an elongated distribution at its extremities, substantial perturbations are strategically induced near particles throughout the iterative process. This enables the algorithm to explore all areas and increase the particles diversity. Based on the above analysis, the flowchart of SCEO is given in Fig. 1.

3.5 Computational Complexity

The computational complexity of an optimization approach is an essential criterion for analyzing performance. In this paper, the Big-O notation is adopted to denote complexity. This complexity is contingent upon key factors such as the particle numbers (N), the dimensions (d) and the iterations (t), and (f) is the fitness evaluation.

$$O(\text{SCEO}) = O(\text{initialization}) + O(\text{fitness evaluation}) + O(\text{memory saving}) + O(\text{concentration update}) + O(\text{SCA updating strategy}) + O(\text{Cauchy mutation}).$$

Consequently, the computational complexity of SCEO can be elucidated below.

$$\begin{aligned} O(\text{SCEO}) &= O(N \times d + t \times f \times d + t \times N + t \times N \times d + N(1 + t + t \times d) + t \times N \times d) \\ &\cong O(t \times f \times d + t \times N \times d). \end{aligned}$$

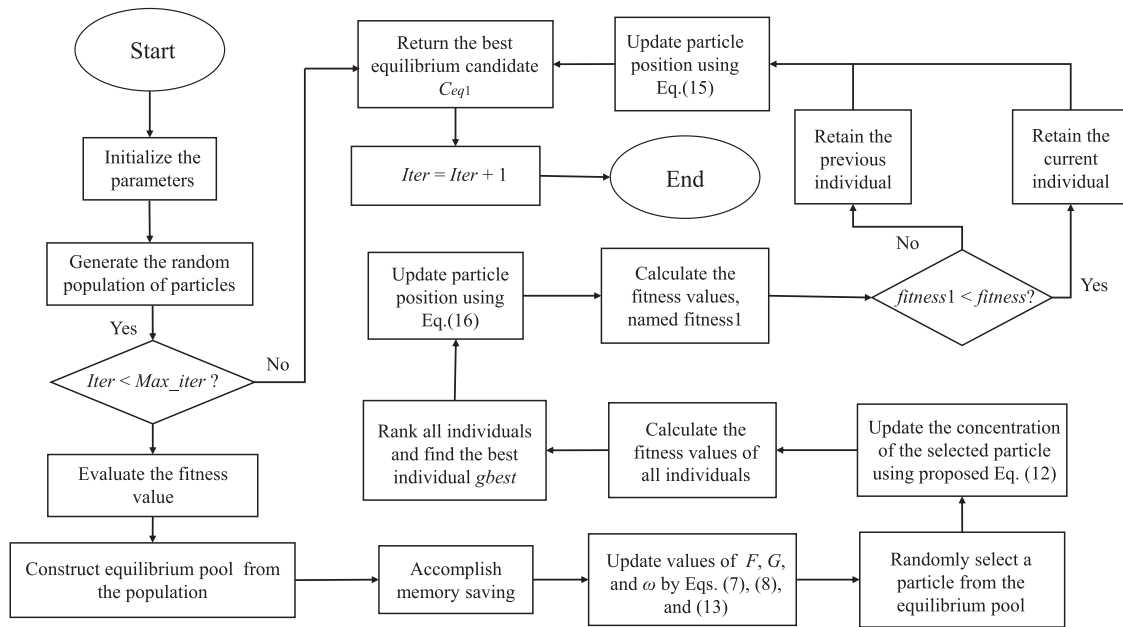


Figure 1: The flowchart of SCEO

The computational complexity of EO is expressed as $O(t \times f \times d + t \times N \times d)$. In essence, it can be summarized that the SCEO maintains the same computational complexity as the basic EO algorithm.

3.6 Convergence Analysis

In this subsection, we delve into a theoretical exploration of SCEO to substantiate its efficiency. The methodology employed for this proof is outlined in the subsequent steps.

We classify SCEO into three stages: adaptive inertial weight position update, adaptive sine cosine position update, and Cauchy mutation update.

Phase 1: From Eq. (12), during the iteration, $\omega X_{eq}, F, \lambda, V, GCP$ are constants held constant, then $\frac{1-F}{\lambda V}$ is also a constant, let $GCP = a_1, \frac{1-F}{\lambda V} = a_2, (t+1)th: X_{eq} = n_1, (t+2)th: X_{eq} = n_2$.

Thus, the problem is elaborated below:

$$X(t+2) - (F - \lambda a_1 a_2)^2 X(t) = n_1 (F - \lambda a_1 a_2) (1 - F + a_1 a_2) + n_2 (1 - F + a_1 a_2) \tag{18}$$

Let $F - \lambda a_1 a_2 = A$, we get

$$X(t+2) - A^2 X(t) = n_1 A (1 - F + a_1 a_2) + n_2 (1 - F + a_1 a_2) \tag{19}$$

The characteristic equation of Eq. (19) is

$$\lambda^2 - A^2 = 0 \tag{20}$$

Clearly, $\Delta = 4A^2 \geq 0$, the following two cases need to be considered.

(1) $\Delta = 0$. Only when $A = 0$, $\Delta = 0$. At this point, the characteristic roots are $\lambda_1 = \lambda_2 = 0$. The general solution is

$$X(t) = p_0 + p_1\lambda_1^t + p_2t\lambda_2^t = p_0 \quad (21)$$

where p_0 is the constant coefficient to be determined.

(2) $\Delta > 0$, where $A \neq 0$. The characteristic roots are $\lambda_{1,2} = \pm\sqrt{A^2}$. The general solution is

$$X(t) = p_0 + p_1\lambda_1^t + p_2\lambda_2^t \quad (22)$$

where p_0, p_1, p_2 are constant coefficients to be determined.

When $t \rightarrow \infty$, $X(t)$ has a limit and converges to a finite value, it implies that the iteration converges. If $X(t)$ of the above two cases is made to converge, the conditions need to be satisfied $\|\lambda_1\| < 1$, $\|\lambda_2\| < 1$.

When $\Delta = 0$, $\lambda_1 = \lambda_2 = 0$, the convergence condition is satisfied. In this case, $X(t)$ converges.

When $\Delta > 0$, the convergence region should satisfy:

$$\begin{cases} 4A^2 > 0 \\ |\pm\sqrt{A^2}| < 1 \end{cases} \quad (23)$$

From Eq. (23), the convergence region is $-1 < A < 1$. According to Eqs. (7) and (10), $F - \lambda a_1 a_2 \in (-1, 1)$, so $A \in (-1, 1)$ in the convergence range. In this case, $X(t)$ converges.

In summary, Phase 1 converges.

From Eq. (16), it can be seen that Phase 2 has the same search pattern as Phase 1, and it can be proved in the same way that Phase 2 also converges.

Phase 3: From Eq. (15), *Cauchy*(0,1) is constants held constant, let $(t + 1)$ th: *Cauchy* (0, 1) = V_1 , $(t + 2)$ th: *Cauchy* (0, 1) = V_2 .

A second-order linear differential equation is obtained, and the problem be expressed in the following form:

$$X(t+2) - (1 + V_1)(1 + V_2)X(t) = 0 \quad (24)$$

The characteristic equation of Eq. (24) is

$$\lambda^2 - (1 + V_1)(1 + V_2) = 0 \quad (25)$$

Clearly, $\Delta = 4(1 + V_1)(1 + V_2)$. Because $V_1 > 0$, $V_2 > 0$, $\Delta > 0$.

When $\Delta > 0$, the characteristic roots are $\lambda_{1,2} = \pm\sqrt{(1 + V_1)(1 + V_2)}$. The general solution is

$$X(t) = s_1\lambda_1^t + s_2\lambda_2^t \quad (26)$$

where s_1, s_2 are constant coefficients.

When $t \rightarrow \infty$, $X(t)$ has a limit and converges to a limited value, it implies that the iteration converges. If $X(t)$ is made to converge, the conditions need to be satisfied $\|\lambda_1\| < 1$, $\|\lambda_2\| < 1$.

Because $\lim_{t \rightarrow \infty} V_1 = 0$, $\lim_{t \rightarrow \infty} V_2 = 0$, $\lim_{t \rightarrow \infty} (\lambda_{1,2}) = \lim_{t \rightarrow \infty} (\pm\sqrt{(1 + V_1)(1 + V_2)}) = \pm 1$.

It satisfies the convergence condition. In this case, $X(t)$ converges. In summary, Phase 3 converges.

Overall, the SCEO algorithm converges.

4 Simulation and Comparisons

4.1 15 Test Functions

To ascertain the efficacy of the SCEO, 15 typical benchmark functions are applied. These functions are meticulously detailed in Table 1. According to Table 1, they have various features, which contain 7 unimodal functions and 8 multi-modal functions.

Table 1: Specific characterization of 15 benchmark test functions

| Function type | Function formulation | Search range | f_{\min} |
|---------------|---|----------------|------------|
| Unimodal | $f_1(x) = \sum_{i=1}^D x_i^2$ | $[-100,100]$ | 0 |
| | $f_2(x) = \sum_{i=1}^D ix_i^2$ | $[-10,10]$ | 0 |
| | $f_3(x) = \sum_{i=1}^D (\sum_{j=1}^i x_j)^2$ | $[-100,100]$ | 0 |
| | $f_4(x) = \max_i \{ x_i , 1 \leq x_i \leq D\}$ | $[-100,100]$ | 0 |
| | $f_5(x) = \sum_{i=1}^D ix_i^4$ | $[-1.28,1.28]$ | 0 |
| | $f_6(x) = \sum_{i=1}^D (10^6)^{(i-1)/(D-1)} x_i^2$ | $[-100,100]$ | 0 |
| | $f_7(x) = 10^6 \cdot x_1^2 + \sum_{i=2}^D x_i^6$ | $[-1,1]$ | 0 |
| Multi-modal | $f_8(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | $[-600,600]$ | 0 |
| | $f_9(x) = \sin^2(\pi x_1) + \sum_{i=1}^{D-1} [x_i^2 \cdot (1 + 10 \sin^2(\pi x_i)) + (x_i - 1)^2 - \sin^2(2\pi x_i)]$ | $[-10,10]$ | 0 |
| | $f_{10}(x) = 0.1D - (0.1 \sum_{i=1}^D \cos(5\pi x_i) - \sum_{i=1}^D x_i^2)$ | $[-1,1]$ | 0 |
| | $f_{11}(x) = \sum_{i=1}^D x_i^2 + (\sum_{i=1}^D 0.5x_i)^2 + (\sum_{i=1}^D 0.5x_i)^4$ | $[-5,10]$ | 0 |
| | $f_{12}(x) = \sum_{i=1}^D (0.2x_i^2 + 0.1x_i^2 \cdot \sin(2x_i))$ | $[-10,10]$ | 0 |
| | $f_{13}(x) = \left[\frac{1}{D-1} \sum_{i=1}^D (\sqrt{x_i} (\sin(50.0x_i^{0.2}) + 1)) \right]^2$ | $[-100,100]$ | 0 |
| | $f_{14}(x) = \sum_{i=1}^{D-1} [x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) - 0.4 \cos(4\pi x_{i+1}) + 0.7]$ | $[-15,15]$ | 0 |
| | $f_{15}(x) = \sum_{i=1}^{D-1} (x_i^2 + 2x_{i+1}^2)^{0.25} \cdot \left((\sin 50 (x_i^2 + x_{i+1}^2)^{0.1})^2 + 1 \right)$ | $[-10,10]$ | 0 |

4.2 Compared with EO and EO Variants

To thoroughly assess the performance of SCEO, this paper compares the SCEO algorithm with five novel EO variants, including the basic EO [9], LWMEO [15], m-EO [16], IS-EO [21], and IEO [17]. To ensure equitable evaluation, a consistent population scale of 30 and a uniform maximum iteration count of 500 are maintained across all tests. Each optimizer is subjected to 30 independent runs, all

conducted under identical conditions with a fixed function dimension of 100. Mean minimum (Mean) values and their corresponding standard deviations (Std) are employed as key indicators to gauge the performance of all methods. Furthermore, the Wilcoxon signed rank test is adopted to provide a comprehensive assessment. The findings, including Friedman's rank and the Wilcoxon rank-sum test, are succinctly presented in Table 2. The symbols "+/=/=" in Table 2 signify that the SCEO algorithm demonstrates superior, inferior, or similar performance compared to other algorithms, respectively. The "N/A" notation indicates instances where no statistical significance is observed.

Table 2: Comparisons of six optimizers with 100 dimensions

| Function | Results | EO | m-EO | LWMEO | IS-EO | IEO | SCEO |
|------------|---------|----------|----------|----------|----------|----------|-----------|
| f_1 | Mean | 3.54E-29 | 3.64E-38 | 0.0025 | 13.7161 | 5.21E-29 | 0 |
| | Std | 1.02E-28 | 1.07E-37 | 0.0014 | 53.8931 | 1.34E-28 | 0 |
| f_2 | Mean | 4.00E-29 | 8.77E-39 | 0.0397 | 4.0697 | 2.50E-30 | 0 |
| | Std | 6.27E-29 | 6.52E-39 | 0.0235 | 11.6954 | 2.44E-30 | 0 |
| f_3 | Mean | 23.5001 | 62.3350 | 3.86E+04 | 3.65E+05 | 161.4214 | 0 |
| | Std | 78.1813 | 223.4405 | 2.77E+04 | 1.12E+05 | 425.8067 | 0 |
| f_4 | Mean | 0.0220 | 3.52E-06 | 46.8066 | 70.5539 | 0.0024 | 2.52E-192 |
| | Std | 0.1151 | 8.44E-06 | 15.3180 | 15.3173 | 0.0022 | 0 |
| f_5 | Mean | 3.04E-51 | 2.11E-55 | 1.56E-06 | 2.53E-04 | 6.14E-49 | 0 |
| | Std | 9.68E-51 | 8.18E-55 | 1.39E-06 | 0.0013 | 9.82E-48 | 0 |
| f_6 | Mean | 3.29E-25 | 3.52E-34 | 1.02E+03 | 5.26E+08 | 6.13E-26 | 0 |
| | Std | 7.49E-25 | 1.38E-33 | 499.1461 | 2.82E+08 | 1.38E-25 | 0 |
| f_7 | Mean | 4.80E-66 | 7.25E-77 | 2.47E-05 | 0.4454 | 7.09E-62 | 0 |
| | Std | 7.09E-65 | 3.56E-76 | 2.68E-05 | 0.4936 | 3.74E-62 | 0 |
| f_8 | Mean | 0 | 0 | 0.0069 | 1.3297 | 0 | 0 |
| | Std | 0 | 0 | 0.0097 | 2.5550 | 0 | 0 |
| f_9 | Mean | 1.71E-25 | 7.29E-36 | 1.73E+03 | 64.6921 | 6.81E-26 | 0 |
| | Std | 3.28E-25 | 2.65E-35 | 903.4377 | 182.6546 | 2.52E-25 | 0 |
| f_{10} | Mean | 0 | 0 | 6.6663 | 0.0184 | 0 | 0 |
| | Std | 0 | 0 | 1.6731 | 0.0550 | 0 | 0 |
| f_{11} | Mean | 6.94E-28 | 4.02E-35 | 0.0044 | 1.0719 | 4.79E-28 | 0 |
| | Std | 1.60E-27 | 1.70E-34 | 0.0018 | 1.7113 | 1.26E-27 | 0 |
| f_{12} | Mean | 7.52E-32 | 7.50E-41 | 15.7772 | 0.1156 | 8.05E-32 | 0 |
| | Std | 9.87E-32 | 3.73E-40 | 40.0621 | 0.4238 | 1.30E-31 | 0 |
| f_{13} | Mean | 1.36E-09 | 6.14E-19 | 6.1309 | 0.5616 | 9.15E-10 | 0 |
| | Std | 4.06E-10 | 6.71E-19 | 1.1493 | 0.5346 | 4.48E-10 | 0 |
| f_{14} | Mean | 0 | 0 | 52.2477 | 31.3688 | 0 | 0 |
| | Std | 0 | 0 | 4.5693 | 47.5438 | 0 | 0 |
| f_{15} | Mean | 0.2708 | 0.1243 | 6.7335 | 3.2442 | 0.2882 | 0 |
| | Std | 0.0754 | 0.0412 | 1.0296 | 0.6360 | 0.0614 | 0 |
| Average | f-rank | 2.9333 | 1.8667 | 5.4000 | 5.6000 | 3.0000 | 1 |
| Overall | f-rank | 3 | 2 | 5 | 6 | 4 | 1 |
| p -value | +/=/- | 12/2/1 | 12/3/0 | 15/0/0 | 15/0/0 | 12/3/0 | N/A |

Based on Table 2, it reveals that the SCEO finds the theoretical best solution on all cases except f_4 . For this function, although the theoretical optimal values are not achieved, the SCEO is still better than the other competitors. Compared with the basic EO and IEO, SCEO performs better across 12 test functions. With respect to LWMEO and IS-EO, SCEO gets better results on all of the benchmark problems. For f_2 , SCEO and m-EO provide similar results. SCEO is superior to m-EO on 12 functions, and on the rest 3 functions the two algorithms get similar results. For f_8, f_{10} , and f_{14} , SCEO, m-EO and IEO obtains similar results. SCEO resembles basic EO on 2 functions and is inferior to basic EO on f_8 . Based on the Friedman’s rank test, it can be observed that SCEO ranks first followed by m-EO, EO, IEO, LWMEO, and IS-EO, which reveals that SCEO provides the best performance. The p -values consistently remain below the threshold of 0.05 within the 100-dimensional environments, which reveals that the distinctions between the SCEO and the other methods are statistically significant. In summary, SCEO presents more competitive performance than other comparative algorithms for 15 traditional benchmark functions.

4.3 Comparison with Other Metaheuristics

To prove the validity of the SCEO, it is compared with some advanced metaheuristics, including the enhanced SSA (RDSSA) [22], the improved grey wolf optimizer (OGWO) [23], the modified moth-flame optimization algorithm (DMMFO) [24], adaptive salp swarm optimization (ASSO) [25], and improved particle swarm optimization (BPSO) [26]. 15 test functions in Table 1 are adopted and the function’s dimension is configured to 100. The Mean and Std values for the six metaheuristics are meticulously documented in Table 3. Additionally, the outcomes of Friedman’s rank test and the Wilcoxon rank-sum test are presented at the base of the table. In Table 3, the symbols “+/-/=” are employed to signify instances where the SCEO algorithm demonstrates superior, inferior, or comparable performance to other algorithms, respectively. The notation “N/A” is employed when no statistically significant difference is observed.

Table 3: Comparisons of six metaheuristics with 100 dimensions

| Function | Results | RDSSA | OGWO | DMMFO | ASSO | BPSO | SCEO |
|----------|---------|-----------|----------|----------|----------|----------|-----------|
| f_1 | Mean | 1.34E-44 | 2.49E-17 | 3.40E+04 | 2.00E-26 | 3.34E-09 | 0 |
| | Std | 7.32E-44 | 7.59E-17 | 8.45E+03 | 2.63E-27 | 7.98E-09 | 0 |
| f_2 | Mean | 6.50E-49 | 9.01E-16 | 1.44E+04 | 9.97E-27 | 1.57E-07 | 0 |
| | Std | 3.56E-48 | 1.18E-15 | 2.73E+03 | 1.81E-27 | 3.17E-07 | 0 |
| f_3 | Mean | 6.50E-42 | 8.17E+03 | 2.33E+05 | 2.12E-25 | 1.79E+02 | 0 |
| | Std | 9.44E-41 | 2.10E+03 | 3.37E+04 | 1.15E-25 | 9.92E+01 | 0 |
| f_4 | Mean | 1.71E-24 | 1.6364 | 89.3226 | 3.51E-14 | 0.0190 | 2.52E-192 |
| | Std | 9.34E-24 | 2.6427 | 1.9068 | 3.88E-15 | 0.0159 | 0 |
| f_5 | Mean | 3.67E-98 | 4.11E-28 | 98.0966 | 1.46E-59 | 5.23E-11 | 0 |
| | Std | 1.95E-97 | 1.16E-27 | 35.2120 | 4.44E-60 | 1.73E-10 | 0 |
| f_6 | Mean | 6.13E-39 | 5.50E-12 | 2.11E+08 | 1.31E-21 | 5.85E-07 | 0 |
| | Std | 4.07E-38 | 6.52E-12 | 8.12E+07 | 4.02E-22 | 1.07E-06 | 0 |
| f_7 | Mean | 6.32E-102 | 5.61E-38 | 0.9955 | 4.20E-30 | 2.00E-16 | 0 |
| | Std | 6.99E-102 | 8.36E-38 | 0.5638 | 9.68E-30 | 5.08E-16 | 0 |
| f_8 | Mean | 0 | 3.27E-03 | 306.3598 | 0 | 1.01E-10 | 0 |

(Continued)

Table 3 (continued)

| Function | Results | RDSSA | OGWO | DMMFO | ASSO | BPSO | SCEO |
|------------|---------|----------|----------|----------|----------|----------|------|
| f_9 | Std | 0 | 6.58E-03 | 63.3286 | 0 | 3.28E-10 | 0 |
| | Mean | 6.53E-62 | 2.17E-23 | 1.50E+03 | 8.15E-27 | 2.75E-04 | 0 |
| f_{10} | Std | 3.58E-61 | 7.85E-23 | 295.8222 | 9.37E-28 | 9.73E-04 | 0 |
| | Mean | 0 | 9.27E-15 | 12.6508 | 0 | 1.68E-06 | 0 |
| f_{11} | Std | 0 | 6.71E-15 | 1.5389 | 0 | 5.61E-06 | 0 |
| | Mean | 3.63E-37 | 9.49E-16 | 506.1923 | 1.82E-28 | 2.80E-08 | 0 |
| f_{12} | Std | 1.99E-36 | 1.06E-15 | 83.3650 | 4.76E-29 | 8.40E-08 | 0 |
| | Mean | 7.39E-65 | 2.33E-26 | 178.2879 | 4.15E-29 | 2.35E-09 | 0 |
| f_{13} | Std | 1.31E-64 | 1.28E-25 | 34.3196 | 5.43E-30 | 6.96E-09 | 0 |
| | Mean | 6.17E-15 | 8.49E-05 | 8.3849 | 1.29E-07 | 0.0038 | 0 |
| f_{14} | Std | 5.74E-14 | 8.81E-05 | 0.2792 | 4.82E-09 | 0.0026 | 0 |
| | Mean | 0 | 2.04E-15 | 2.12E+03 | 0 | 8.82E-07 | 0 |
| f_{15} | Std | 0 | 2.24E-15 | 375.8016 | 0 | 4.48E-06 | 0 |
| | Mean | 7.69E-02 | 9.12E-01 | 7.4673 | 1.18E-07 | 0.8032 | 0 |
| | Std | 1.07E-02 | 2.67E-01 | 0.2637 | 3.45E-09 | 0.2328 | 0 |
| Average | f-rank | 1.8667 | 4.2000 | 6.0000 | 2.6000 | 4.7333 | 1 |
| Overall | f-rank | 2 | 4 | 6 | 3 | 5 | 1 |
| p -value | +/=/- | 12/3/0 | 15/0/0 | 15/0/0 | 12/3/0 | 15/0/0 | N/A |

From the data in Table 3, for f_8, f_{10}, f_{14} , RDSSA, ASSO, and SCEO obtain the theoretical optimal solution. Compared with DMMFO, OGWO, and BPSO SCEO reaches better performance on all of its cases. With respect to RDSSA, and ASSO, SCEO obtains better on 12 cases and has similar results on 3 cases. According to the Friedman mean ranks acquired by eight approaches on 15 test functions, it can be shown that SCEO is first, followed by RDSSA, ASSO, OGWO, BPSO, and DMMFO. The p -values consistently remain below the threshold of 0.05 within the 100-dimensional environments, which reveals that the differences observed between SCEO and the alternative methods hold substantial statistical significance.

Table 4 presents the average runtime duration (measured in seconds) of seven different techniques across the set of 15 benchmark functions. Notably, Table 4 highlights that EO boasts the shortest average runtime compared to the other techniques. Importantly, the average runtime of SCEO closely aligns with that of EO across all scenarios. This observation underscores the fact that the devised SCEO achieves performance enhancements without compromising runtime efficiency.

Table 4: Comparison of runtime (seconds) of the algorithms with 100 dimensions

| Algorithm | EO | RDSSA | OGWO | DMMFO | ASSO | BPSO | SCEO |
|-----------------|--------|---------|---------|---------|--------|---------|--------|
| Average runtime | 7.4162 | 11.4671 | 11.4796 | 12.7285 | 9.7434 | 51.7899 | 7.9734 |

4.4 Convergence Analysis

In this subsection, we investigate the convergence behavior of each method by visually showing the convergence process of these methods. The 15 benchmark test problems are employed from Table 1, and the dimension is set to 100. Fig. 2 displays the convergence curves of eleven methods across selected cases.

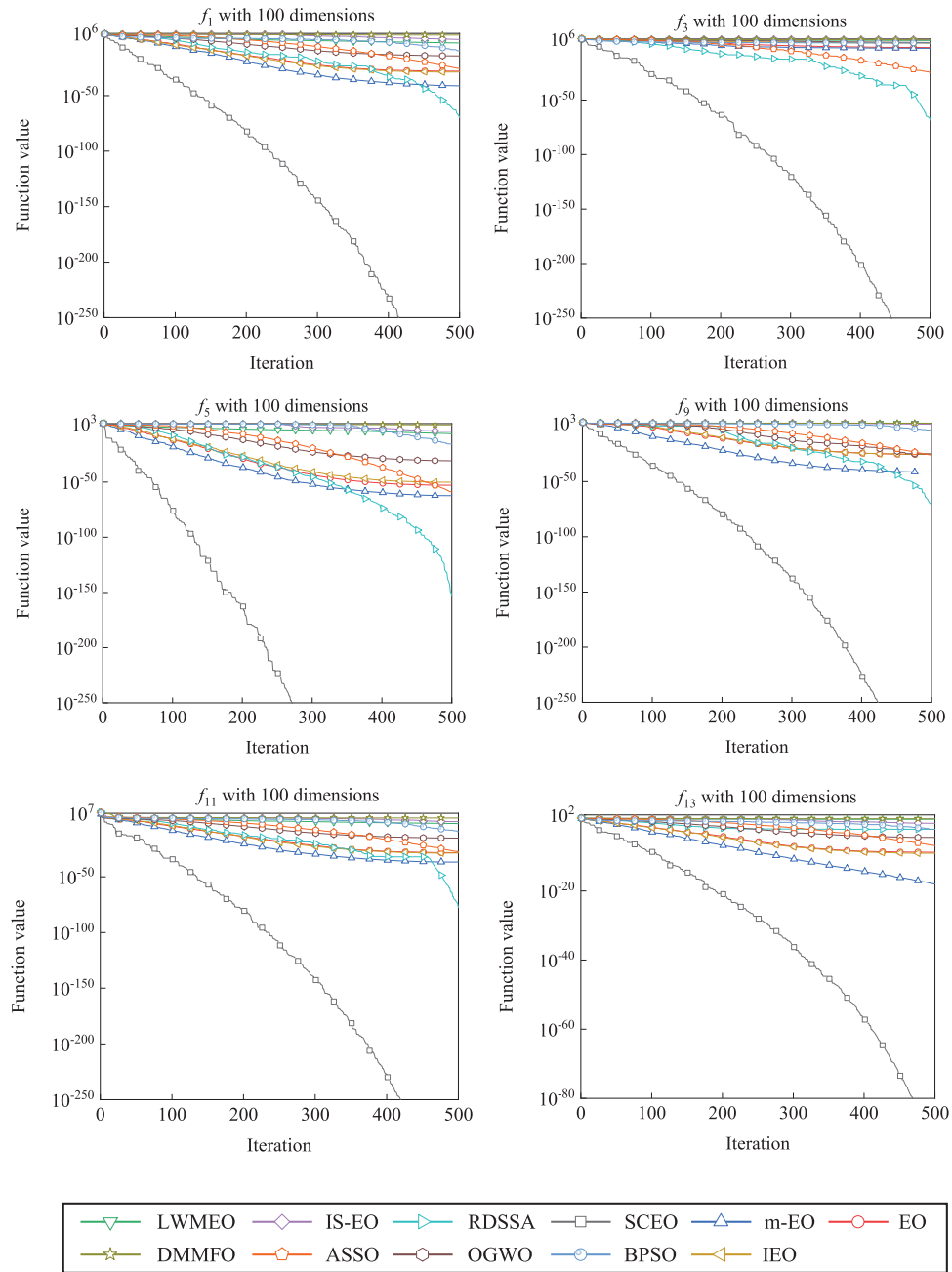


Figure 2: Convergence curves of eleven algorithms with 100 dimensions

Observing Fig. 2, it becomes evident that SCEO exhibits a gradual convergence rate in the initial phases of the search process. At the later stage, the convergence speed is accelerated and the convergence curve is smoother. This behavior is attributed to a strategic trade-off: during the early stages, the algorithm intentionally prioritizes preserving population diversity, thus modestly compromising early convergence. Subsequently, the algorithm transitions from broad global exploration to focused local search, boosting solution accuracy by exploiting previously explored areas. According to the experimental results, SCEO finds better solutions quicker than other all compared algorithms across more than half of the cases tested. Notably, SCEO not only accelerates convergence but also demonstrates heightened accuracy in solution outcomes when contrasted with alternative algorithms. Overall, these results underscore the pronounced convergence advantages of SCEO over other EO variants and the broader spectrum of compared algorithms.

4.5 Experiments on CEC 2017 Test Suite

This subsection comprehensively evaluates the performance of SCEO employing the CEC2017 test suite. The outcomes of SCEO are meticulously compared against the basic EO, various EO variants, and a spectrum of advanced metaheuristics including the EO, m-EO, LWME0, IS-EO, IEO, RDSSA, OGWO, DMMFO, ASSO, BPSO, and VC-SSA [18]. For this experiment, the upper limit of evaluations is prudently established at $D \times 10^4$. Consistency is maintained in terms of parameter settings for both the compared methods and SCEO, mirroring those employed in Section 4.2. All methods are subjected to 30 independent runs, and the functional dimension remains consistent at 100. The performance of the 13 methods on the CEC2017 benchmark function with 100 dimensions is summarized in Table 5.

According to Table 5, SCEO performs better to other comparative algorithms across 15 functions and is inferior to the other algorithms on 14 functions. With respect to m-EO, IS-EO, RDSSA, OGWO, DMMFO, and ASSO, SCEO is superior on all functions. Compared with BPSO, SCEO gets better and worse results on 24 and 5 functions. For F15 and F19, SCEO is worse compared with the EO. SCEO beats IEO in 18 cases and is inferior in 11 cases. SCEO is better than LWME0 on 25 functions, but SCEO is inferior on F11, F13, F15, and F18. For F15, F17, and F19, IEO and BPSO have better results than SCEO. For F11, LWME0 and BPSO are superior to SCEO. According to the average ranking, SCEO obtains the first rank, followed by IEO, EO, LWME0, BPSO, VC-SSA, m-EO, OGWO, DMMFO, RDSSA, IS-EO, ASSO. SCEO outperforms EO on 27 functions and underperforms on 2 functions. SCEO outperforms VC-SSA on all functions. Overall, these data reveal that SCEO is significantly superior to its competitors on the CEC2017 test suite.

5 Robot Path Planning

In robotics, MRPP is the critical intermediate part. MRPP is the process of finding the best path without collisions with obstacles from a specified starting point to a predetermined destination point, based on certain evaluation standards. Since mobile robots emerged, many scholars have conducted a series of studies to design efficient path-planning methods for mobile robots.

5.1 Robot Path Planning Problem Description

Recently, many metaheuristics-based MRPP techniques have been developed. Wang et al. [27] use a modified SSA variant to tackle the MRPP problem. Wu et al. [28] harness an modified ant colony optimization (ACO) for the MRPP task. Wang et al. [29] introduce an improved SSA and apply the novel SSA variant to the MRPP problem. Luan et al. [30] propose an MRPP method based on a hybrid genetic algorithm (HGA).

In this subsection, we adopt the SCEO to plan the best collision-avoidance path. Acknowledging the intricacies of the MRPP problem, an adaptation function with the route length and obstacle avoidance is established to assess the mass of the produced paths based on the SCEO algorithm in the MRPP problem. The fitness function is described below.

$$\text{fit} = kL + (1 - k) E \quad (27)$$

where k is the weight, L is the route length, E is the path collision penalty function.

5.2 Results and Analysis

In this subsection, we utilize the developed SCEO for the MRPP task. The SCEO is compared with some classical optimization techniques such as GWO [31], ABC [32], PSO [10], FA [11], and VC-SSA [18]. It is examined in the three maps provided in [33]. The general parameter settings are the same as those in the original work. The shortest path lengths generated by the six algorithms on the three maps are shown in Table 6. The trajectory routes for six algorithms are shown in Figs. 3–5.

Table 6: The shortest route length comparisons of six methods under three environmental terrains

| Terrain | SCEO length | GWO length | ABC length | PSO length | FA length | VC-SSA length |
|---------|----------------|---------------|---------------|---------------|--------------|------------------|
| Map 1 | 7.4443 | 7.7625 | 7.7527 | 7.7328 | 7.5281 | 7.4516 |
| Map 2 | 14.3043 | 14.3565 | 14.8418 | 14.5037 | 14.4288 | 14.3115 |
| Map 3 | 15.7845 | 16.3281 | 16.2919 | 16.1925 | 16.1031 | 15.8430 |

From the data in Table 6, it can be viewed that the path planning task SCEO-based plans the shortest path for collision avoidance on all three terrains compared with the other approaches, which illustrates that the SCEO can get satisfying paths for simple and complex environmental settings in the MRPP task. The trajectory routes produced by all algorithms are described in detail below.

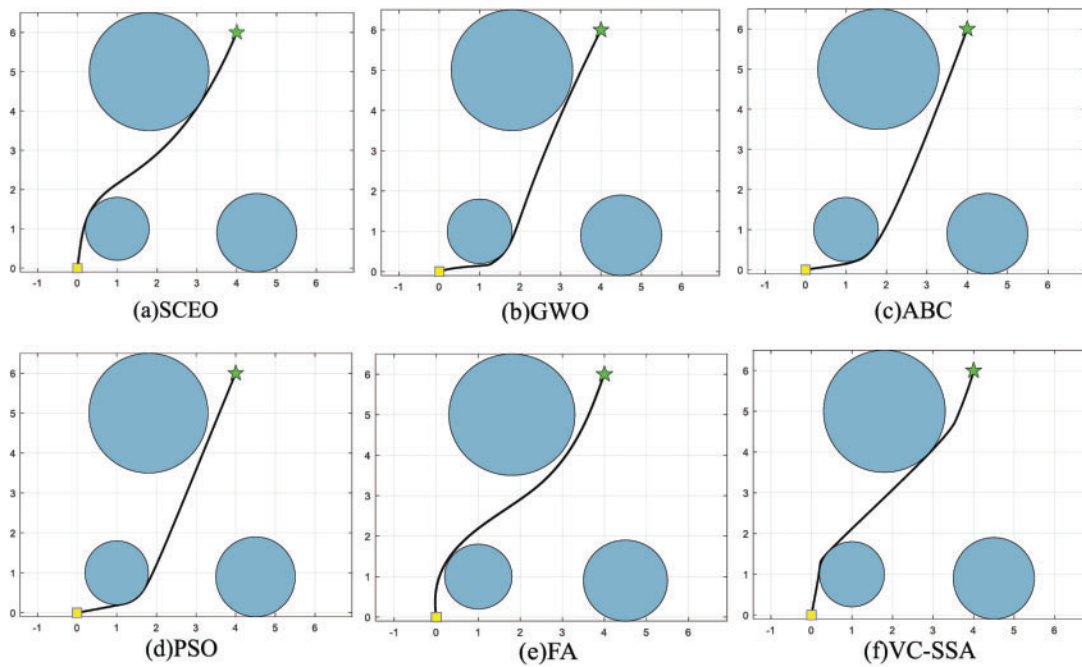


Figure 3: Trajectory routes of the six methods in Map 1

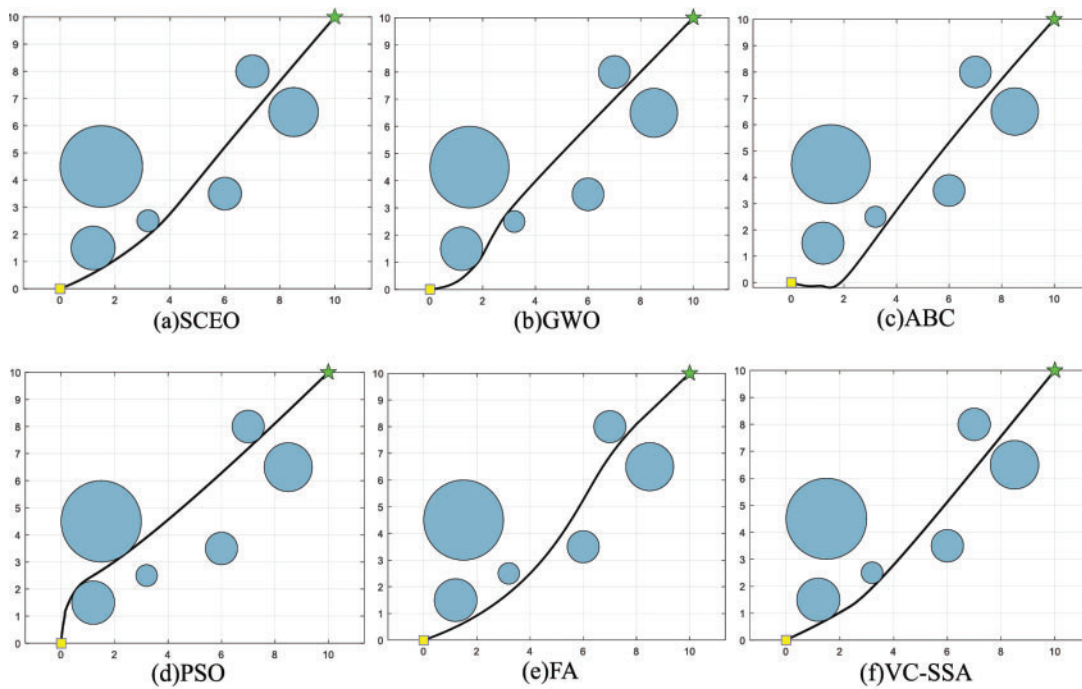


Figure 4: Trajectory routes of the six methods in Map 2

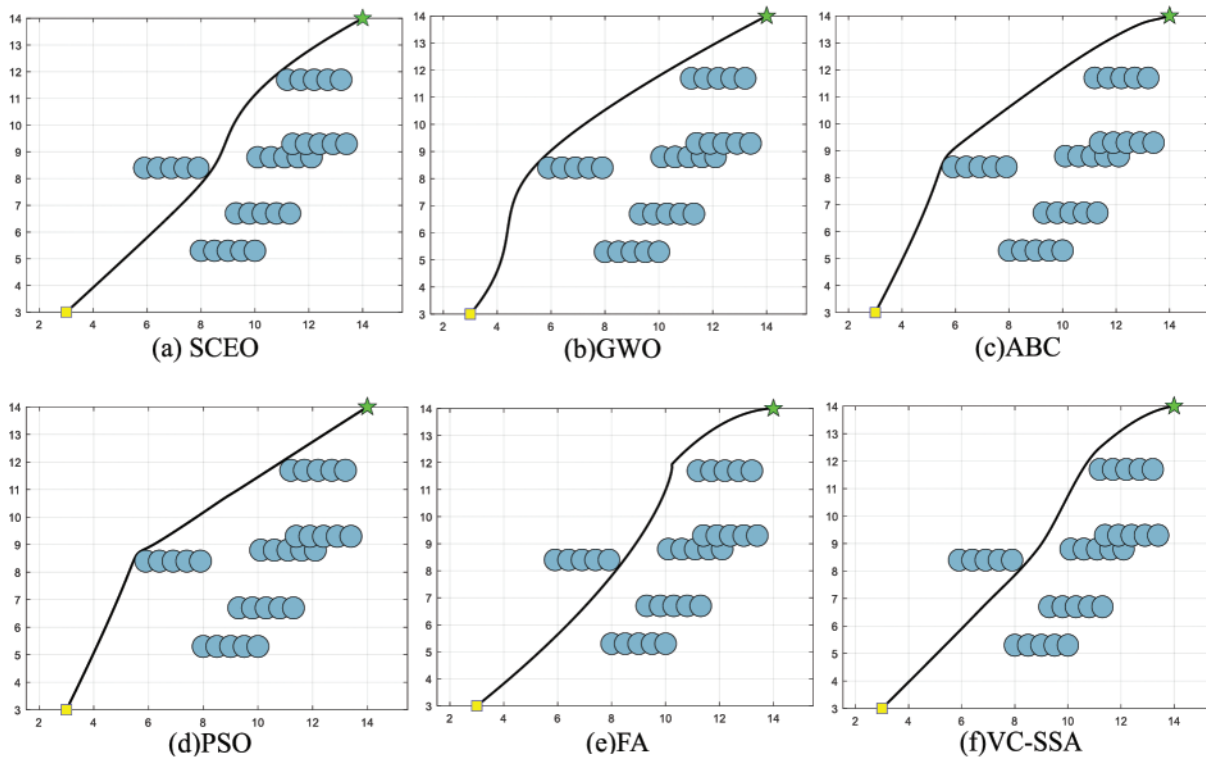


Figure 5: Trajectory routes of the six methods in Map 3

Figs. 3a–3f displays the comparison of the trajectory routes of the six algorithms in the first map. As can be noticed from the figure, all methods can plan a safe route because this environment is a simple terrain. Compared with other algorithms, SCEO has the shortest path. PSO, ABC, and GWO map an identical path, which is smooth but not optimal. FA, VC-SSA, and SCEO design an alternative trajectory route. The route of VC-SSA has more inflection points. FA has a smooth path but not its optimal path. Figs. 4a–4f depict the trajectory routes of the second environment map. Upon examining the figures, it becomes evident that SCEO yields the shortest path among the showcased options. SCEO, VC-SSA, ABC, and FA produce similar trajectory routes, while GWO and PSO provide another trajectory routes. The route of ABC gets stuck in local optimum, which results in redundancy of the produced trajectories, and thus the path is the longest. The paths of VC-SSA, GWO, PSO, FA are smooth but not optimal. The best trajectories of environment map are shown in Figs. 5a–5f. GWO, ABC, and PSO provide a trajectory route through the outside of the danger area, which avoids collision with the obstacle area but increases the fuel burn of the robot. The route of FA has more inflection points and increases the path length. The route of VC-SSA is smoother but not optimal. In conclusion, SCEO is the most favorable algorithmic tool, and is the superior algorithm among all the techniques.

6 Conclusion

In this article, a novel EO variant, referred to as SCEO, is developed to concentrate on reaching an appropriate exploration-exploitation balance, preventing the reduction of population diversity and escaping from the local optimal. SCEO introduces three mechanisms to enhance stability. First,

a new update equation based on an adaptive inertia weight mechanism is employed to efficiently balance between exploration-exploitation. Next, the adaptive SCA method is combined to enhance the exploration capability. Lastly, to enhance population diversity, the Cauchy mutation mechanism is introduced. The resultant SCEO is subjected to a comprehensive evaluation, spanning 15 classical benchmark problems, the CEC2017 test function suite, and the Mobile Robot Path Planning (MRPP) challenge. These evaluations involve a comparison with the foundational EO, various EO variants, and a diverse array of other metaheuristic techniques. The simulation results reveal that improved mechanisms are extremely effective and the comprehensive performance of SCEO emerges as remarkably competitive. SCEO presents great potential for solving various complex problems. Consequently, in our future work, we will concentrate on the applications of SCEO to practical and complex engineering optimization problems, such as vehicle scheduling problems.

Acknowledgement: The authors extend their gratitude to the Ministry of Education of Yunnan Province and Yunnan University for their invaluable support and assistance.

Funding Statement: This research endeavor receives support from the National Natural Science Foundation of China [Grant Nos. 61461053, 61461054, and 61072079], Yunnan Provincial Education Department Scientific Research Fund Project [2022Y008].

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yuting Liu, Hongwei Ding, Gaurav Dhiman, Zhijun Yang; data collection: Yuting Liu, Zongshan Wang, Peng Hu; analysis and interpretation of results: Yuting Liu, Zongshan Wang; draft manuscript preparation: Yuting Liu, Zongshan Wang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] D. Rani, "Solving non-linear fixed-charge transportation problems using nature inspired non-linear particle swarm optimization algorithm," *Applied Soft Computing*, vol. 146, pp. 110699, 2023.
- [2] Z. H. Jiang, J. Q. Peng, R. X. Yin, M. M. Hu, J. G. Cao *et al.*, "Stochastic modelling of flexible load characteristics of split-type air conditioners using grey-box modelling and random forest method," *Energy and Buildings*, vol. 273, pp. 112370, 2022.
- [3] Y. T. Yang, H. L. Chen, A. A. Heidari and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, no. 8, pp. 114864, 2021.
- [4] I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. L. Chen and A. H. Gandomi, "INFO: An efficient optimization algorithm based on weighted mean of vectors," *Expert Systems with Applications*, vol. 195, no. 12, pp. 116516, 2022.
- [5] H. Su, D. Zhao, A. A. Heidari, L. Liu, X. Zhang *et al.*, "RIME: A physics-based optimization," *Neurocomputing*, vol. 532, no. 5, pp. 183–214, 2023.
- [6] M. Dehghani, Z. Montazeri, E. Trojovská and P. Trojovský, "Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 259, no. 1, pp. 110011, 2023.

- [7] M. Abdel-Basset, R. Mohamed, M. Jameel and M. Abouhawwash, "Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems," *Knowledge-Based Systems*, vol. 262, no. 3, pp. 110248, 2023.
- [8] M. Nssibi, G. Manita and O. Korbaa, "Advances in nature-inspired metaheuristic optimization for feature selection problem: A comprehensive survey," *Computer Science Review*, vol. 49, pp. 100559, 2023.
- [9] A. Faramarzi, M. Heidarinejad, B. Stephens and M. Seyedali, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, pp. 105190, 2020.
- [10] R. Poli, J. Kennedy and T. Blackwell, "Particle swarm optimization: An overview," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [11] I. Fister, I. Fister Jr, X. S. Yang and J. Brest, "A comprehensive review of firefly algorithms," *Swarm and Evolutionary Computation*, vol. 13, no. 3, pp. 34–46, 2013.
- [12] A. Wunna, M. K. Naik, R. Panda, B. Jena and A. Abraham, "A novel interdependence based multi-level thresholding technique using adaptive equilibrium optimizer," *Engineering Applications of Artificial Intelligence*, vol. 94, pp. 103836, 2020.
- [13] M. Abdel-Basset, R. Mohamed, S. Mirjalili, R. K. Chakraborty and M. J. Ryan, "Solar photovoltaic parameter estimation using an improved equilibrium optimizer," *Solar Energy*, vol. 209, no. 3, pp. 694–708, 2020.
- [14] J. Wang, B. Yang, D. Li, C. Zeng, Y. Chen *et al.*, "Photovoltaic cell parameter estimation based on improved equilibrium optimizer algorithm," *Energy Conversion and Management*, vol. 236, no. 3, pp. 114051, 2021.
- [15] J. Liu, W. Li and Y. Li, "LWMEO: An efficient equilibrium optimizer for complex functions and engineering design problems," *Expert Systems with Applications*, vol. 198, no. 9, pp. 116828, 2022.
- [16] S. Gupta, K. Deep and S. Mirjalili, "An efficient equilibrium optimizer with mutation strategy for numerical optimization," *Applied Soft Computing*, vol. 96, no. 1, pp. 106542, 2020.
- [17] L. Yang, Z. Xu, Y. Liu and G. Tian, "An improved equilibrium optimizer with a decreasing equilibrium pool," *Symmetry*, vol. 14, no. 6, pp. 1227, 2022.
- [18] H. W. Ding, X. G. Cao, Z. S. Wang, G. Dhiman, P. Hou *et al.*, "Velocity clamping-assisted adaptive salp swarm algorithm: Balance analysis and case studies," *Mathematical Biosciences and Engineering*, vol. 19, no. 8, pp. 7756–7804, 2022.
- [19] Z. S. Wang, H. W. Ding, J. Wang, P. Hou, A. S. Li *et al.*, "Adaptive guided salp swarm algorithm with velocity clamping mechanism for solving optimization problems," *Journal of Computational Design and Engineering*, vol. 9, no. 6, pp. 2196–2234, 2022.
- [20] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, no. 63, pp. 120–133, 2016.
- [21] X. Zhang and Q. Lin, "Information-utilization strengthened equilibrium optimizer," *Artificial Intelligence Review*, vol. 55, no. 5, pp. 4241–4274, 2022.
- [22] H. Ren, J. Li, H. Chen and C. Li, "Stability of salp swarm algorithm with random replacement and double adaptive weighting," *Applied Mathematical Modelling*, vol. 95, no. 5, pp. 503–523, 2021.
- [23] X. Yu, W. Y. Xu and C. L. Li, "Opposition-based learning grey wolf optimizer for global optimization," *Knowledge-Based Systems*, vol. 226, pp. 107139, 2021.
- [24] L. Ma, C. Wang, G. N. Xie, M. Shi, Y. Ye *et al.*, "Moth-flame optimization algorithm based on diversity and mutation strategy," *Applied Intelligence*, vol. 51, no. 8, pp. 5836–5872, 2021.
- [25] F. A. Ozbay and B. Alatas, "Adaptive salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media," *Multimedia Tools and Applications*, vol. 80, no. 26–27, pp. 34333–34357, 2021.
- [26] H. R. R. Zaman and F. S. Gharehchopogh, "An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems," *Engineering with Computers*, vol. 38, no. 4, pp. 2797–2831, 2022.
- [27] Z. S. Wang, H. W. Ding, J. J. Yang, J. Wang, B. Li *et al.*, "Advanced orthogonal opposition-based learning-driven dynamic salp swarm algorithm: Framework and case studies," *IET Control Theory & Applications*, vol. 16, no. 10, pp. 945–971, 2022.

- [28] L. Wu, X. Huang, J. Cui, C. Liu and W. Xiao, "Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot," *Expert Systems with Applications*, vol. 215, no. 23, pp. 119410, 2023.
- [29] Z. S. Wang, H. W. Ding, Z. J. Yang, B. Li, Z. Guan *et al.*, "Rank-driven salp swarm algorithm with orthogonal opposition-based learning for global optimization," *Applied Intelligence*, vol. 52, no. 7, pp. 7922–7964, 2022.
- [30] P. G. Luan and N. T. Thinh, "Hybrid genetic algorithm based smooth global-path planning for a mobile robot," *Mechanics Based Design of Structures and Machines*, vol. 51, no. 3, pp. 1758–1774, 2023.
- [31] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [32] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [33] D. Agarwal and P. S. Bharti, "Implementing modified swarm intelligence algorithm based on Slime moulds for path planning and obstacle avoidance problem in mobile robots," *Applied Soft Computing*, vol. 107, no. 4, pp. 107372, 2021.